AUTOMATED KNEE DEGENERATIVE ARTHRITIS REPORTS GENERATION THROUGH VISUAL DATA

A Project Report

Submitted by:

K.N.G.S.S.ADITHYA (208T1A05F3)

N.KRISHNA MAHITHA (208T1A05F2)

B.N.LOKESH (208T1A05D3)

L.S.CH.DEEPIKA (208T1A05F5)

G. MEENAKSHI (208T1A05E3)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

at

Under the esteemed guidance of

Mr. V. Naga Malleswara rao

Assistant Professor

DHANEKULA INSTITUTE OF ENGINEERING AND TECHNOLOGY

GANGURU, A.P. (INDIA) - 521139

AFFILIATED TO JNTUK, KAKINADA, ANDHRA PRADESH (INDIA)

APRIL & 2024

DECLARATION

We hereby declare that the project entitled **"AUTOMATED KNEE DEGENERATIVE ARTHRITIS REPORTS GENERATION THROUGH VISUAL DATA ANALYSIS"** submitted for the B. Tech (CSE) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Students

K.N.G.S.S.ADITHYA	(208T1A05F3)
N.KRISHNA MAHITHA	(208T1A05F2)
B.N.LOKESH	(208T1A05D3)
L.S.CH.DEEPIKA	(208T1A05F5)
G.MEENAKSHI	(208T1A05E3)

Place:

Date:

DHANEKULA INSTITUTE OF ENGINEERING AND TECHNOLOGY

GANGURU, A.P. (INDIA) - 521139

AFFILIATED TO JNTUK, KAKINADA, ANDHRA PRADESH (INDIA)

CERTIFICATE

This is to certify that the project titled “**AUTOMATED KNEE DEGENERATIVE ARTHRITIS REPORTS GENERATION THROUGH VISUAL DATA ANALYSIS**” is the bonafide work carried out by **K.N.G.S.S.Adithya (208T1A05F3), N.Krishna Mahitha (208T1A05F2), B.N.Lokesh (208T1A05D3), L.S.CH.Deepika (208T10A5F5), G.Meenakshi (208T1A05E3)** students of B Tech (CSE) of Dhanekula Institute of Engineering and Technology, affiliated to JNT University, Kakinada, AP(India) during the academic year 2023-24, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

SIGNATURE OF THE GUIDE	**SIGNATURE OF THE HOD**
Mr. V.Naga Malleswara Rao (Asst. Professor, CSE)	Dr. K. Sowmya (HOD & Professor)

EXTERNAL EXAMINER

VISION – MISSION - PEOs

Institute Vision	Pioneering Professional Education through Quality
Institute Mission	Providing Quality Education through state-of-art infrastructure, laboratories and committed staff. Moulding Students as proficient, competent, and socially responsible engineering personnel with ingenious intellect. Involving faculty members and students in research and development works for betterment of society.
Department Vision	To empower students of Computer Science and Engineering Department to be technologically adept, innovative, global citizens possessing human values.
Department Mission	Encourage students to become self-motivated and problem-solving individuals. Prepare students for professional career with academic excellence and leadership skills. Empower the rural youth with computer education. Create Centre's of excellence in Computer Science and Engineering
Program Educational Objectives (PEOs)	Graduates of B.Tech (Computer Science & Engineering)will be able to **PEO1:** Excel in Professional career by demonstrating the capabilities of solving real time problems throughComputer-based system, Machine learning and allied software applications. **PEO2:** Able to pursue higher education and research. **PEO3:** Communicate effectively, recognize, and incorporate appropriate tools and technologies in the chosen profession. **PEO4:** Adapt to technological advancements by continuous learning, team collaboration and decision making.

POs

1	**Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2	**Problem analysis**: Identify, formulate, review research literature,and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.'
6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and withsociety at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcome Statements (PSO`s):

1	Have expertise in algorithms, networking, web applications and software engineering for efficient design of computer-based systems of varying complexity.
2	Qualify in national international level competitive examinations for successful higher studies and employment.

PROJECT MAPPINGS

Batch No:	C7
Project Title	AUTOMATED KNEE DEGENERATIVE ARTHRITIS REPORTS GENERATION THROUGH VISUAL DATA ANALYSIS
Project Domain	Machine Learning
Type of the Project	Application
Guide Name	Mr.V.NagaMalleswara Rao
Student Roll No	Student Name
208T1A05F3	K.N.G.S.S.Adithya
208T1A05F2	N.Krishna Mahitha
208T1A05D3	B.N.Lokesh
208T1A05F5	L.S.CH.Deepika
208T1A05E3	G.Meenakshi

COURSE OUTCOMES: At the end of the Course/Subject, the students will be able to

CO. No	Course Outcomes (COs)	POs	PSOs	Blooms Taxonomy & Level
R20C501.1	Identify the real-world problem with a set of requirements to design a solution.	1,2,3,4,6,8, 9, 10, 11,12	1,2	Level-3 Applying
R20C501.2	Implement, Test and Validate the solution against the requirements for a given problem.	1,2,3,4,5,8, 9, 10,11,12	1,2	Level-4 Analyzing
R20C501.3	Lead a team as a responsible member in developing software solutions for real world problems and societal issues with ethics.	1,3,4,5,6,8,9,10, 11,12	1,2	Level-4 Analyzing
R20C501.4	Participate in discussions to bring technical and behavioral ideas for good solutions.	1,2,3,4,6,7,9,10, 12	1,2	Level-5 Evaluating
R20C501.5	Express ideas with good communication skills during presentations.	1,5,7,8,9,10	1,2	Level-6 Creating
R20C501.6	Learn new technologies to contribute in the software industry for optimal solutions	1,2,3,5,8,9,10,11, 12	1,2	Level-6 Creating

Course Outcomes vs PO`s Mapping:

Courses Out Comes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
R20C501.1	3	3	3	3	-	3	-	3	3	3	3	3
R20C501.2	3	3	3	3	3	-	-	3	3	3	3	3
R20C501.3	3	-	3	3	3	3	-	3	3	3	3	3
R20C501.4	3	3	3	3	-	3	3	-	3	3	-	3
R20C501.5	3	-	-	-	3	-	3	3	3	3	-	-
R20C501.6	3	3	3	-	3	-	-	3	3	3	3	3
Total	18	12	15	12	12	9	6	15	18	18	15	15
Average	3	3	3	3	3	3	3	3	3	3	3	3

Justification of Mapping of Course Outcomes with Program Outcomes:

1. R20C501.1 is strongly mapped with PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO10, PO11, PO12 because it involves using engineering knowledge and teamwork to analyze and solve problems which include development of solutions for complex problems while considering society and environment.
2. R20C501.2 is strongly mapped with PO1, PO2, PO3, PO4, PO5, PO8, PO9, PO10, PO11, PO12 because we apply engineering knowledge to analyze and solve problems which include development of solutions where we use the latest tools and follow ethics and also communicate effectively & manage the project.
3. R20C501.3 is strongly linked with PO1, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11, and PO12 because engineering knowledge is applied to solve time management problems while considering society and the environment. Also, we follow professional ethics, communicate in team for solutions and manage project.
4. R20C501.4 is strongly mapped with PO1, PO2, PO3, PO4, PO6, PO7, PO9, PO10, PO12 because we apply engineering knowledge to analyze and solve problems and develop solutions for complex problems which include health, society, and the environment with safety and ethics, which includes communication to manage project.

5. R20C501.5 is strongly mapped with PO1, PO5, PO7, PO8, PO9, PO10 because we apply engineering knowledge to meet requirements and estimates how it affects the environment where we follow professional ethics, work individually or in a team, and communicate effectively for providing the solution.
6. R20C501.6 is strongly mapped with PO1, PO2, PO3, PO5, PO8, PO9, PO10, PO11, PO12 because we apply engineering concepts to analyze the problems and develop the solutions for health issues which includes the usage of modern tools where we follow professional ethics, communicate effectively and recognize the importance of learning latest technologies.
7. R20C501.1 is strongly mapped to PSO-1 and PSO-2 which involves identifying the real world problem with a set of requirements to design a solution has it requires expertise in CNN model algorithm and networking knowledge to build secure applications using network firewalls and network security groups and web application development skills to build web applications.It also helps in recognizing a problem at national and international examinations.

Course Outcomes vs PSOs Mapping:

Courses Outcomes	PSO1	PSO2
R20C501.1	3	3
R20C501.2	3	3
R20C501.3	3	3
R20C501.4	3	3
R20C501.5	3	3
R20C501.6	3	3
Total	18	18
Average	3	3

Justification of Mapping of Course Outcomes with Program Specific Outcomes:

1. R20C501.1 is strongly mapped to PSO-1 and PSO-2 which involves identifying the real world problem with a set of requirements to design a solution has it requires expertise in CNN model algorithm and networking knowledge to build secure applications using network firewalls and network security groups and web application development skills to build web applications. It also helps in recognizing a problem at national and international examinations.

2. R20C501.2 is strongly mapped to PSO-1 and PSO-2 to implement a system, expertise in algorithms is crucial and helps to perform in national and international competitive examinations and this allows to apply theoretical knowledge to practical solutions.
3. R20C501.3 is strongly mapped to PSO-1 and PSO-2 as the team can develop solutions for real world problems which includes expertise in the algorithms and leads to the active participation and involvement in the national and international examination.
4. R20C501.4 is strongly mapped to PSO-1 and PSO-2. As the quality of the resulted proposals from the discussions held will be dependent on the expertise one has in the algorithms, networking and web application contexts. This will help in national and international examinations as one can actually think of increasing the quality and enhancing the solution.
5. R20C501.5 is strongly mapped to PSO-1 and PSO-2 as good communication can lead to better conveying of the complex algorithms and networking concepts and also helps in expressing the solutions at national and international level of examinations.
6. R20C501.6 is strongly mapped to PSO-1 and PSO-2 as it is essential to handle different levels of complex problems, it requires learning new technologies and staying updated with those technologies and such learners can excel in national and international recruitments too.

Mapping Level	Mapping Description
1	Low Level Mapping with PO & PSO
2	Moderate Mapping with PO & PSO
3	High Level Mapping with PO & PSO

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without whom it would ever have come into existence. To them we lay the words of gratitude imprinted with us.

We would like to thank our respected Principal, **Dr. RAVI KADIYALA** and, **Dr. K. SOWMYA**, Head of the Department, Computer Science and Engineering for their support throughout our major project.

It is our sincere obligation to thank our guide, **Mr. V. NAGA MALLESWARA RAO**, Department of Computer Science and Engineering, for his timely valuable guidance and suggestions for this major project.

We would like to express our immense pleasure in expressing an immeasurable sense of gratitude to **Mr. M. RAVI KANTH**, Assistant Professor and Project Coordinators for giving opportunity to make this project a successful one.

We also extend our thanks to all the faculty members of the Computer Science & Engineering department for their valuable contributions in this project.

We would like to extend our warm appreciation to all our friends for sharing their knowledge and valuable contributions in this project.

Finally, we express our deep sense of gratitude to our parents for their continuous support throughout our academic career and their encouragement in the completion of this project successfully.

K.N.G.S.S.ADITHYA	208T1A05F3
N.KRISHNAMAHITHA	208T1A05F2
B.N.LOKESH	208T1A05D3
L.S.CH.DEEPIKA	208T1A05F5
G.MEENAKSHI	208T1A05E3

ABSTRACT

Knee degenerative arthritis is a common medical condition that affects the knee joint. Knee Degenerative arthritis causes major disability in patients all over the world. The computerized reporting procedure requires effort and expertise. Manual diagnosis, segmentation of knee joints are still used in clinical practice. Manual diagnosis of this disease involves observing X-ray images of the knee area and classifying it under five grades using the Kellgren – Lawrence (KL) system. Despite the fact that they are time-consuming and sensitive to user variance. As a result, we have the proposed system employing the CNN model with Computer Vision to increase the clinical workflow efficiency and overcome the constraints of the generally used method. We can also implement the report generation system that generates medical reports based on X-ray image features. By extracting the relevant features such as joint space narrowing, and cartilage degeneration, the system generates detailed and objective reports.

Keywords: ***Knee Degenerative Arthritis, X-rays, CNN, Computer Vision.***

List of Figures

Figure No	Name of the Figure	Page No
Figure 1.1	The Joint	4
Figure 1.2	The Human Knee	6
Figure 1.3	Normal vs Knee Degenerative Arthritis	12
Figure 1.4	Stages of Knee Degenerative Arthritis	15
Figure 1.5	Stages and Symptoms of Knee Degenerative Arthritis	19
Figure 1.6	Sample Image	20
Figure 1.7	Types of Image Processing	24
Figure 1.8	Example of Computer Vision	27
Figure 1.9	Flask Integration	33
Figure 3.1	System Architecture	46
Figure 3.2	UML Diagrams Classification	47
Figure 3.3	Use Case Diagram	48
Figure 3.4	Class Diagram	49
Figure 3.5	Sequence Diagram	50
Figure 3.6	Collaboration Diagram	51
Figure 3.7	Activity Diagram	52
Figure 3.8	Component diagram	53
Figure 3.9	Deployment diagram	54
Figure 3.10	State Chart Diagram	55
Figure 3.11	Artificial Neuron	57
Figure 3.12	Feedforward Vs Feedback Networks	59
Figure 3.13	Work Flow of Computer Vision	62

Figure 3.14	CNN Architecture	68
Figure 3.15	Convolutional Layer in CNN	72
Figure 3.16	Pooling Layer in CNN	73
Figure 3.17	Flattening Layer in CNN	74
Figure 3.18	Fully Connected Layer in CNN	75
Figure 3.19	Architecture of InceptionV3	78
Figure 3.20	Architecture of Xception	81
Figure 3.21	Flow Chart of Algorithm	91
Figure 3.22	Home Page	94
Figure 3.23	Upload Page	94
Figure 5.1	Acceptance Testing Image	100
Figure 5.1	Upload Page	101
Figure 5.2	After Image Upload	101
Figure 5.3	Invalid Image	102
Figure 5.4	Prediction for Healthy Knee	102
Figure 5.5	Prediction for Doubtful Knee	103
Figure 5.6	Prediction for Minimal Level Arthritis	103
Figure 5.7	Prediction for Moderate Level Arthritis	104
Figure 5.8	Prediction for Severe level Arthritis	104
Figure 5.10	Confusion matrix	105
Figure 5.10	Model Evaluation for Inception V3	107
Figure 5.11	Xception model Fine Tuning	107

List of Tables

Table No:	Table Name	Page No:
Table 1	Unit Testing Table for Inception V3 Model	97
Table 2	Integration Testing Table for Xception Model	98
Table 3	Integration Testing Table	99
Table 4	Acceptance Testing Table	100
Table 5	Classification Report for Xception Model	106

Table of Contents

	Declaration	i
	Certificate	ii
	Vision-Mission-PEO's	iii
	PO's-PSO's	iv
	Project Mappings	vi
	Acknowledgement	ix
	Abstract	xi
	List of Figures	xii
	List of Tables	xiv
1.	**INTRODUCTION**	**1**
	1.1 Problem Definition	2
	1.2 Project Overview	2
	1.2.1 Origin of the Problem	2
	1.2.2 Real Time Application of Proposed work	3
	1.2.3 Objective	3
	1.2.4 Basic Concepts	3
	1.2.5 Scope	29
	1.3 Hardware Specification	29
	1.4 Software Specification	29

2.	**LITERATURE SURVEY**	**30**
	2.1 Literature Study	30
	2.2 Existing System	33
	2.3 Proposed System	34
	2.4 Feasibility Study	34
3.	**SYSTEM ANALYSIS & DESIGN**	**41**
	3.1 Requirements Specification	41
	3.1.1 Functional Requirements	44
	3.1.2 Non-Functional Requirements	45
	3.2 System Architecture	46
	3.3 UML Diagrams	46
	3.3.1 Use Case Diagram	47
	3.3.2 Class Diagram	49
	3.3.3 Sequence Diagram	50
	3.3.4 Collaboration Diagram	51
	3.3.5 Activity Diagram	52
	3.3.6 Component Diagram	53
	3.3.7 Deployment Diagram	54
	3.3.8 State Chart Diagram	55
	3.4 Algorithms	56
	3.4.1 Deep Learning	56

	3.4.2 Open CV	61
	3.4.3 CNN	66
	3.4.4 Transfer Learning	76
	3.4.4.1 Inception V3	77
	3.4.4.2 Xception	80
	3.5 Pseudo Codes	90
	3.6 Flow Charts	91
	3.7 Software Environment	91
	3.8 Steps for executing the project	92
4.	**TESTING**	**95**
	4.1 Types of Tests	95
	4.2 Levels of Testing	97
	4.2.1 Unit Testing	97
	4.2.2 Integration Testing	99
	4.2.3 Acceptance Testing	99
5.	**RESULTS**	**101**
	5.1 Output Screens	101
	5.2 Metrics	105
6.	**CONCLUSIONS**	**108**
7.	**FUTURE SCOPE**	**109**
8.	**REFERENCES**	**110**
9.	**PUBLISHED PAPER**	**112**

INTRODUCTION

1. INTRODUCTION

Knee arthritis, also known as osteoarthritis of the knee, is a degenerative joint disease that primarily affects the cartilage in the knee joint. Cartilage is the smooth tissue that covers the ends of bones and allows them to glide smoothly against each other during movement.

When this cartilage wears down over time, it can lead to pain, stiffness, and decreased mobility in the knee. Because of the higher prevalence of asymptomatic OA, it is approximated that 250 million people all over the world suffer from OA[1].

In today's world, technology is changing where we deal with health issues. One of the big problem many people experience is knee arthritis, especially when they get older. An estimated 47.8 million Americans currently suffer from arthritis, an often debilitating joint disease, costing the United States' economy nearly $6.1 billion per year[2].

It's a painful condition that affects millions of people worldwide. With more people living longer and different lifestyles, knee arthritis is becoming more common. So, we need new and smart ways to diagnose, treat, and keep track of it. This thesis focuses on using technology to help manage knee arthritis better.

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].

Our goal is to make it easier for users to understand what's going on with their knees and to give a better idea of what to expect. We're using tools like data analysis and prediction to make this happen. Our vision is that this will lead to better treatment plans and help patients take more control over their health.

By bringing together different ideas and methods, we aim to create a strong system for making reports that goes beyond in healthcare. We believe that we can use technology to make life better for people with knee arthritis and improve healthcare for everyone. So, we explore the details of creating better reports for managing knee arthritis and look forward to a future where healthcare is more personalized and informed.

1.1 PROBLEM DEFINITION

The primary goal is to develop a prediction engine which will allow the users to check whether they have knee degenerative arthritis sitting at home. The user need not visit the doctor unless for further treatments. The prediction engine requires a large dataset and efficient deep learning algorithms to predict the presence of the disease.

1.2 PROJECT OVERVIEW

Osteoarthritis is becoming more common in middle-aged men and women around the world. Majority of those affected are not aware of their condition. They ignore the problem because they believe that it will not affect them. They don't feel severe discomfort until a certain time, which is already too late. If it is not diagnosed sooner, one may not lead a happy or healthy life. The severity of the condition must be determined manually by the doctors by going through the X-ray images in order to advise the patients which is time and cost consuming. Therefore, we develop a Deep Learning models that can accurately predict the severity of knee osteoarthritis among the patients from their x-ray image as healthy, doubtful, mild, moderate and severe.

1.2.1 ORIGIN OF THE PROBLEM

Reasons for Knee Degenerative Arthritis

Pain is the most common symptom of osteoarthritis in the knee. Knee pain is a common complaint that affects people of all ages. Knee pain may be the result of an injury, such as a ruptured ligament or torn cartilage. Medical conditions including arthritis, and infections also can cause knee pain. Many types of minor knee pain respond well to self-care measures. Physical therapy and knee braces also can help relieve pain. In some cases, however, our knee may require surgical repair.

Other reasons are:

- Our knee feels stiff, particularly when we've been sitting for a long time.
- Our knee looks swollen or feels puffy.
- We hear a cracking or grinding noise when we move our knee.
- Our knee feels wobbly, as if it could buckle or "give out."
- Our knee might lock up, or feel as if it is stuck.

1.2.2 REAL TIME APPLICATION OF PROPOSED WORK:

The proposed system has the potential to revolutionize knee osteoarthritis diagnosis and management. By providing real-time, objective assessments, which include:

1. Assist doctors in making treatment decisions more quickly.
2. Make sure everyone gets the same quality of information about their knees.
3. Give doctors more time to spend with patients.
4. Assign physicians to the most urgent cases first.
5. Help doctors in remote locations to make decisions.
6. Provide researchers with lot of data to help them better understand about arthritis condition.

1.2.3 OBJECTIVE

The objective of this research project is to utilize Convolutional Neural Network (CNN) methods, particularly with pre-trained models like Inception for image classification and by developing multi-stage classification with Xception reporting methods. This system aims to accurately classify knee images into various stages of degeneration and produce detailed reports. In order to improve patient care and healthcare delivery, the research aims to assess the performance of the proposed system in terms of classification accuracy, report generation efficiency, and clinical utility.

1.2.4 BASIC CONCEPTS

THE JOINT

A joint is any place in our body where two bones meet. They’re part of our skeletal system. We might see joints referred to as articulations. We have hundreds of joints throughout our body, and many ways healthcare providers group them together (classification).

Joints are usually classified based on:

Their function: How they move.

Their composition: What they’re made of (histologically).

Joints support our body from head to toe. Whether it’s a joint we are aware of (like our ankle) or some we have maybe never heard of (like the joints that hold our skull together), all of our joints help us use our body every day.

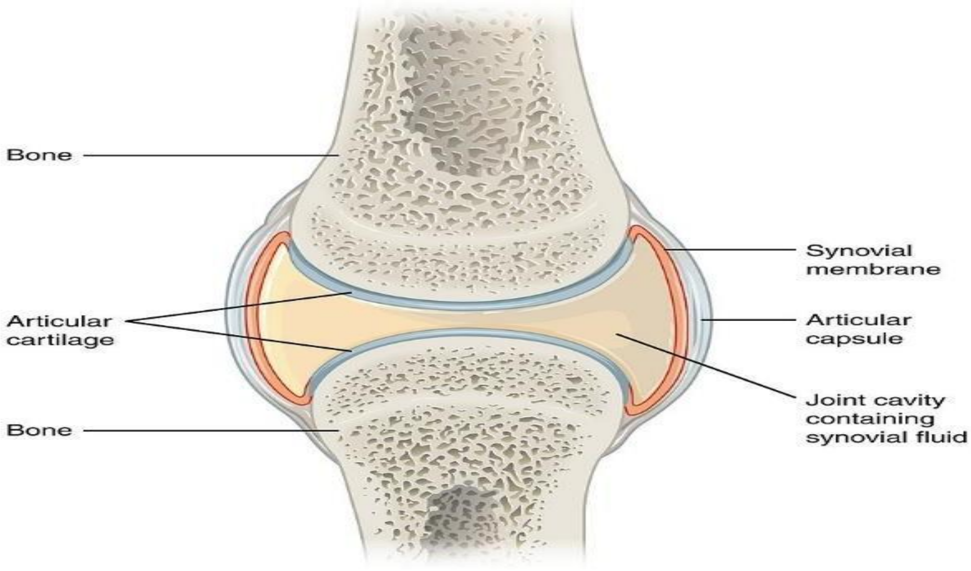


Figure 1.1 The joint

Our joints are made of bones and the connective tissues that hold them together, including:

Cartilage

Cartilage is a strong, flexible connective tissue that protects your joints and bones. It acts as a shock absorber throughout our body. Cartilage at the end of our bones reduces friction and prevents them from rubbing together when we use our joints.

Tendons

A tendon is a cord of strong, flexible tissue, similar to a rope. Tendons connect our muscles to our bones. Tendons let us move our limbs.

They also help prevent muscle injury by absorbing some of the impact our muscles take when we run, jump or do other movements.

Ligaments

A ligament is a fibrous connective tissue that attaches bone to bone, and usually serves to hold structures together and keep them stable.

Nerves

Nerves are like cables that carry electrical impulses between our brain and the rest of our body. These impulses help us feel sensations and move our muscles. They also maintain certain autonomic functions like breathing, sweating or digesting food.

THE KNEE

The knee is a hinge joint that is responsible for weight-bearing and movement. It consists of bones, meniscus, ligaments, and tendons.

The knee joint is a type of synovial joint – one in which the ends of two bones are free moving, joined only by connecting ligaments and a fluid-filled cavity called the synovial space[4].

The knee joint is one of the largest and most complex joints in the body. It is constructed by 4 bones and an extensive network of ligaments and muscles.

The knee is designed to fulfill a number of functions:

- Support the body in an upright position without the need for muscles to work.
- Helps to lower and raise the body.
- Provides stability.
- Acts as a shock absorber.
- Allows twisting of the leg.
- Makes walking more efficient.
- Helps propel the body forward.

Below, explains the basic components of knee anatomy.

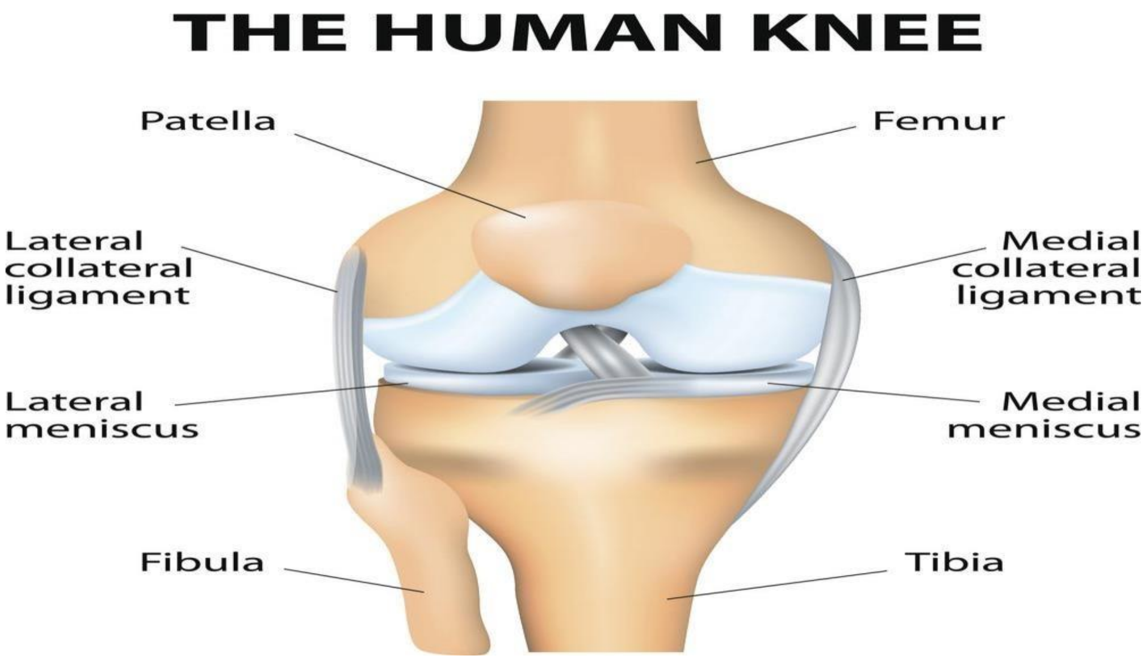


Figure 1.2 The Human Knee

Bones

The femur (thigh bone), tibia (shin bone), and patella (kneecap) make up the bones of the knee. The knee joint keeps these bones in place.

The patella is a small, triangle shaped bone that sits at the front of the knee, within the quadriceps muscle. It is lined with the thickest layer of cartilage in the body because it endures a great deal of force.

Cartilage

There are two types of cartilage in the knee:

Meniscus

These are crescent-shaped discs that act as a cushion, or "shock absorber" so that the bones of the knee can move through their range of motion without rubbing directly against each other.

The menisci also contain nerves that help improve balance and stability and ensure the correct weight distribution between the femur and tibia.

The knee has two menisci:

medial – on the inner side of the knee, this is largest of the two

lateral – on the outer side of the knee

Articular cartilage

Articular cartilage is hyaline cartilage and is 2 to 4 mm thick. Unlike most tissues, articular cartilage does not have blood vessels, nerves, or lymphatics. It is composed of a dense extracellular matrix (ECM) with a sparse distribution of highly specialized cells called chondrocytes.

Articular cartilage found on the femur, the top of the tibia, and the back of the patella; it is a thin, shiny layer of cartilage. It acts as a shock absorber and helps bones move smoothly over one another.

Knee cartilage is a smooth, rubbery tissue that covers the ends of the bones in the knee joint and cushions it, allowing it to bend and straighten. It also reduces friction in the joint.

Ligaments

Ligaments are tough and fibrous tissues; they act like strong ropes to connect bones to other bones, preventing too much motion and promoting stability.

Knee ligament injuries can be caused by trauma, such as a car accident. Or they can be caused by sports injuries. An example is a twisting knee injury in basketball.

Tendons

These tough bands of soft tissue provide stability to the joint. They are similar to ligaments, but instead of linking bone to bone, they connect bone to muscle.

The largest tendon in the knee is the patellar tendon, which covers the kneecap, runs up the thigh, and attaches to the quadriceps.

Muscles

Although they are not technically part of the knee joint, the hamstrings and quadriceps are the muscles that strengthen the leg and help flex the knee.

The quadriceps are four muscles that straighten the knee. The hamstrings are three muscles at the back of the thigh that bend the knee.

Joint capsule

The joint capsule is a membrane bag that surrounds the knee joint. It is filled with a liquid called synovial fluid, which lubricates and nourishes the joint.

Bursa

There are approximately 14 of these small fluid-filled sacs within the knee joint. They reduce friction between the tissues of the knee and prevent inflammation.

Common Injuries

Knees are most often injured during sports activities, exercising, or as a result of a fall. Pain and swelling, difficulty with weight bearing, and instability are the most common symptoms experienced with a knee injury.

KNEE DEGENERATIVE ARTHRITIS

Knee osteoarthritis (OA), also known as degenerative joint disease, is a common type of knee arthritis that's caused by wear and tear and cartilage loss. Osteoarthritis of the knee happens when cartilage in our knee joint breaks down. When this happens, the bones in our knee joint rub together, causing friction that makes our knees hurt, become stiff or swell.

It is a chronic degenerative joint disease that clinically manifests as pain, joint deformity and limited mobility that typically causes disability[5].

Osteoarthritis in the knee can’t be cured but there are treatments that can relieve symptoms and slow our condition’s progress. Osteoarthritis of the knee is very common. Approximately 46% of people will develop it during their lifetimes. Knee osteoarthritis is classified as either primary or secondary, depending on its cause.

Primary knee osteoarthritis is the result of articular cartilage degeneration without any known reason. This is typically thought of as degeneration due to age as well as wear and tear. Secondary knee osteoarthritis is the result of articular cartilage degeneration due to a known reason.

Women are more likely than men to develop osteoarthritis of the knee. Most people develop this condition after age 40. But other factors such as injury or genetics can cause it to happen earlier.

Knee pain is the most common symptom of osteoarthritis in the knee, making it painful for us to jog, run, climb stairs or kneel. It can also make our knees feel stiff or swollen. Over time, osteoarthritis of the knee can change the shape of our knee joint, making our joint feel unstable or wobbly.

Strong evidence shows that age, ethnicity, BMI, the number of co-morbidities, MRI-detected infrapatellar synovitis, joint effusion, and both radiographic and the baseline of OA severity are predictive for clinical progression of knee osteoarthritis.

X-ray is the first-choice imaging modality in the diagnosis of KOA. In general, anteroposterior and lateral radiographs of the knee joint, particularly the weight-bearing views, are required for the comparison of bilateral knee joints[6].

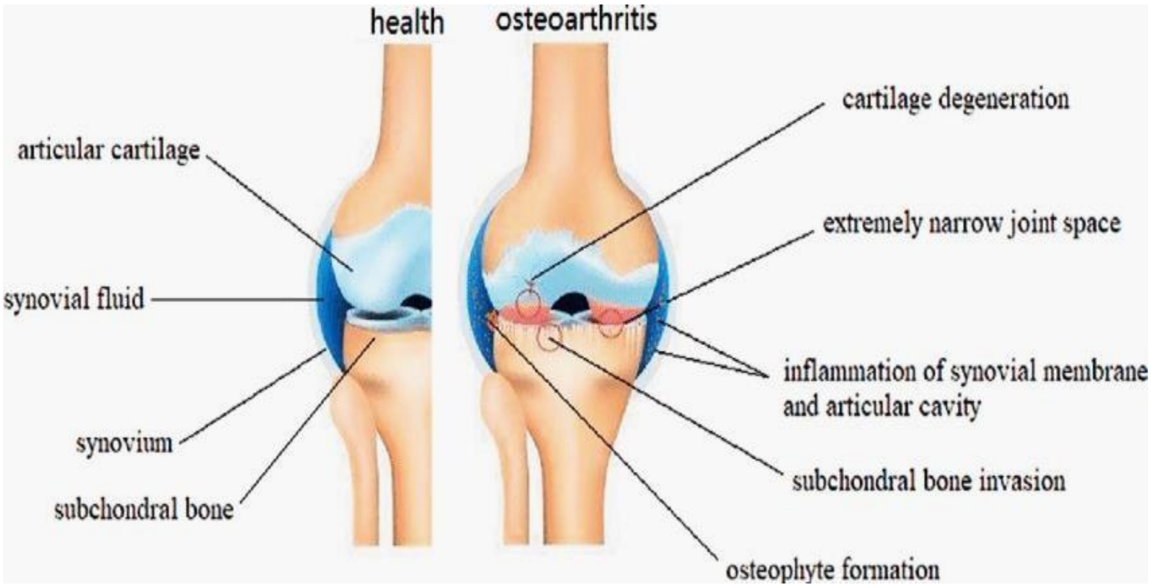


Figure 1.3 Normal vs Knee Degenerative Arthritis

Causes of Knee Degenerative Arthritis

Osteoarthritis of the knee happens when your knee joint cartilage wears out or is damaged. Articular cartilage is tough, rubbery tissue on the ends of your bones that lets you bend and move. Meniscal cartilage absorbs shock from pressure on your knee. Primary osteoarthritis usually develops slowly as you age. As you get older, normal wear and tear on your joints might contribute to their cartilage breaking down. You have crooked bones or joints, such as having knocked knees.

You can wear out or damage your knee joint cartilage if:

- You're overweight. If your body mass index (BMI) is 30 or more, you're seven times more likely to develop osteoarthritis in your knee than someone with a lower BMI.
- You injure your knee or have an old knee injury.
- You frequently put stress on your knee at your job or playing sports.
- You inherited a tendency to develop osteoarthritis of the knee.
- You have crooked bones or joints, such as having knocked knees.

Symptoms of Knee Degenerative Arthritis

Pain is the most common symptom of osteoarthritis in the knee. Your knee might hurt when you move it, or even when you are just sitting still.

Other symptoms are:

- Your knee feels stiff, particularly when you first get up or when you've been sitting for a long time.
- Your knee looks swollen or feels puffy.
- You hear a cracking or grinding noise when you move your knee.
- Your knee feels wobbly, as if it could buckle or "give out."
- Your knee might lock up, or feel as if it is stuck.

Diagnosis for Knee Degenerative Arthritis

Your healthcare provider will do a physical examination and ask about your medical history. The physical examination might include checks to see:

- If your knee joint area is red or sore.
- If there's a sign you injured your knee.
- How much you can move your knee. This is called your range of motion.
- If your knee feels "loose," which can mean your joint isn't stable.
- The way you walk, in case you have gait problems that affect your knee. A gait problem is when you don't walk as you would normally.

Tests for Knee Degenerative Arthritis

- X-ray.
- Magnetic resonance imagery (MRI).
- Blood tests.
- Joint aspiration (arthrocentesis).
- Management and Treatment.

Treatments for Knee Degenerative Arthritis

Treatment might include nonsurgical treatments, injections and surgery. Typically, healthcare providers try non-surgical treatments before recommending surgery.

Non-surgical treatments include:

- Using pain medications.
- Doing physical therapy.
- Maintaining a healthy weight.
- Using a knee brace.
- Using orthotics such as insoles or special footwear.
- Cortisone (steroid) injections.

Surgical treatments include:

- Cartilage grafting. Healthy cartilage is used to fill a hole in your cartilage.
- Knee osteotomy.
- Partial knee replacement.
- Total knee replacement.

Factors for recommending surgery

Your provider might recommend surgery if:

- Your symptoms aren't better after non-surgical treatments such as medication and physical therapy.
- Your symptoms affect your quality of life.
- Tests show your knee joint is beginning to disintegrate.

Prevention

While you can't always prevent osteoarthritis of the knee, there are steps you can take to reduce the risk you'll develop it:

- Maintain a healthy weight.
- Get plenty of rest.
- If you jog or run, do so on grass or soft surfaces.
- Vary your fitness routine with low-impact exercises such as swimming or cycling.
- Add light strength training to your fitness routine.

STAGES OF KNEE DEGENERATIVE ARTHRITIS

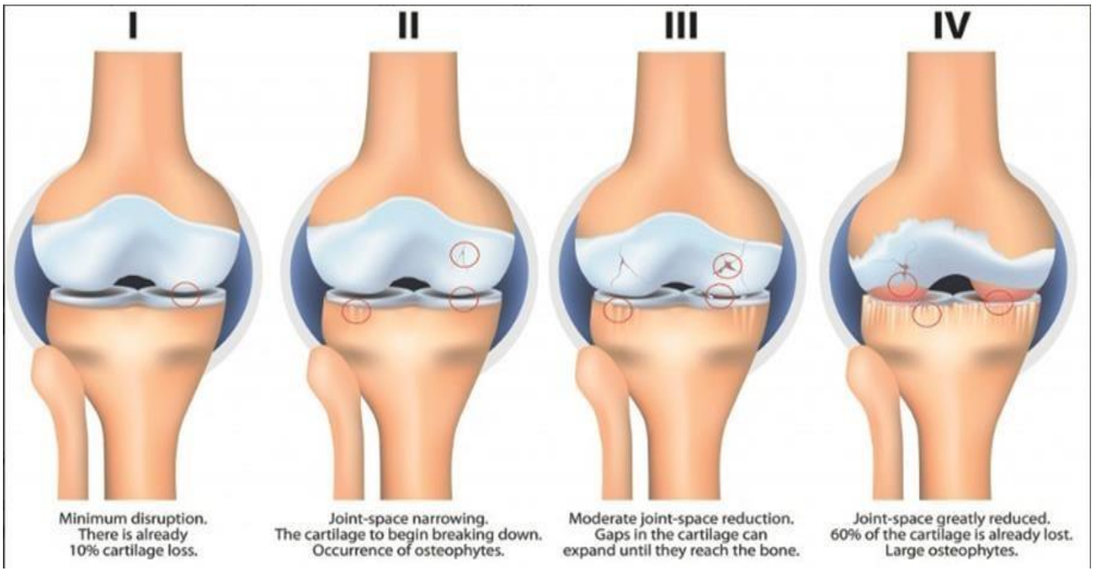


Figure 1.4 Stages of Knee Degenerative Arthritis

STAGE 0 (PRE-OSTEOARTHRITIS)

Stage 0 is considered pre-osteoarthritis (pre-OA) and describes a normal, healthy joint before the disease manifests.

However, this stage can also describe an early stage of OA when damage is beginning to occur on a cellular level, without clinical signs or symptoms.

Symptoms and Signs

You usually wouldn't have any noticeable symptoms or detectable signs of OA during this stage. You may have experienced several healed or healing injuries of one or more of your joints at this stage, or you might be overusing one or more joints.

Changes to the joint lining that may lead to problems later could be happening at this stage.

Diagnosis

The joint changes of pre-OA might not be apparent with imaging tests yet, but it may be possible for pre-OA to be diagnosed with an MRI examination.

Treatment

Treatment of pre-OA will vary and depends on other health factors. Your healthcare provider may recommend over-the-counter (OTC) medications, supplements, and lifestyle changes.

STAGE 1 (EARLY OR DOUBTFUL)

Stage one of OA is considered early or doubtful. You may begin to lose some of the cartilage between your joints. However, the space between your joints wouldn't be getting smaller at this point. You may start to develop bone spurs, which are growths on the ends of the bones.

Symptoms and Signs

Some people do not have any symptoms or signs during stage one. Others may start to experience mild pain in the joints.

Diagnosis

Your healthcare provider may do a physical exam and order an MRI, X-rays, and laboratory tests if there is a concern about your joints.

Treatment

Most people do not seek treatment during stage one because they do not experience any symptoms.

Treatment during stage one is not invasive and focuses on lifestyle changes, supplements, and over-the-counter medications. Lifestyle changes may include exercise, weight loss, yoga, and tai chi.

If you have pain, OTC medications may include nonsteroidal anti-inflammatory drugs (NSAIDs**).**

STAGE 2 (MILD OR MINIMAL)

During stage two of OA, bone spurs grow and become painful. The space between joints may begin to narrow a little. Enzymes can begin to break down the cartilage.

Symptoms and Signs

The symptoms of OA in stage two can vary. Some people may start to experience more pain during activity or after a period of increased activity. You may have trouble bending or straightening the affected joints. Sometimes, the pain and stiffness can impair movement.

Diagnosis

Your healthcare provider may order X-rays to check for bone spurs and other problems. The X-rays may show bone spurs, but the cartilage may continue to look normal. Diagnosis relies on an assessment of your symptoms, a physical exam, and other tests.

Treatment

Your practitioner may recommend OTC medications, such as NSAIDs, for pain. You may also need to make lifestyle changes, like losing weight and doing low-impact exercises.

Other treatment options may include strength training and supplements. You may need to wear a brace, shoe insert, wrap, or knee support.

STAGE 3 (MODERATE)

Stage three of OA is considered moderate, and the cartilage between the bones begins to show signs of wear.

The space between joints becomes visibly narrower. More bone spurs may develop and they can enlarge.

Symptoms and Signs

Most people have frequent pain when moving, walking, or doing other activities that use the joints. Stiffness in the joints may be worse in the morning and after prolonged sitting. Swelling in the joints may also be visible.

Diagnosis

Diagnosis during stage three relies on symptoms and a physical exam. You may also have X-rays and an MRI.Arthroscopy, a minimally invasive procedure, may be used in the diagnosis as well. Diagnostic arthroscopy involves the insertion of a small scope into the joint to examine it.

Treatment

Your healthcare provider may start treatment during stage three with OTC medications, like NSAIDs for pain. If they are not enough, your practitioner may prescribe pain medication for you. You may need hyaluronic acid or corticosteroid injections into the joints for pain relief.

Lifestyle changes, such as losing weight and exercising, continue to be important during stage three. You may also need physical therapy.

STAGE 4 (SEVERE)

The amount of cartilage in the affected joints in stage four is much lower—and in some cases, it may be completely gone. The space between the joints is much smaller, and there is less synovial fluid to lubricate the joints. Bone spurs are much larger.

Severe Osteoarthritis: Risk Factors, Staging, Symptoms

Symptoms and Signs

Most people have a lot of pain when using their affected joints. Daily activities may be difficult or impossible to do. Swelling, and inflammation can also be severe.

Diagnosis

During stage four, diagnosis relies on symptoms, physical exam, lab tests, X-rays, and MRI.

Treatment

By stage four, non-invasive treatments and lifestyle changes may not be enough. Your healthcare provider may recommend an osteotomy or bone realignment surgery to reduce pain. Arthroplasty or knee replacement surgery is another option.

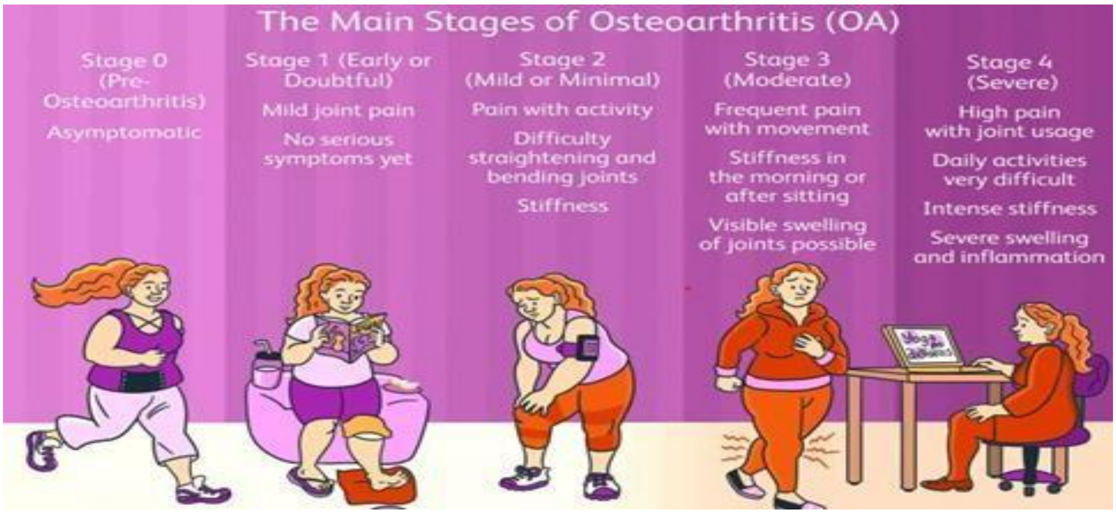


Figure 1.5 Stages and Symptoms of Knee Degenerative Arthritis

DIGITAL IMAGE

Digital Images plays main role in the day-to-day life. The visual effect plays major role than any other media. When we see an image without saying, without explaining anything we understand the concept.

What is an Image?

Visual representation of an object is called as Image. An image is a two dimensional function that represents a measure of some characteristic such as brightness or color of a viewed scene.

Figure 1.6 Sample Image

What is a digital image?

Digital image is composed of a finite number of elements having a particular location and value. These elements are called picture elements, image elements, pels and pixels. A real image can be represented as a two dimensional continuous light intensity function g(x,y) where x and y denote the spatial coordinates and the value of g is proportional to the brightness (or gray level) of the image at that point.

Types of Image

Generally the images can be classified into two types. They are

i. Analog Image

ii. Digital Image

i) Analog Image

The image which is having continuously varying physical quantity in the spatial data such as x, y of the particular axis is known as Analog Image. Analog image can be mathematically represented as a continuous range of values representing position and intensity. The image produced on the screen of a CRT monitor, Television and medical images are analog images.

ii) Digital Image

A digital image is composed of picture elements called pixels with discrete data. Pixels are the smallest sample of an image. A pixel represents the brightness at one point. The common formats of digital images are TIFF, GIF, JPEG, PNG, and Post-Script.

DIGITAL IMAGE PROCESSING

Processing the images using digital computers is termed as Digital Image Processing. Digital image processing concepts are allied in the fields of defence , medical diagnosis, astronomy, archaeology, industry, law enforcement, forensics, remote sensing etc.

Flexibility and Adaptability

Modification in hardware components is not required in order to reprogram digital computers to solve different tasks. This feature makes digital computers an ideal device for processing image signals adaptively.

Data Storage and Transmission

The digital data can be effectively stored since the development of different image compression algorithm is in progress. The digital data can be easily transmitted from one place to another and from one device to another using the computer and its technologies.

Different image processing techniques include image enhancement, image restoration, image fusion and image watermarking for its effective applications.

CAMERA CALIBRATION

In computer vision, camera calibration is crucial to achieve the main goal of calculating the intrinsic parameters and camera distortion parameters. These characteristics are required if photos are used to estimate the pose of objects that are identified or to scan things in 3D. Assuming a pinhole which is used to perform camera calibration. A basic camera design without a lens is the pinhole camera. When light enters the aperture, it reverses the image and projects it on the other side of the camera.

IMAGE PROCESSING

Image processing is a subset of computer vision because a computer vision system employs image processing techniques to simulate human vision at a large scale.

Various mathematical functions are performed in image alterations such as smoothing, sharpening, contrasting, and stretching.

In the detection of a defect through image processing, feature extraction, edge detection, morphological operators, and data training are some of the essential phases.

In order to replace manual inspection, image processing techniques is used to detect defects to reduce time and cost. There are two approaches in image processing, such as analog image processing and digital image processing. For analog image processing, the technique for processing pictures, prints, and other tangible reproductions of images with images as input are performed.

On the other hand, digital image processing entails using complex algorithms to manipulate images or information linked with an image, such as features or bounding boxes digital image to generate information. All in that, defects can be detected faster and more efficiently with the help of image processing. Processing on image can be of three types .They are low-level, mid-level, high level.

Low-level Processing

- Preprocessing to remove noise.
- Contrast enhancement.
- Image sharpening.

Medium Level Processing

- Segmentation.
- Edge detection
- Object extraction.

High Level Processing

- Image analysis.
- Scene interpretation.

Why Image Processing?

Since the digital image is invisible, it must be prepared for viewing on one or more output device (laser printer, monitor).The digital image can be optimized for the application by enhancing the appearance of the structures within it.

There are three of image processing used.

- Image to Image transformation
- Image to Information transformations
- Information to Image transformations

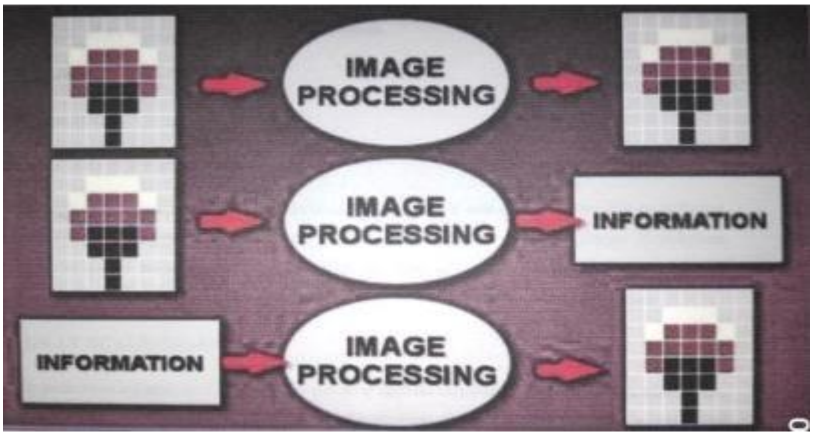


Figure 1.7 Types of Image Processing

PIXEL

Pixel is the smallest element of an image. Each pixel correspond to any one value. In an 8-bit gray scale image, the value of the pixel between 0 and 255.Each pixel store a value proportional to the light intensity at that particular location. It is indicated in either Pixels per inch or Dots per inch.

RESOLUTION

The resolution can be defined in many ways. Such as pixel resolution, spatial resolution, temporal resolution, spectral resolution. In pixel resolution, the term resolution refers to the total number of count of pixels in an digital image. For example, If an image has M rows and N columns, then its resolution can be defined as MX N. Higher is the pixel resolution, the higher is the quality of the image.

Resolution of an image is of generally two types.

- Low Resolution image
- High Resolution

Since high resolution is not a cost effective process It is not always possible to achieve high resolution images with low cost. Hence it is desirable Imaging. In Super Resolution imaging, with the help of certain methods and algorithms we can be able to produce high resolution images from the low resolution image.

GRAY SCALE IMAGE

A gray scale picture is a capacity I (xylem) of the two spatial directions of the picture plane. I(x,y) is the force of picture at the point (x, y) on the picture plane. I (xylem) take non negative expect the picture is limited by a rectangle.

COLOR IMAGE

It can be spoken to by three capacities, R (xylem) for red, G (xylem) for green and B (xylem) for blue. A picture might be persistent as for the x and y facilitates and furthermore in adequacy. Changing over such a picture to advanced shape requires that the directions and the adequacy to be digitized. Digitizing the facilitate's esteems is called inspecting. Digitizing the adequacy esteems is called quantization.

TENSOR FLOW

Tensor flow is an open source software library for high performance numerical computation. It allows simple deployment of computation across a range of platforms (CPUs, GPUs, TPUs) due to its versatile design also from desktops to clusters of servers to mobile and edge devices. Tensor flow was designed and developed by researchers and engineers from the Google Brain team at intervals Google's AI organization, it comes with robust support for machine learning and deep learning and the versatile numerical computation core is used across several alternative scientific domains.

MACHINE LEARNING

Machine learning is the study of computer algorithms that improve automatically through experience. It is a subfield of artificial intelligence (AI) that uses algorithms trained on data sets to create self-learning models that are capable of predicting outcomes and classifying information without human intervention.

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.

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

A machine learning algorithm is an algorithm that can learn from data. It can be used to calculate these rules automatically.

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 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).

Supervised classification

Supervised classification is based on the idea that a user can select sample pixels in an imagethat are representative of specific classes and then direct the image processing software to use these training sites as references for the classification of all other pixels in the image. Training sites (also known as testing sets or input classes) are selected based on the knowledge of the user.

Unsupervised classification

Unsupervised classification is where the outcomes (groupings of pixels with common characteristics) are based on the software analysis of an image without the user providing sample classes. The computer uses techniques to determine which pixels are related and groups them into classes.

COMPUTER VISON

Computer vision is the field of AI that enables computers and machines to make decisions or make recommendations based on important information from pictures, videos and other visuals.

However, computer vision allows the AI to see, monitor and understand if the computer is doing what it needs.

CV works a lot like human vision. However, human vision has some advantage of environmental continuity and teaches how to tell objects piece by piece, how low they are, whether they are moving, and whether the image is false.

By combining cameras, data, and algorithms, computer vision teaches robots to perform these tasks far more quickly than they could with the help of retinas, optical jitter, and the visual brain. As a result of the fact that a system trained to examine items / product assets can examine thousands of products or process nanoseconds, it can quickly surpass mortal capabilities, noting unpleasant blights or problems.

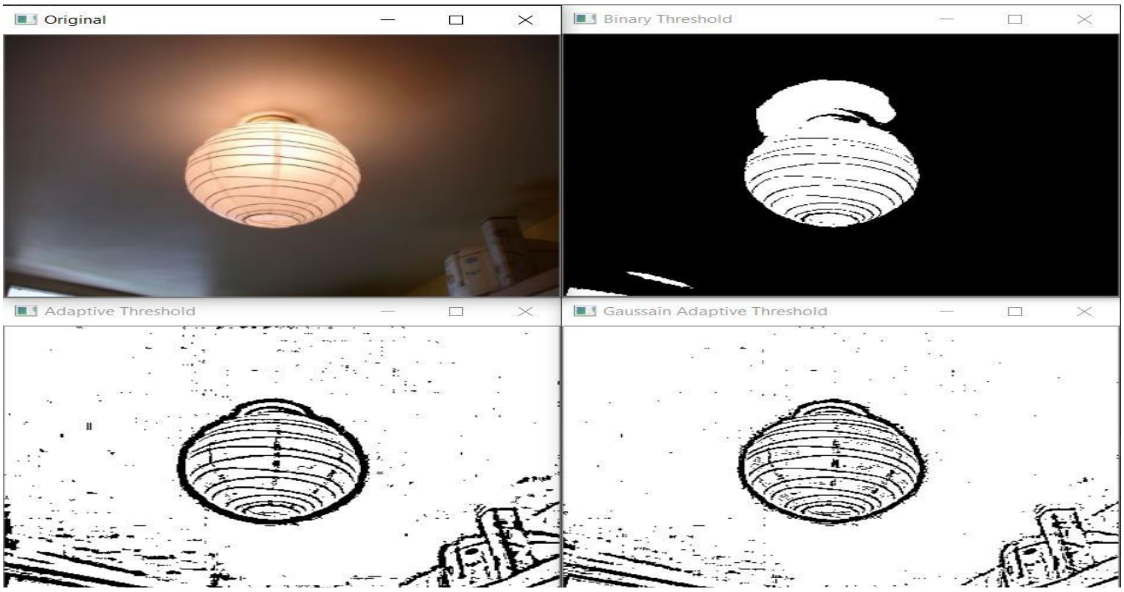


Figure 1.8 Example of Computer Vision

Computer vision requires a lot of data. It continually analyses the data until it can spot the differences and identify the photos.

FEATURE EXTRACTION

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups.So when we want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables.

Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples.

So,Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features. These features are easy to process, but still able to describe the actual data set with accuracy and originality.

IMAGE THRESHOLDING

Image thresholding is a simple, yet effective, way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images.

DEEP LEARNING

Deep Learning is a subset of Machine Learning that involves building and training neural networks with multiple layers. These networks are capable of learning complex patterns and features from large amounts of data, without being explicitly programmed to do so. Deep learning has been applied to a wide range of applications, including image and speech recognition, natural language processing, and autonomous systems.

CONVOLUTIONAL NEURAL NETWORKS (CNNs)

CNNs are a type of deep learning algorithm commonly used in image classification and object detection tasks. CNNs are designed to automatically learn and extract features from images. By applying convolutional filters to them. These filters are learned during the training process and can identify patterns and edges in the input images. CNNs also use pooling layers to reduce the dimensionality of the data and increase computational efficiency. The combination of convolutional and pooling layers enables CNNs to detect complex features in images and make accurate predictions about the contents of an image.

TRANSFER LEARNING

Transfer learning is a technique in deep learning where a pre-trained model is used as a starting point for a new task. In transfer learning, the pre-trained model is fine-tuned on a new dataset that is related to the original dataset used to train the model. This approach can save significant time and resources compared to training a model from scratch, especially when the new dataset is relatively small.

Transfer learning is commonly used in computer vision applications, including image classification and object detection, and can significantly improve the accuracy and efficiency of these models.

DATA AUGMENTATION

Data augmentation is a technique used to increase the size and diversity of a dataset by generating new samples from the existing data. This can include techniques such as cropping, rotating, scaling, and flipping images, as well as adding noise or changing the brightness and contrast of images. Data augmentation is commonly used in deep learning, especially when the size of the training dataset is limited. By generating new samples from existing data, data augmentation can improve the generalization and robustness of deep learning models, making them more effective at handling real-world variations in data.

In our project which uses the CNN algorithm, we may consider applying transfer learning and data augmentation techniques to improve the accuracy and generalization of the model.

By starting with a pre-trained CNN model and fine-tuning it on our specific dataset, we can take advantage of the knowledge and features learned by the pre-trained model, while still adapting it to your specific task. Additionally, by applying data augmentation techniques, we can increase the size and diversity of our dataset, making the model more robust and better able to handle real-world variations in the data.

OPTIMIZATION

Optimization refers to the process of finding the best solution or set of values for a given problem, usually involving mathematical or computational models.

In the context of machine learning, optimization is a crucial step in training models to perform well on new, unseen data. The goal of optimization in machine learning is to find the values of the model's parameters that minimize the error between the predicted output and the actual output. This process is often done using an iterative optimization algorithm, such as gradient descent. Gradient descent is a commonly used optimization algorithm that works by iteratively adjusting the model's parameters in the direction of steepest descent of the loss function.

The loss function measures the difference between the predicted output and the actual output for a given set of input data. The goal of gradient descent is to find the values of the model's parameters that minimize the loss function.

Other optimization algorithms used in machine learning include stochastic gradient descent, Adam optimization, and conjugate gradient descent, among others. These algorithms differ in their approach to adjusting the model's parameters and can be better suited to different types of machine learning problems.

In addition to optimizing the model's parameters, it is also important to consider other factors that can affect the model's performance, such as the choice of hyper parameters, regularization techniques, and the size and quality of the training data.

There are various techniques for optimization in machine learning, including:

1. **Gradient descent**: As mentioned earlier, gradient descent is a commonly used optimization algorithm that works by iteratively adjusting the model's parameters in the direction of steepest descent of the loss function. There are several variants of gradient descent, such as batch gradient descent, stochastic gradient descent, and mini-batch gradient descent, which differ in the amount of training data used in each iteration.

2. **Adam optimization**: Adam optimization is a popular variant of stochastic gradient descent that uses adaptive learning rates for each parameter, based on estimates of the first and second moments of the gradients.

3. **Conjugate gradient descent**: Conjugate gradient descent is an optimization algorithm that iteratively solves a system of linear equations to find the optimal parameters.

These are just a few examples of optimization techniques used in machine learning.

TENSOR FLOW

Tensor flow is an open source software library for high performance numerical computation. It allows simple deployment of computation across a range of platforms (CPUs, GPUs, TPUs) due to its versatile design also from desktops to clusters of servers to mobile and edge devices.

Tensor flow was designed and developed by researchers and engineers from the Google Brain team at intervals Google's AI organization, it comes with robust support for machine learning and deep learning and the versatile numerical computation core is used across several alternative scientific domains.

Google COLAB

The Google Collaboratory ("Colab") is a notebook (like a Jupyter Notebook) where you can run Python code in your Google Drive. You can write text, write code, run that code, and see the output – all in line in the same notebook.

Benefits of Google Colab

- Sharing notebooks is as easy as sharing any Google document. You can also get the app and run the code from your phone.
- You can use the powerful and popular Python language in your Google Drive, and the set-up will take less than five minutes.
- Because Python runs on a server (and not in your local browser or on your local computer) you can easily use it to interact with an online database and analyse data in situations where you need to keep the code private and save successfully in google cloud.

VS Code

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It has gained immense popularity among developers due to its lightweight design, extensive customization options, and powerful features. Visual Studio Code has become a popular choice for developers across different domains, ranging from web development and data science to machine learning and game development, thanks to its versatility, performance, and extensibility.

FLASK

Flask is a web application framework for Python that allows developers to build web applications quickly and efficiently. It provides tools and libraries for handling web requests, routing, and templating, making it easy to create dynamic and interactive websites.

Flask follows the WSGI (Web Server Gateway Interface) standard, making it compatible with various web servers and deployment environments. Flask is commonly used to integrate the frontend (client-side) and backend (server-side) components of web applications.

The backend of a web application built with Flask handles tasks such as processing data, interacting with databases, and managing user authentication. On the other hand, the frontend consists of the user interface elements that users interact with, such as HTML, CSS, and JavaScript.

Flask facilitates the integration of frontend and backend by providing a flexible and lightweight framework for building the server-side components of web applications. Developers can create routes to handle different types of requests (e.g., GET, POST) and render HTML templates to generate dynamic web pages.

Additionally, Flask supports the creation of APIs (Application Programming Interfaces) that allow frontend and backend components to communicate with each other asynchronously.

Overall, Flask simplifies the development process by providing a structured approach to building web applications and enables seamless integration between the frontend and backend components, resulting in a cohesive and interactive user experience.

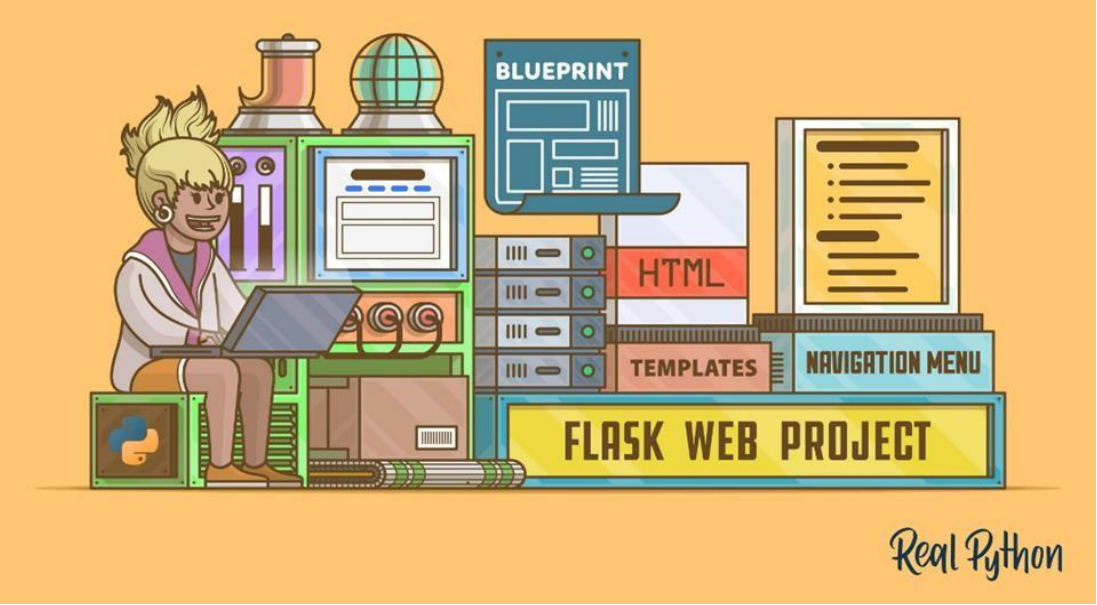


Figure 1.9 Flask Integration

1.2.5 SCOPE OF THE SYSTEM

- Based on the X- rays detecting the Knee Degenerative Arthritis.
- The aim is to apply deep learning in detecting the Knee Degenerative Arthritis and its stage.
- Using various models on the dataset and stick to the model with better accuracy.

1.3 HARDWARE SPECIFICATIONS:

- Processor -I3/Intel Processor
- Hard Disk -160GB
- RAM -8GB
- Key Board -Standard Window Keyboard
- Mouse -Two or Three Button Mouse
- Monitor -Any

1.4 SOFTWARE SPECIFICATIONS:

- Operating System : Windows 7/8/10
- Programming Language : Python
- Libraries : : NumPy, TensorFlow, Matplotlib, OpenCV,
- IDE : VS code
- Technology : :Python 3.5+

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 LITERATURE STUDY

2.1.1 FULLY AUTOMATIC KNEE OSTEOARTHRITIS SEVERITY GRADING USING DEEP NEURAL NETWORKS WITH A NOVEL ORDINAL LOSS

Pingjun Chen, Linlin Gao, Xiaoshuang Shi, Kyle Allen, Lin Yang, 2019

This article discusses the impact of knee osteoarthritis (OA) on older adults, highlighting the importance of early detection and intervention in slowing down the progression of the disease. The current grading system, based on visual inspection, is subject to interpretation and can vary widely depending on the experience of the physician. To address this issue, the article proposes the use of two deep convolutional neural networks (CNNs) to automatically measure the severity of knee OA, as assessed by the Kellgren-Lawrence grading system.

Summary:

The paper discusses the use of a customised YOLOv2 model for detecting knee joints and fine-tuning CNN models with a novel ordinal loss for knee KL grading. The approach achieves state-ofthe-art performance for both knee joint detection and knee KL grading. Furthermore, the proposed ordinal loss helps improve classification accuracy and reduces the mean absolute error (MAE).

2.1.2 AUTOMATIC GENERATION OF MEDICAL IMAGING DIAGNOSTIC REPORT WITH HIERARCHICAL RECURRENT NEURAL NETWORK

Changchang Yin, Buyue Qian, Jishang Wei, Xiaoyu Li, 2019

This article presents a novel framework for accurately detecting abnormalities and automatically generating medical reports. The topic matching system is incorporated into the report generating model, which is based on a HRNN, in order to increase the precision and variety of the reports produced.

Summary:

The paper introduces a new framework that aims to detect diseases and generate medical reports from initial images. The proposed framework includes a GLP mechanism, which outperforms GFP in the abnormality detection experiment.

2.1.3 AUTOMATIC RADIOLOGY REPORT GENERATION BASED ON MULTI-VIEW IMAGE FUSION AND MEDICAL CONCEPT ENRICHMENT.

Jianbo Yuan , Haofu Liao, Rui Luo, and Jiebo Luo, 2020

The generation of radiology reports is a time-consuming process that requires a high level of expertise. To address this, there is a need for radiology report generation to alleviate the workload. While deep learning techniques have been successful in tasks such as image classification and captioning, generating radiology reports is challenging due to the need for understanding and linking complex medical visual content with accurate natural language descriptions.

Summary:

Thanks to modern techniques for training convolutional neural networks, even the most basic architectures can achieve remarkable performance. For instance, networks consisting solely of convolutions and subsampling operations can outperform, or at least match, state-of-theart models on CIFAR-10 and CIFAR-100. A similar architecture can also deliver competitive outcomes on ImageNet. It is noteworthy that contrary to previous findings, including explicit pooling operations like max-pooling doesn't always enhance performance of CNNs. This is especially true when the network is large enough to learn all the necessary invariances using convolutional layers alone for the given dataset.

2.1.4 MACHINE LEARNING IN KNEE OSTEOARTHRITIS: A REVIEW

Kokkotis, C.; Moustakidis, S.; Papageorgiou, E.; Giakas, G.; Tsaopoulos, 2020

The authors of [13] present a review paper focused toward the main aspects of how ML techniques can be used to diagnose and predict KOA. Based on papers published from 2006 to 2019, the survey was divided into four sections: segmentation, optimum post-treatment planning techniques, classification, and predictions/ regression Aspects such as learning techniques, validation, classification, segmentation, etc., were all reviewed and summarized. Their work highlights how ML has played a significant role in the creation of new, automated pre- or post-treatment solutions for KOA.

Summary:

The paper introduces machine learning techniques that have been applied in knee osteoarthritis (KOA) research, primarily using MRI or X-ray data to predict knee cartilage morphology with accuracies ranging from 76.1% to 92%. Feature engineering methods such as Wavelet Packet, PCA, and Histogram techniques are commonly employed to extract relevant information from the data. Various learning algorithms including SVM, CNN ResNet-34, and DNN are utilized, with validation techniques such as 10-fold cross-validation and manual validation are used in the experiment.

2.1.5 A NOVEL METHOD TO PREDICT KNEE OSTEOARTHRITIS PROGRESSION ON MRI USING MACHINE LEARNING METHODS

Yaodong Du, Rania Almajalid, Juan Shan,2020

This study explored the hidden biomedical information from knee MR images for osteoarthritis (OA) prediction. The paper have computed the Cartilage Damage Index (CDI) information from 36 informative locations on tibiofemoral cartilage compartment from 3D MR imaging and used PCA analysis to process the feature set. Four machine learning methods (artificial neural network (ANN), support vector machine (SVM), random forest and naive Bayes) were employed to predict the progression of OA, which was measured by change of Kellgren and Lawrence (KL) grade, Joint Space Narrowing on Medial compartment (JSM) grade and Joint Space Narrowing on Lateral compartment (JSL) grade. To examine the different effect of medial and lateral informative locations, it could be considered to select more points from the medial compartment while reduce the number of points from the lateral compartment to improve clinical CDI design.

Summary:

Machine learning methods were utilized to predict changes in KL, JSM, and JSL grades in knee osteoarthritis by analyzing 36 informative locations on tibiofemoral compartments using PCA. Different classifiers, including ANN, SVM, random forest, and naïve Bayes, were employed, with varying performance across feature sets. Evaluation metrics such as precision, recall, F-measure, MCC, and AUC were used to assess prediction accuracy, the best performance was achieved by random forest with AUC = 0.785 and F-measure = 0.743, while for JSL grade prediction, the best performance was achieved by the ANN with AUC = 0.695 and F measure = 0.796 for JSM grade prediction and 36 features yielding improved accuracy for JSL grade. The percentage of PCA components also influenced classifier performance.

2.2 EXISTING SYSTEM

The existing system discusses a study on the prediction of knee osteoarthritis (KOA) progression using clinical data from the Osteoarthritis Initiative (OAI).The classification process was applied separately to the left and right leg. A 78.3% and 77.7% accuracy were achieved in left and right leg by Logistic Regression. However, Support Vector Machine (SVM) and Multilayer Perceptron (MLP) also performed better.

The system also discusses feature selection methods, including filter, wrapper, and embedded methods. The filter method selected features based on their relevance to the target variable, while the wrapper method used a classification model to drive the feature selection process. The embedded method included a logistic regression classifier to drive the elimination process. The evaluation was conducted with respect to model's overall performance, robustness and highest achieved accuracy. It also highlights the importance of a multi-parametric approach to handle the complexity of the available data.

2.3 PROPOSED SYSTEM

The proposed system uses computer vision and Convolutional Neural Networks (CNN) for image analysis and for report generation.

In this system, pre trained models are used to extract features from knee joint images, enabling the detection and classification of knee images. These extracted features are then processed into models that generate detailed medical reports in a contextually relevant manner.

The computer vision and CNN technology not only accelerates the diagnosis process but also reduces the potential for human error, ultimately providing the quality of care and improving patient outcomes in the management of knee degenerative arthritis. The generated reports are not only accurate but also follow a logical flow. A user interface is created where an image is uploaded and reports are generated with particular detection along with its symptoms and solution.

2.4 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are.

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified.

Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM ANALYSIS AND DESIGN

3. ANALYSIS AND DESIGN

3.1. REQUIREMENTS SPECIFICATION

SOFTWARE REQUIREMENTS

Software Requirements is a field within software engineering that deals with establishing the needs of stakeholders that are to be solved by software. The IEEE Standard Glossary of Software Engineering Terminology defines a requirement as:

1. A condition or capability needed by a user to solve a problem or achieve an objective.
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
3. A documented representation of a condition or capability as in 1 or 2. The activities related to working with software requirements can broadly be broken down into elicitation, analysis, specification, and management.

The software requirements are a description of features and functionalities of the target system. The requirements can be obvious or hidden, known or unknown, expected, or unexpected from the client's point of view. The process to gather the software requirements from clients, analyze and document them is known as software requirement analysis.

Google Colab

Google Colab provides a versatile platform for collaborative coding with built-in Python libraries. Leveraging its integration with Google Drive, it allows seamless access to data and facilitates real-time collaboration. With its extensive library support, including TensorFlow, PyTorch, and scikit-learn, users can easily execute machine learning tasks, conduct data analysis, and develop complex algorithms. The ability to run code directly in the browser without any setup requirements makes it an ideal choice for prototyping and sharing projects. Its interactive environment and free GPU access further enhance its appeal to researchers, students, and developers alike.

VS Code

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It has gained immense popularity among developers due to its lightweight design, extensive customization options, and powerful features. Visual Studio Code has become a popular choice for developers across different domains, ranging from web development and data science to machine learning and game development, thanks to its versatility, performance, and extensibility.

Python

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI). Python is rich in modules. A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use.

Why Python?

- General purpose programming language.
- Increasing popularity for use in data science.
- Easy to build end-to-end products like web applications.
- Since the goal of this project is to build a web application, Python is a better choice.

Matplotlib

Matplotlib is a versatile Python library for creating static, interactive, and animated visualizations in just a few lines of code. It offers a wide range of plotting functions to generate publication-quality plots for scientific and engineering applications.

NumPy

NumPy is a fundamental Python library for numerical computing, providing powerful array operations and mathematical functions. Its efficient handling of large datasets and vectorized operations makes it indispensable for scientific computing and data analysis tasks.

Seaborn

Seaborn is a Python data visualization library built on top of Matplotlib,offering high-level abstractions and stylish statistical graphics. It simplifies the creation of informative and aesthetically pleasing visualizations through its intuitive API and built-in dataset support.

Tensorflow

TensorFlow is a popular open-source machine learning framework developed by Google, designed for building and training neural networks. Its flexible architecture enables easy deployment across various platforms and devices, making it a go-to choice for deep learning projects.

Scikit-learn

Scikit-learn is a powerful Python library for machine learning, providing simple and efficient tools for data mining and analysis. With a wide range of algorithms and model selection utilities, it facilitates rapid development and experimentation in classification, regression, clustering, and more.

OpenCV

OpenCV is a widely-used computer vision library in Python, offering a plethora of tools and algorithms for image and video processing tasks. With its comprehensive functionalities, including object detection, feature extraction, and image manipulation, it's a cornerstone for various computer vision applications.

HTML

HTML is a markup language used to create the structure and content of web pages, using elements and tags to define their layout and functionality. It provides a standardized way to display information on the internet, enabling the creation of interactive and accessible websites.

Cascading Style Sheets (CSS)

CSS, or Cascading Style Sheets, is a stylesheet language used to define the presentation and layout of HTML documents, controlling aspects such as colors, fonts, spacing, and positioning.

By separating content from design, CSS simplifies web development and enhances the consistency and flexibility of web pages across different devices and platforms.

Java Script

JavaScript is a scripting language primarily used for adding interactivity to web pages, handling events, and manipulating the Document Object Model (DOM). It enables dynamic updates and client-side processing, enhancing the functionality and user experience of web applications.

USER REQUIREMENTS:

User requirements play a critical role in the success of a project, especially in domains like healthcare where the end-users are often healthcare professionals and patients.

1. **Access to a compatible device:** The user will need to have a compatible system that meets the website's hardware and software requirements.
2. **Good Network Connection:** The user should have access to proper network connection in order to get faster results.
3. **Properly scanned X-Ray:** The user should be able to provide properly scanned X-Ray in order to receive accurate results.
4. **Proper image format:** User should upload the scanned X-Ray in JPEG or JPG or PNG format only.

3.1.1 FUNCTIONAL REQUIREMENTS

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly the system services in detail. The requirements in requirement engineering help direct the development of the engineered product.

Major Functional Requirements for our work are mentioned below:

Take Image

This function enables the user to take images of a X-Ray.

Process the image

Preprocessing is done to image to remove noise and adjust dimensions and brightness.

Get insights

The insights are drawn from preprocessed image and severity of the disease is determined.

Display reports

The area where cartilage is worn out is highlighted using RGB colors and a graph is displayed, depicting severity levels.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and software requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

Some typical non-functional requirements are:

Usability: The system must be easy to operate. Users who don’t have any experience using such systems must also be comfortable with this system.

Reliability: The reliability of the system essentially depends on the software tools and hardware tools used for the system development.

Performance: The processing speed must be considered for desired performance. There should be an acceptable time delay.

Accuracy: The output of the system provided should be more accurate.

Efficiency: It describes the efficiency of the system in accessing the data.

Major Non-Functional Requirements for our work are mentioned below:

Platform:

Windows version of the software need to be developed.

Safety and Security:

No loss or minimum loss, Privacy and Security of data is major concerned which can be achieved by applying concept of OOPs (such as Encapsulation, Abstraction, and Inheritance).

3.2 SYSTEM ARCHITECTURE

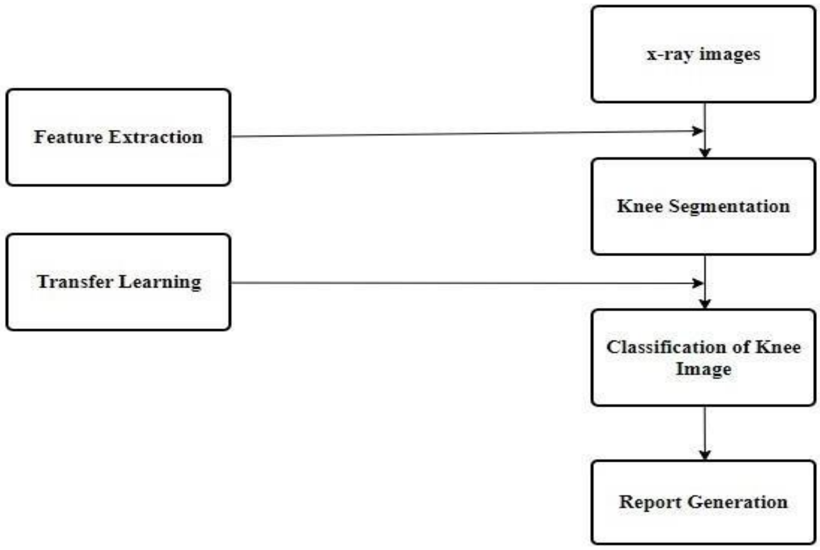


Figure 3.1 System Architecture

3.3 UML DIAGRAMS

Unified Modeling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quitelike blueprints used in other fields of engineering.

UML is linked with object-oriented design and analysis. UML makes the use of elements andforms associations between them to form diagrams. Diagrams in UML can be broadlyclassified as:

1. Structural Diagrams – Capture static aspects or structure of a system. Structural Diagramsinclude Component Diagrams, Object Diagrams, Class Diagrams.

2. Behavior Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modeling language.

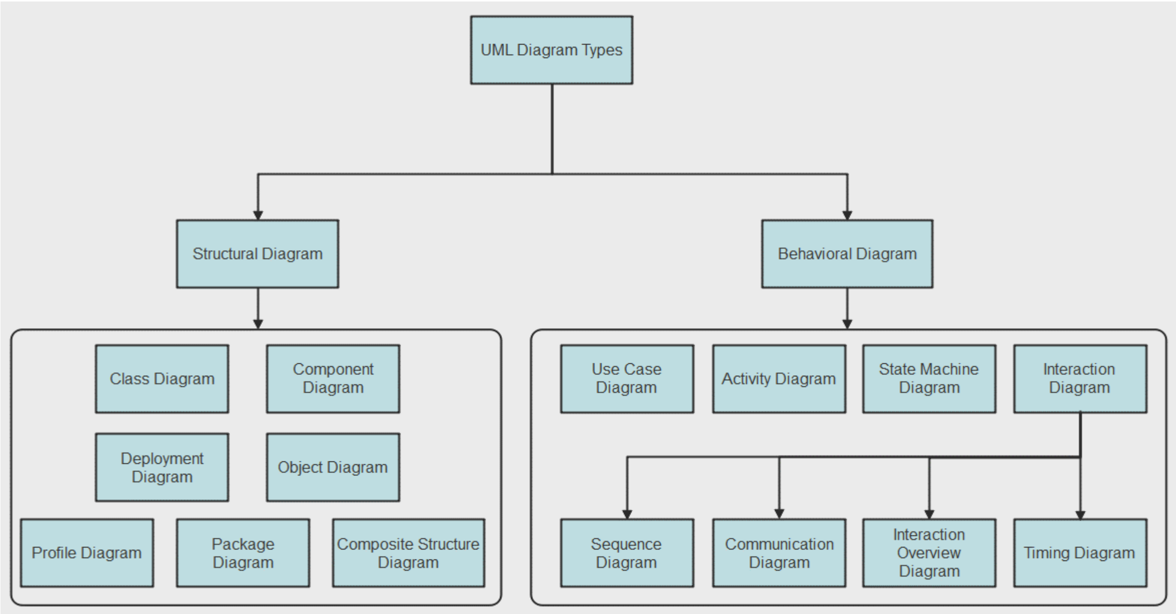


Figure 3.2 UML Diagrams Classification

3.3.1 USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and tells how the user handles a system.

For this system, as shown in the below diagram there will be five use cases and two actors. The use cases are Upload Image, Image Preprocessing , Classification of Knee image or not, Classification of Knee image, Report Generation.

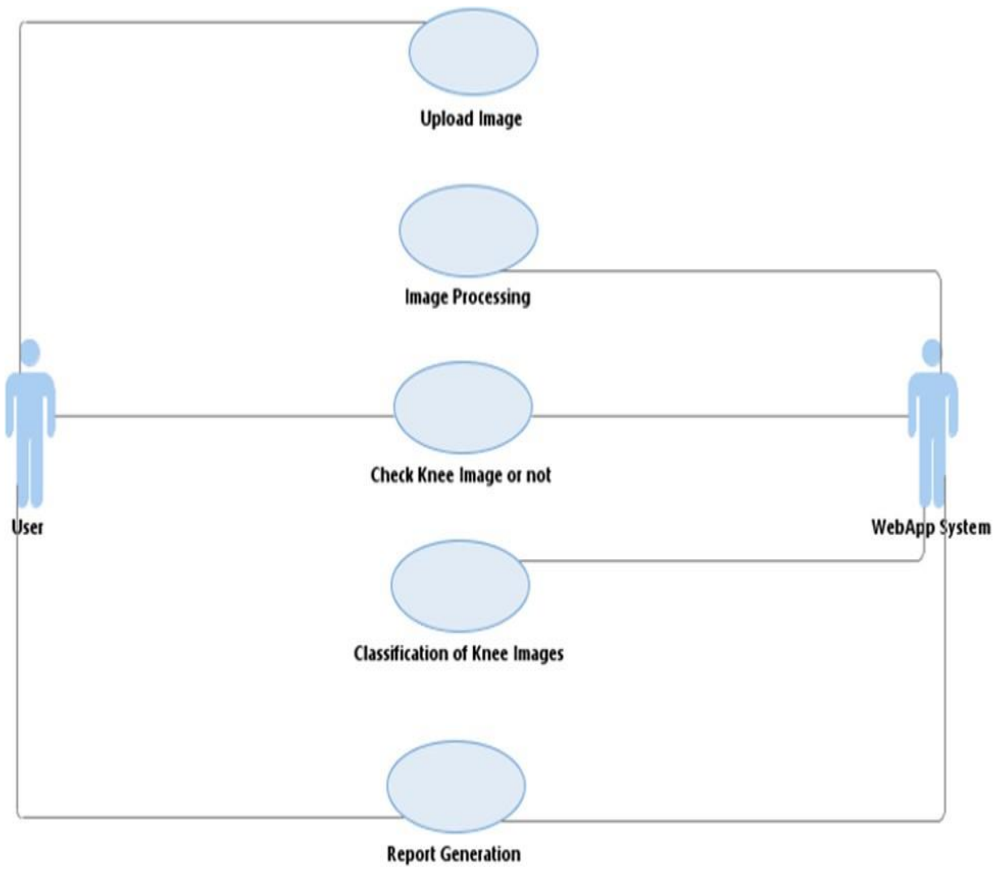


Figure 3.3 Use Case Diagram

Following below are the breakdown of the individual Use Cases. Each showing the Actors, Input and Output respectively:

ACTORS AND THEIR USE CASES

ACTORS: USER, WEB APP SYSTEM

ACTIONS:

Upload Image: The user should upload an image.

Image Processing: The image will be processed with image preprocessing techniques.

Check Knee Image or not: After the processing the image if the image is knee then the image will be further proceeded. If the image is not knee then an error message will be generated.

Classification of Knee Image: If the image is knee then by Transfer Learning the image will be classified into one of the stage.

Report Generation: Finally the report will be generated for the classified image.

3.3.2 CLASS DIAGRAM

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and it may inherit from other classes. A class diagram is used to visualize, describe, document various aspects of the system, and construct executable software code. The class diagram provides a structural overview of the components and their relationships within the knee degenerative arthritis system.

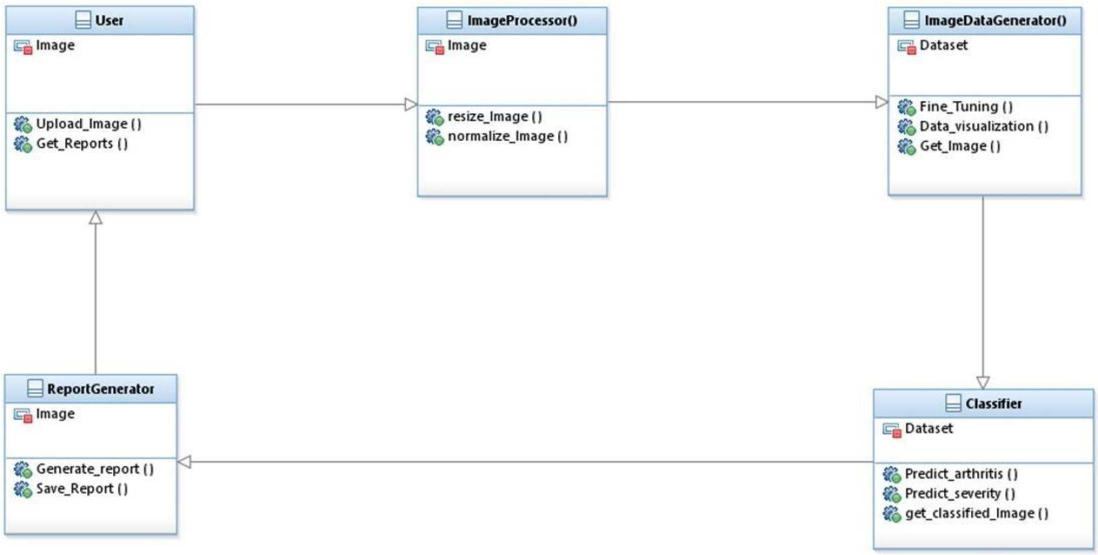


Figure 3.4 Class Diagram

The diagram shows how various classes and the methods are involved. A class diagram for Knee Degeneartive Arthritis System would typically include classes representing key elements of the knee arthritis system. Consider classes like "User","ImageProcessor","ImageDataGenerator", "Classifier", "ReportGenerator". Relationships may include associations between these classes. It's essential capture the key entities and their interactions in the diagram.

3.3.3 SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.

Sequence Diagrams are graphs of relationships which describe how activities are done. Here, the system interacts with the patient. It takes the necessary data from the user/patient, which supplies the system the necessary queries and then sends the output of the generated system.

In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching. A sequence diagram is a type of interaction diagram because it describes how-and in what order a group of objects work together. A sequence diagram for knee degenerative arthritis system would illustrate the interactions and messages exchanged between different components or objects in the system.

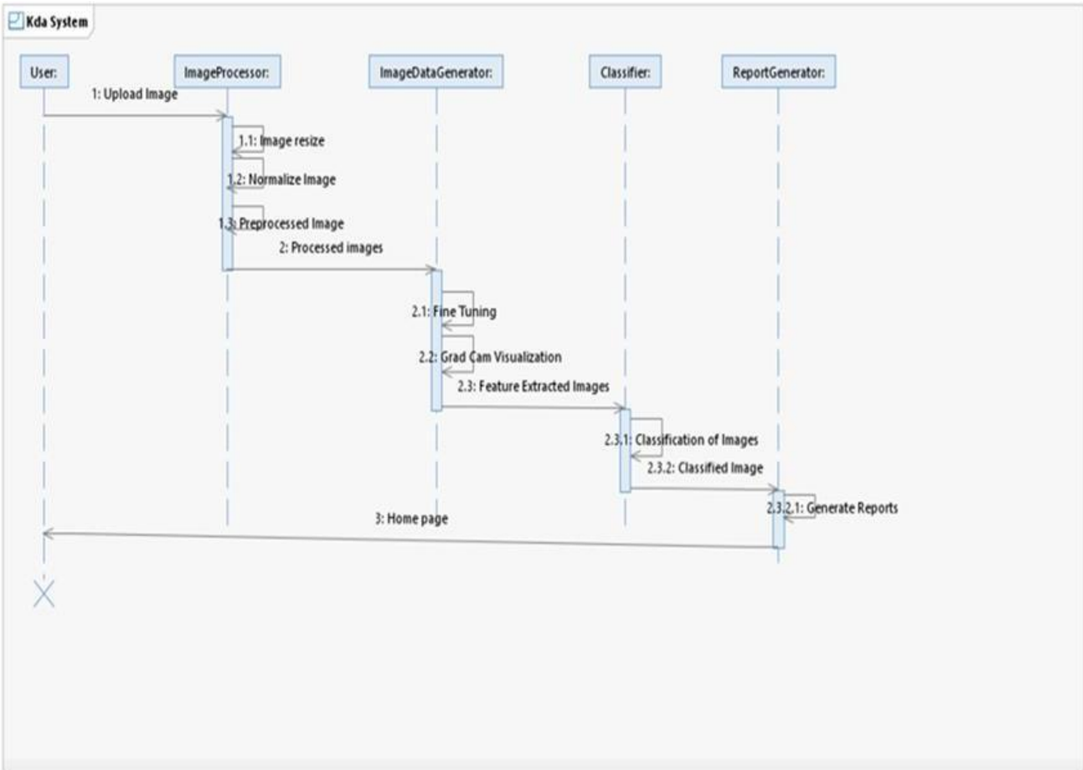


Figure 3.5 Sequence Diagram

3.3.4 COLLABORATION DIAGRAM

Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

NOTATIONS:

1. **Objects**: The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.

2. **Actors**: In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

3. **Links**: The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

4.**Messages**: It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labelled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction.

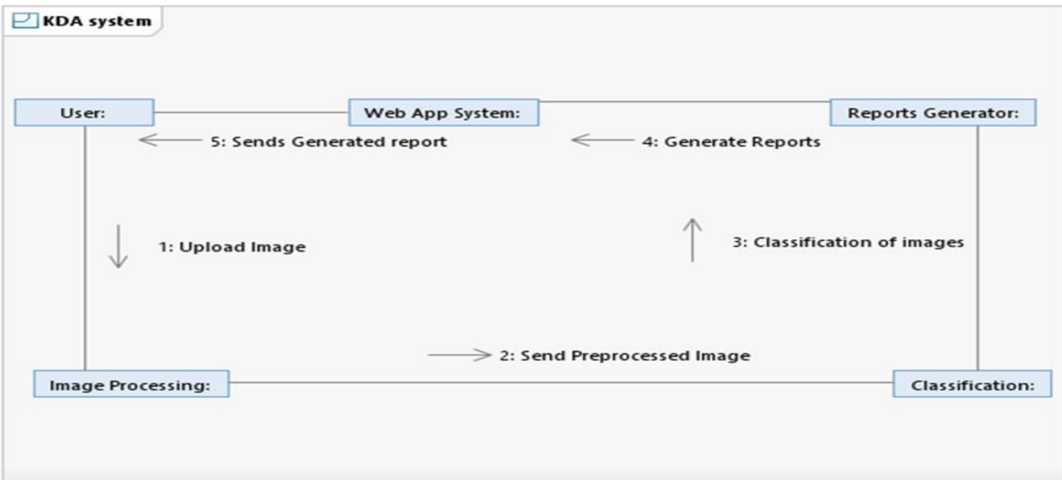


Figure 3.6 Collaboration Diagram

3.3.5 ACTIVITY DIAGRAM

Activity diagram is a type of Unified Modelling Language (UML) flow chart that shows of the system from one activity to another. The diagram outlines the sequential steps and activities involved in the detection process. Arrows would connect these activities to illustrate the flow and decision points might indicate conditions to guide the process based on the results.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. It captures the dynamic behaviour of the system The activity diagram is used to show message flow from one activity to another. It provides a visual representation of the workflow in the knee degenerative arthritis system from uploading image to reporting.

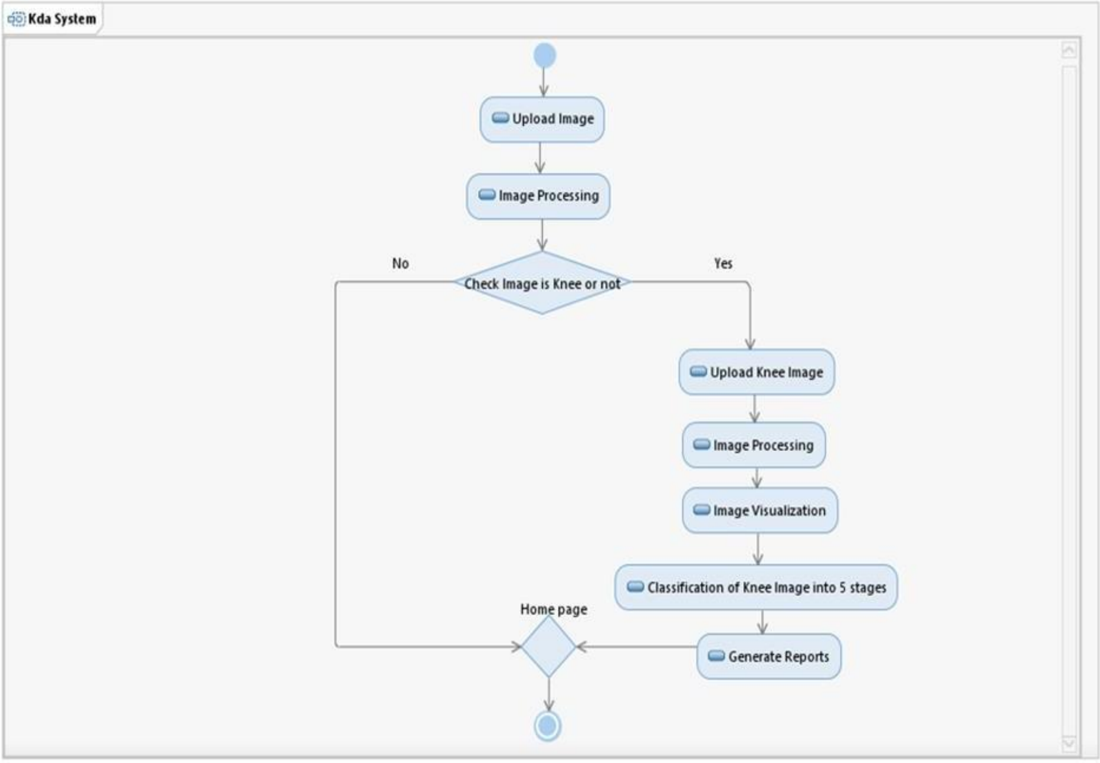


Figure 3.7 Activity Diagram

3.3.6 COMPONENT DIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component diagram describes the organization and writing of the physical components in a system. Arrows would illustrate how these components are connected, showing dependencies and relationships. The diagram provides an overview of the systems structure, key components responsible for knee degenerative arthritis system.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as −

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

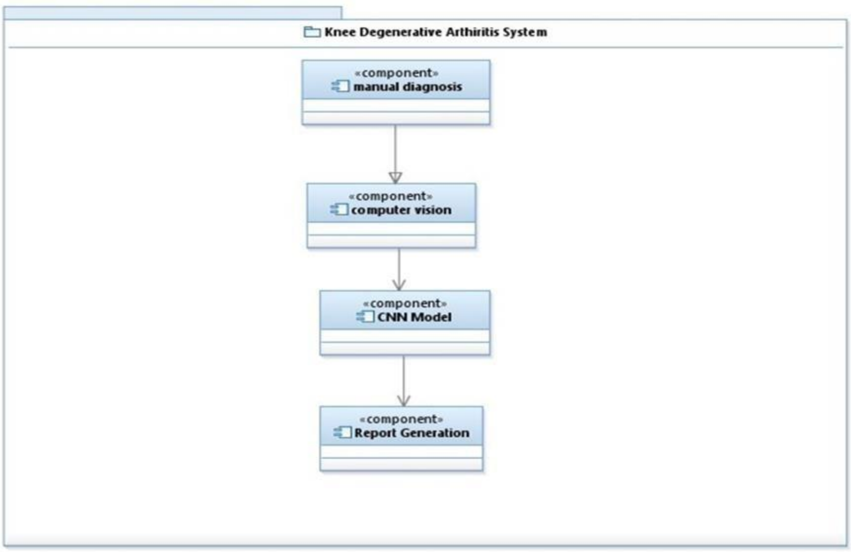


Figure 3.8 Component diagram

3.3.7 DEPLOYMENT DIAGRAM

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths. Both the deployment diagram and the component diagram are closely interrelated to each other as they focus on software and hardware components.

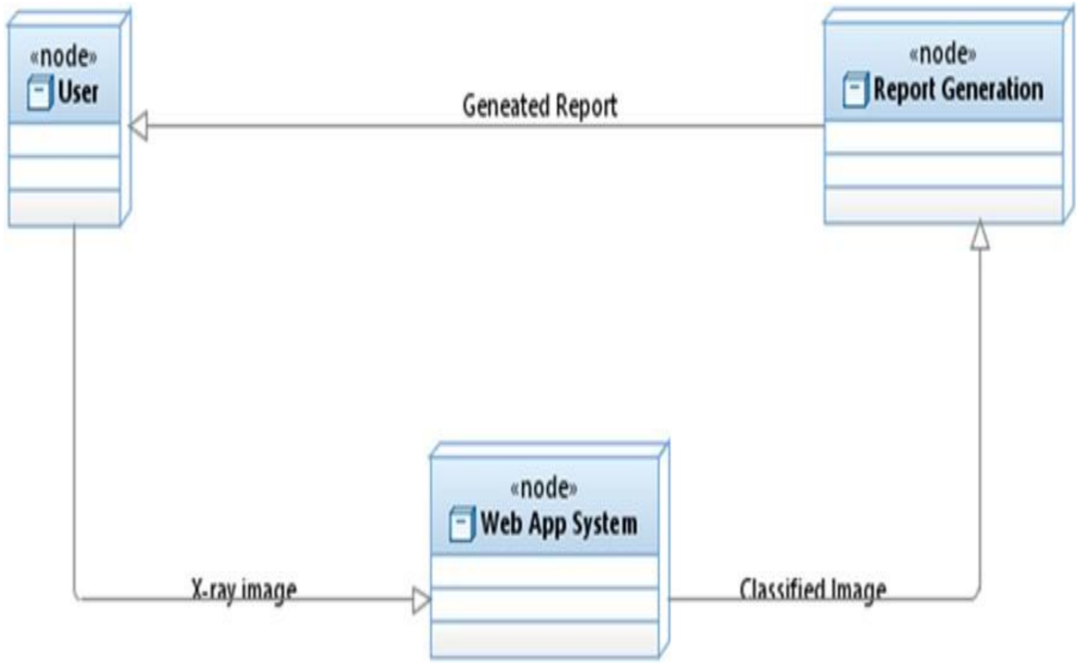


Figure 3.9 Deployment diagram

3.3.8 STATE CHART DIAGRAM

State Chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State Chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists.

The main purposes of using State Chart diagrams

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

A State Chart Diagram for knee degenerative arthritis would typically depict various states and transitions within the system. States might include "Idle", "Upload Image", "Image Preprocessing", "Classification of images ", "Report Generation". Transitions would represent the flow of the process, indicating when the system moves from one state to another based on specific conditions or events. This helps visualize the dynamic behavior of the knee degenerative arthritis system.

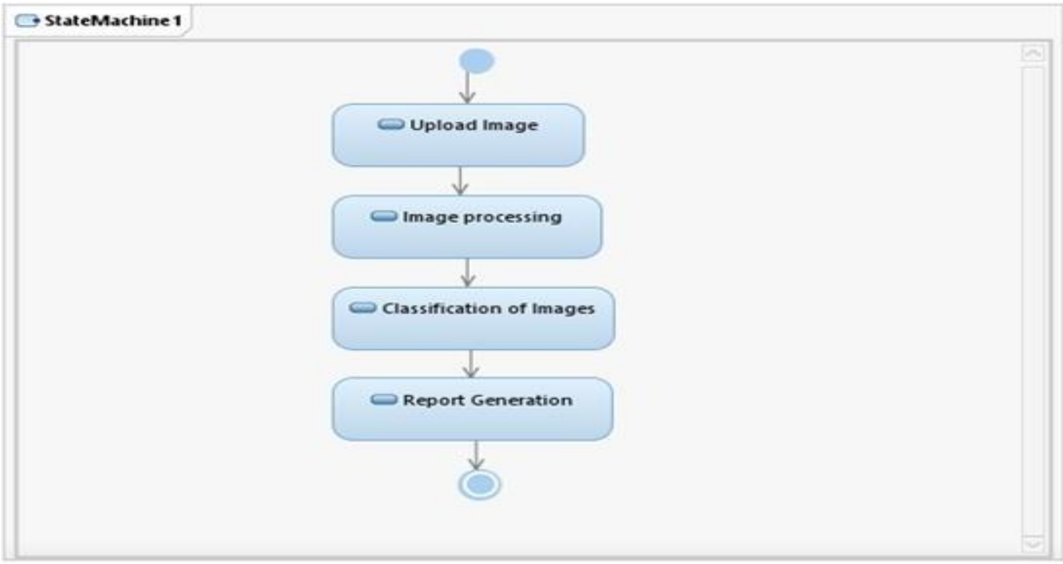


Figure 3.10 State Chart Diagram

3.4 ALGORITHMS

3.4.1 DEEP LEARNING

Deep learning is a machine learning technique. It teaches a computer to filter inputs through layers to learn how to predict and classify information. Observations can be in the form of images, text, or sound. The inspiration for deep learning is the way that the human brain filters information. Its purpose is to mimic how the human brain works to create some real magic. In the human brain, there are about 100 billion neurons. Each neuron connects to about 100,000 of its neighbors. We’re kind of recreating that, but in a way and at a level that works for machines. In our brains, a neuron has a body, dendrites, and an axon. The signal from one neuron travels down the axon and transfers to the dendrites of the next neuron. That connection where the signal passes is called a synapse. Neurons by themselves are kind of useless. But when you have lots of them, they work together to create some serious magic. That’s the idea behind a deep learning algorithm!

You get input from observation and you put your input into one layer. That layer creates an output which in turn becomes the input for the next layer, and so on. This happens over and over until your final output signal! The neuron (node) gets a signal or signals (input values), which pass through the neuron. That neuron delivers the output signal.

Think of the input layer as your senses: the things you see, smell, and feel, for example. These are independent variables for one single observation. This information is broken down into numbers and the bits of binary data that a computer can use. You’ll need to either standardize or normalize these variables so that they’re within the same range. They use many layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output of the previous layer for its input. What they learn forms a hierarchy of concepts.

In this hierarchy, each level learns to transform its input data into a more and more abstract and composite representation. That means that for an image, for example, the input might be a matrix of pixels. The first layer might encode the edges and compose the pixels. The next layer might compose an arrangement of edges. The next layer might encode a nose and eyes. The next layer might recognize that the image contains a face, and so on.

What happens inside the neuron?

The input node takes in information in a numerical form. The information is presented as an activation value where each node is given a number.

The higher the number, the greater the activation. Based on the connection strength (weights) and transfer function, the activation value passes to the next node. Each of the nodes sums the activation values that it receives (it calculates the weighted sum) and modifies that sum based on its transfer function. Next, it applies an activation function. An activation function is a function that's applied to this particular neuron. From that, the neuron understands if it needs to pass along a signal or not.

Each of the synapses gets assigned weights, which are crucial to Artificial Neural Networks (ANNs). Weights are how ANNs learn. By adjusting the weights, the ANN decides to what extent signals get passed along. When you're training your network, you're deciding how the weights are adjusted.

This dynamic starts a learning process in the entire ANN. The way through which the nodes modify themselves is called 'Law of Learning'. The total dynamic of an ANN is tied to time. In fact, for the ANN to modify its own connections, the environment has to necessarily act on the ANN more times[7].

The activation runs through the network until it reaches the output nodes. The output nodes then give us the information in a way that we can understand. Your network will use a cost function to compare the output and the actual expected output. The model performance is evaluated by the cost function. It's expressed as the difference between the actual value and the predicted value.

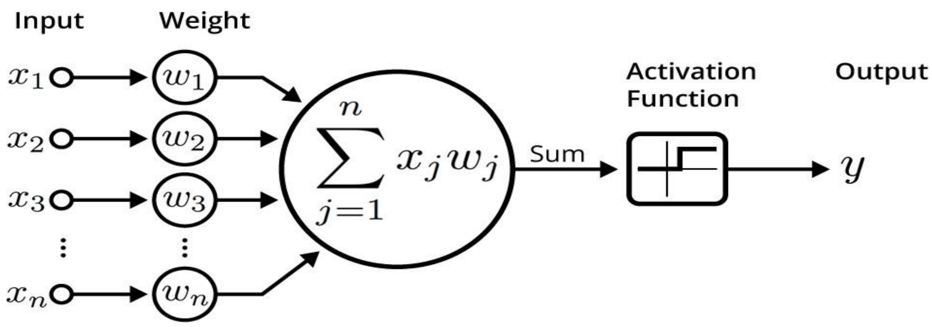


Figure 3.11 Artificial Neuron

There are many different cost functions you can use, you're looking at what the error you have in your network is. You're working to minimize loss function. (In essence, the lower the loss function, the closer it is to your desired output).

The information goes back, and the neural network begins to learn with the goal of minimizing the cost function by tweaking the weights. This process is called backpropagation.

In forward propagation, information is entered into the input layer and propagates forward through the network to get our output values. We compare the values to our expected results. Next, we calculate the errors and propagate the info backward. This allows us to train the network and update the weights. (Backpropagation allows us to adjust all the weights simultaneously.) During this process, because of the way the algorithm is structured, you’re able to adjust all of the weights simultaneously.

This allows you to see which part of the error each of your weights in the neural network is responsible for.

When you’ve adjusted the weights to the optimal level, you’re ready to proceed to the testing phase!

How does an artificial neural network learn?

There are two different approaches to get a program to do what you want. First, there’s the specifically guided and hard-programmed approach. You tell the program exactly what you want it to do. Then there are neural networks. In neural networks, you tell your network the inputs and what you want for the outputs, and then you let it learn on its own.

By allowing the network to learn on its own, you can avoid the necessity of entering in all of the rules. You can create the architecture and then let it go and learn. Once it’s trained up, you can give it a new image and it will be able to distinguish output.

Feedforward and feedback networks

A feedforward network is a network that contains inputs, outputs, and hidden layers. The signals can only travel in one direction (forward). Input data passes into a layer where calculations are performed. Each processing element computes based upon the weighted sum of its inputs. The new values become the new input values that feed the next layer (feed-forward). This continues through all the layers and determines the output. Feedforward networks are often used in, for example, data mining.

A feedback network (for example, a recurrent neural network) has feedback paths. This means that they can have signals traveling in both directions using loops. All possible connections between neurons are allowed. Since loops are present in this type of network, it becomes a non-linear dynamic system which changes continuously until it reaches a state of equilibrium. Feedback networks are often used in optimization problems where the network looks for the best arrangement of interconnected factors.

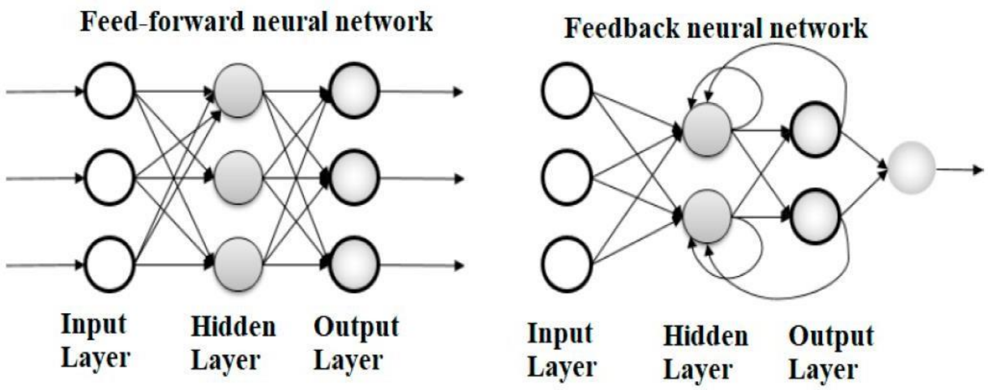


Figure 3.12 Feedforward Vs Feedback Networks

Weighted Sum

Inputs to a neuron can either be features from a training set or outputs from the neurons of a previous layer. Each connection between two neurons has a unique synapse with a unique weight attached. If you want to get from one neuron to the next, you have to travel along the synapse and pay the "toll" (weight). The neuron then applies an activation function to the sum of the weighted inputs from each incoming synapse. It passes the result on to all the neurons in the next layer. When we talk about updating weights in a network, we're talking about adjusting the weights on these synapses.

A neuron's input is the sum of weighted outputs from all the neurons in the previous layer. Each input is multiplied by the weight associated with the synapse connecting the input to the current neuron. If there are 3 inputs or neurons in the previous layer, each neuron in the current layer will have 3 distinct weights: one for each synapse.

Activation function

The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1.

It’s an abstraction that represents the rate of action potential firing in the cell. It’s a number that represents the likelihood that the cell will fire. At it’s simplest, the function is binary: yes (the neuron fires) or no (the neuron doesn’t fire). The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range. If you were using a function that maps a range between 0 and 1 to determine the likelihood that an image is a cat, for example, an output of 0.9 would show a 90% probability that your image is, in fact, a cat.

Threshold function

This is a step function. If the summed value of the input reaches a certain threshold the function passes on 0. If it’s equal to or more than zero, then it would pass on 1. It’s a very rigid, straightforward, yes or no function.

Sigmoid function

This function is used in logistic regression. Unlike the threshold function, it’s a smooth, gradual progression from 0 to 1. It’s useful in the output layer and is used heavily for linear regression.

Hyperbolic Tangent Function

This function is very similar to the sigmoid function. But unlike the sigmoid function which goes from0 to 1, the value goes below zero, from -1 to 1. Even though this isn’t a lot like what happens in a brain, this function gives better results when it comes to training neural networks. Neural networks sometimes get “stuck” during training with the sigmoid function. This happens when there’s a lot of strongly negative input that keeps the output near zero, which messes with the learning process.

Rectifier function

This might be the most popular activation function in the universe of neural networks. It’s the most efficient and biologically plausible.

Even though it has a kink, it’s smooth and gradual after the kink at0. This means, for example, that your output would be either “no” or a percentage of “yes”.

This function doesn’t require normalization or other complicated calculations. The field of artificial intelligence is essential when machines can do tasks that typically require human intelligence. It comes under the layer of ML, where machines can acquire skills and learn from past experience without any involvement of human.

Deep learning comes under machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. The concept of deep learning is based on humans' experiences; the deep learning algorithm would perform a task continuously so that it can improve the outcome.

3.4.2 OPEN CV

OpenCV stands for Open supply pc Vision Library is associate open supply pc vision and machine learning software system library. The purpose of creation of OpenCV was to produce a standard infrastructure for computer vision applications and to accelerate the utilization of machine perception within the business product[8] .

It becomes very easy for businesses to utilize and modify the code with OpenCV as it is a BSD-licensed product. It is a rich wholesome library as it contains 2500 optimized algorithms, which also includes a comprehensive set of both classic and progressive computer vision and machine learning algorithms.

These algorithms is used for various functions such as discover and acknowledging faces. Identify objects classify human actions. In videos, track camera movements, track moving objects.

Extract 3D models of objects, manufacture 3D purpose clouds from stereo cameras, sew pictures along to provide a high-resolution image of a complete scene, find similar pictures from a picture information, remove red eyes from images that are clicked with the flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality.

Officially launched in 1999 the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls.

The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable.

- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.

The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team.

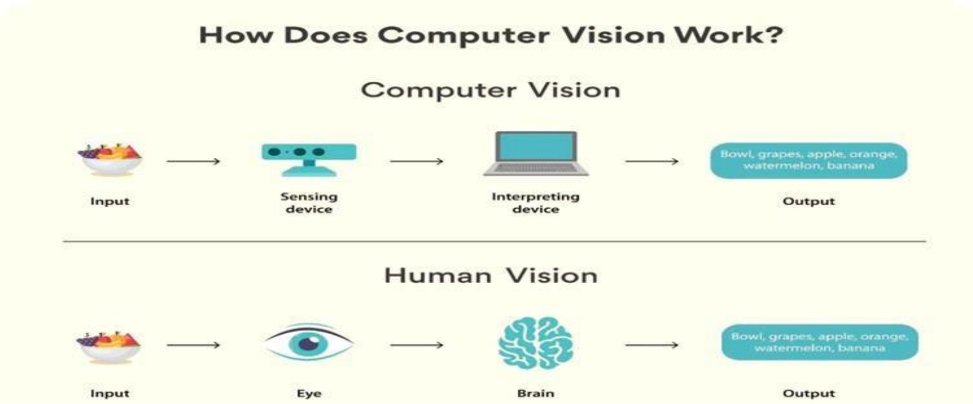


Figure 3.13 Work Flow of Computer Vision

In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.On May 2016, Intel signed an agreement to acquire ITSEEZ, a leading developer of OpenCV. OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real time computer vision.

Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now.

There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

Android developers introduce new application to satisfy the needs of the Smartphone users. Libraries such as OpenGL (Open Graphics Library) and OpenCV (Open Computer Vision) are used for the development of the application [9].

OpenCV's application areas include :

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains :

- Boosting Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearestneighbor algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest

- Support vector machine (SVM)
- Deep neural networks (DNN)

Libraries in OpenCV

Numpy:

NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays. Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier Transform, and random number capabilities.

Numpy Array:

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

SciPy:

SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier transformation and many others.

Matlab :

Python in combination with Numpy, Scipy and Matplotlib can be used as a replacement for MATLAB. The combination of NumPy, SciPy and Matplotlib is a free (meaning both "free" as in "free beer" and "free" as in "freedom") alternative to MATLAB.

Even though MATLAB has a huge number of additional toolboxes available, NumPy has the advantage that Python is a more modern and complete programming language and - as we have said already before - is open source. SciPy adds even more MATLAB-like functionalities to Python.

INTRODUCTION TO CONVOLUTIONAL NEURAL NETWORKS (CNN)

Artificial Neural Networks

The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living neurons and dendrites.

The human brain is composed of 86 billion nerve cells called neurons. They are connected to other thousand cells by Axons. Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward.

ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values.

Neural network

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problem. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be - 1 and 1.

These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship.

3.4.3 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks are very similar to ordinary Neural Networks. They are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity.

Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimize through learning. Each neuron will still receive an input and perform an operation (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire of the network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images.

This allows us to encode image-specific features into the architecture, making the network more suited for image-focused tasks - whilst further reducing the parameters required to set up the model. One of the largest limitations of traditional forms of ANN is that they tend to struggle with the computational complexity required to compute image data.

Common machine learning benchmarking datasets such as the MNIST database of handwritten digits are suitable for most forms of ANN, due to its relatively small image dimensionality of just 28×28. With this dataset a single neuron in the first hidden layer will contain 784 weights ($28\times28\times1$ where 1 bare in mind that MNIST is normalized to just black and white values), which is manageable for most forms of ANN.

If you consider a more substantial color image input of 64 × 64, the number of weights on just a single neuron of the first layer increases substantially to 12, 288. Also take into account that to deal with this scale of input, the network will also need to be a lot larger than one used to classify color-normalized MNIST digits, then you will understand the drawbacks of using such models.

CNN ARCHITECTURE

CNNs are feedforward networks in that information flow takes place in one direction only, from their inputs to their outputs. Just as artificial neural networks (ANN) are biologically inspired, so are CNNs. The visual cortex in the brain, which consists of alternating layers of simple and complex cells (Hubel & Wiesel, 1959, 1962), motivates their architecture.

CNN architectures come in several variations; however, in general, they consist of convolutional and pooling (or subsampling) layers, which are grouped into modules. Either one or more fully connected layers, as in a standard feedforward neural network, follow these modules. Modules are often stacked on top of each other to form a deep model. It illustrates typical CNN architecture for a toy image classification task. An image is input directly to the network, and this is followed by several stages of convolution and pooling. Thereafter, representations from these operations feed one or more fully connected layers.

Finally, the last fully connected layer outputs the class label. Despite this being the most popular base architecture found in the literature, several architecture changes have been proposed in recent years with the objective of improving image classification accuracy or reducing computation costs.

Although for the remainder of this section, we merely fleetingly introduce standard CNN architecture.

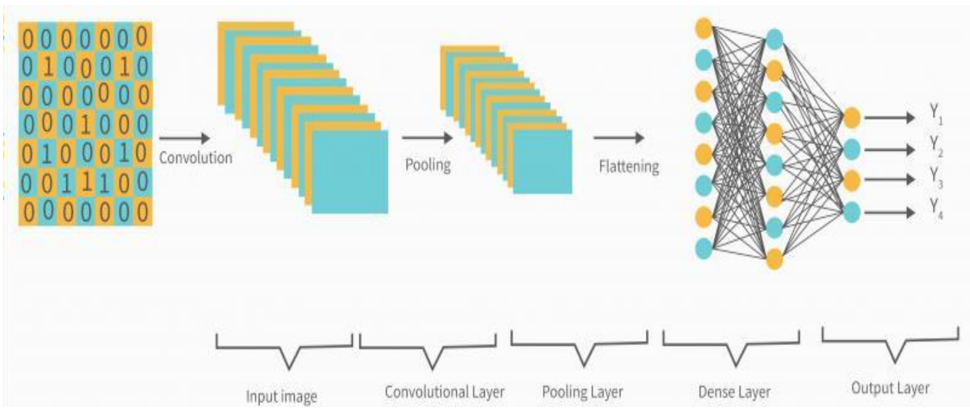


Figure 3.14 CNN Architecture

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.

OVERALL ARCHITECTURE

DL has become an incredibly popular type of ML algorithm in recent years due to the huge growth and evolution of the field of big data[10] .CNNs are comprised of three types of layers. These are convolutional layers, pooling layers and fully connected layers. When these layers are stacked, a CNN architecture has been formed. The basic functionality of the example CNN above can be broken down into four key areas.

1. As found in other forms of ANN, the input layer will hold the pixel values of the image.

2. The convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. The rectified linear unit (commonly shortened to ReLu) aims to apply an ’elementwise’ activation function such as sigmoid to the output of the activation produced by the previous layer.

3. The pooling layer will then simply perform down sampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation.

4. The fully-connected layers will then perform the same duties found in standard ANNs and attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance. Through this simple method of transformation, CNNs are able

to transform the original input layer by layer using convolutional and down sampling techniques to produce class scores for classification and regression purposes.

However, it is important to note that simply understanding the overall architecture of a CNN architecture will not suffice. The creation and optimization of these models can take quite some time, and can be quite confusing.

Advantages of CNN Architecture

- CNN is computationally efficient.
- It performs parameter sharing and uses special convolution and pooling algorithms. CNN models may now run on any device, making them globally appealing.
- It finds the relevant features without the need for human intervention.
- It can be utilized in a variety of industries to execute key tasks such as facial recognition, document analysis, climate comprehension, image recognition, and itemidentification, among others.
- By feeding your data on each level and tuning the CNN a little for a specific purpose,you can extract valuable features from an already trained CNN with its taught weights.

CONVOLUTIONAL LAYERS

The convolutional layers serve as feature extractors, and thus they learn the feature representations of their input images. The neurons in the convolutional layers are arranged into feature maps. Each neuron in a feature map has a receptive field, which is connected to a neighborhood of neurons in the previous layer via a set of trainable weights, sometimes referred to as a filter bank. Inputs are convolved with the learned weights in order to compute a new feature map, and the convolved results are sent through a nonlinear activation function.All neurons within a feature map have weights that are constrained to be equal; however, different feature maps within the same convolutional layer have different weights so that several features can be extracted at each location. The layers parameters focus around the use of learnable kernels.

These kernels are usually small in spatial dimensionality, but spreads along the entirety of the depth of the input. When the data hits a convolutional layer, the layer convolves each filter across the spatial dimensionality of the input to produce a 2D activation map. These activation maps can be visualized.

As we glide through the input, the scalar product is calculated for each value in that kernel. From this the network will learn kernels that ’fire’ when they see a specific feature at a given spatial position of the input. These are commonly known as activations.

The center element of the kernel is placed over the input vector, of which is then calculated and replaced with a weighted sum of itself and any nearby pixels. Every kernel will have a corresponding activation map, of which will be stacked along the depth dimension to form the full output volume from the convolutional layer.

As we alluded to earlier, training ANNs on inputs such as images results in models of which are too big to train effectively. This comes down to the fully connected manner of stan ard ANN neurons, so to mitigate against this every neuron in a convolutional layer is only connected to small region of the input volume. The dimensionality of this region is commonly referred to as the receptive field size of the neuron. The magnitude of the connectivity through the depth is nearly always equal to the depth of the input.

For example, if the input to the network is an image of size $64 \times 64 \times 3$ (a RGB colored image with a dimensionality of 64×64) and we set the receptive field size as 6×6, we would have a total of 108 weights on each neuron within the convolutional layer. ($6 \times 6 \times 3$ where 3 is the magnitude of connectivity across the depth of the volume) To put this into perspective, a standard neuron seen in other forms of ANN would contain 12, 288 weights each.

Convolutional layers are also able to significantly reduce the complexity of the model through the optimization of its output. These are optimized through three hyper parameters, the depth, the stride and setting zero-padding.

The depth of the output volume produced by the convolutional layers can be manually set through the number of neurons within the layer to the same region of the input. This can be seen with other forms of ANNs, where the all of the neurons in the hidden layer are directly connected to every single neuron beforehand. Reducing this hyper parameter can significantly minimize the total number of neurons of the network, but it can also significantly reduce the capabilities of the model.

We are also able to define the stride in which we set the depth around the spatial dimensionality of the input in order to place the receptive field. For example if we were to set a stride as 1, then we would have a heavily overlapped receptive field producing extremely large activations. Setting the stride to a greater number will reduce the amount of overlapping and produce an output of lower spatial dimensions.

Zero-padding is the simple process of padding the border of the input, and is an effective method to give further control as to the dimensionality of the output volumes. It is important to understand that through using these techniques, we will alter the spatial dimensionality of the convolutional layers output.

Despite our best efforts so far we will still find that our models are still enormous if we use an image input of any real dimensionality. However, methods have been developed as to greatly curtail the overall number of parameters within the layer.

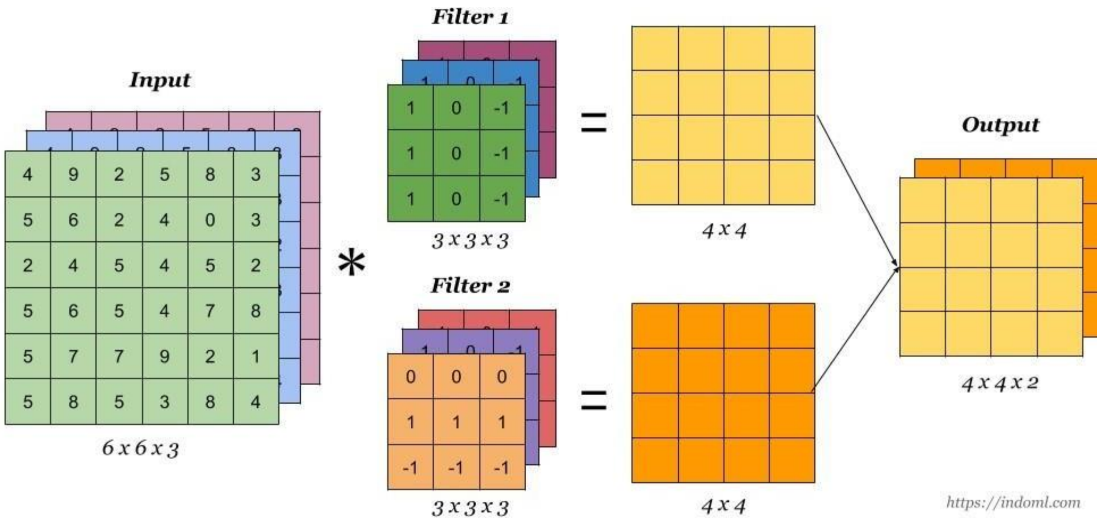


Figure 3.15 Convolutional Layer in CNN

Parameter sharing works on the assumption that if one region feature is useful to compute at a set spatial region, then it is likely to be useful in another region.

If we constrain each individual activation map within the output volume to the same weights and bias, then we will see a massive reduction in the number of parameters being produced by the convolutional layer.As a result of this as the backpropagation stage occurs, each neuron in the output will represent the overall gradient of which can be total across the depth - thus only updating a single set of weights, as opposed to every single one.

Pooling Layers

The purpose of the pooling layers is to reduce the spatial resolution of the feature maps and thus achieve spatial invariance to input distortions and translations.

Initially, it was common practice to use average pooling aggregation layers to propagate the average of all the input values, of a small neighborhood of an image to the next layer. However, in more recent models, max pooling aggregation layers propagate the maximum value within a receptive field to the next layer.

Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model.

The pooling layer operates over each activation map in the input, and scales its dimensionality using the "MAX" function. In most CNNs, these come in the form of max-pooling layers with kernels of a dimensionality of 2 × 2 applied with a stride of 2 along the spatial dimensions of the input. This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size.

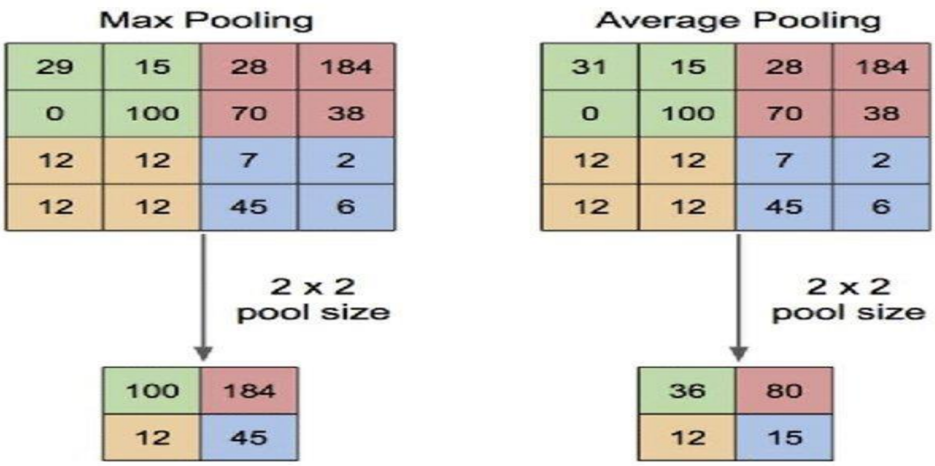


Figure 3.16 Pooling Layer in CNN

Due to the destructive nature of the pooling layer, there are only two generally observed methods of max-pooling. Usually, the stride and filters of the pooling layers are both set to 2 × 2, which will allow the layer to extend through the entirety of the spatial dimensionality of the input. Furthermore overlapping pooling may be utilized, where the stride is set to 2 with a kernel size set to 3. Due to the destructive nature of pooling, having a kernel size above 3 will usually greatly decrease the performance of the model.

It is also important to understand that beyond max-pooling, CNN architectures may contain general pooling. General pooling layers are comprised of pooling neurons that are able to perform a multitude of common operations including L1/L2-normalisation, and average pooling. However, this tutorial will primarily focus on the use of max-pooling.

Flattening Layer:

After several convolutional and pooling layers, the output feature maps are flattened into a one-dimensional vector. The flattening layer essentially reshapes the multi-dimensional feature maps into a format that can be fed into the subsequent fully connected layers for classification. Flattening is a necessary step before passing the features to the fully connected layers, which require inputs in the form of vectors.

The Flatten layer is used and its main purposes:

Transition from Convolutional Layers to Fully Connected Layers: In convolutional neural networks (CNNs), convolutional and pooling layers typically operate on multi-dimensional tensors (e.g., 2D for images). However, before passing data to fully connected layers, which require 1D inputs, we need to flatten the tensor. The Flatten layer serves this purpose by converting a 2D or 3D tensor into a 1D vector.

Simplifying the Data for Dense Layers: Fully connected layers in deep neural networks expect flattened input. By flattening the data, we remove any spatial or temporal structure present in the tensor and convert it into a form that can be processed by densely connected neurons.

Reducing Model Complexity: The Flatten layer reduces the dimensionality of the data, which can help reduce the number of parameters in the subsequent fully connected layers. This is important to manage model complexity and avoid overfitting, especially in deep networks.

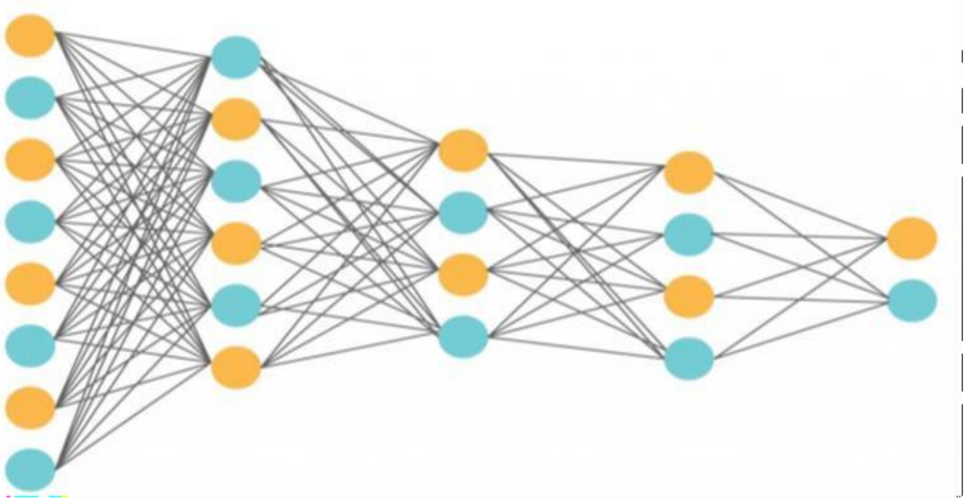


Figure 3.17 Flattening Layer in CNN

Fully Connected Layers

Several convolutional and pooling layers are usually stacked on top of each other to extract more abstract feature representations in moving through the network. The fully connected layers that follow these layers interpret these feature representations and perform the function of high-level reasoning.

For classification problems, it is standard to use the softmax operator on top of a DCNN. While early success was enjoyed by using radial basis functions (RBFs), as the classifier on top of the convolutional towers found that replacing the softmax operator with a SVM leads to improved classification accuracy.

The fully-connected layer contains neurons of which are directly connected to the neurons in the two adjacent layers, without being connected to any layers within them. This is analogous to way that neurons are arranged in traditional forms of ANN.

Despite the relatively small number of layers required to form a CNN, there is no set way of formulating a CNN architecture. That being said, it would be idiotic to simply throw a few of layers together and expect it to work. Through reading of related literature it is obvious that much like other forms of ANNs, CNNs tend to follow a common architecture. This common architecture, where convolutional layers are stacked, followed by pooling layers in a repeated manner before feeding forward to fully-connected layers.

Convolutional Neural Networks differ to other forms of Artificial Neural Network in that instead of focusing on the entirety of the problem domain, knowledge about the specific type of input is exploited. This in turn allows for a much simpler network architecture to be set up. This paper has outlined the basic concepts of Convolutional Neural Networks, explaining the layers required to build one and detailing how best to structure the network in most image analysis tasks.

Research in the field of image analysis using neural networks has somewhat slowed in recent times. This is partly due to the incorrect belief surrounding the level of complexity and knowledge required to begin modelling these superbly powerful machine learning algorithms. The authors hope that this paper has in some way reduced this confusion, and made the field more accessible to beginners.

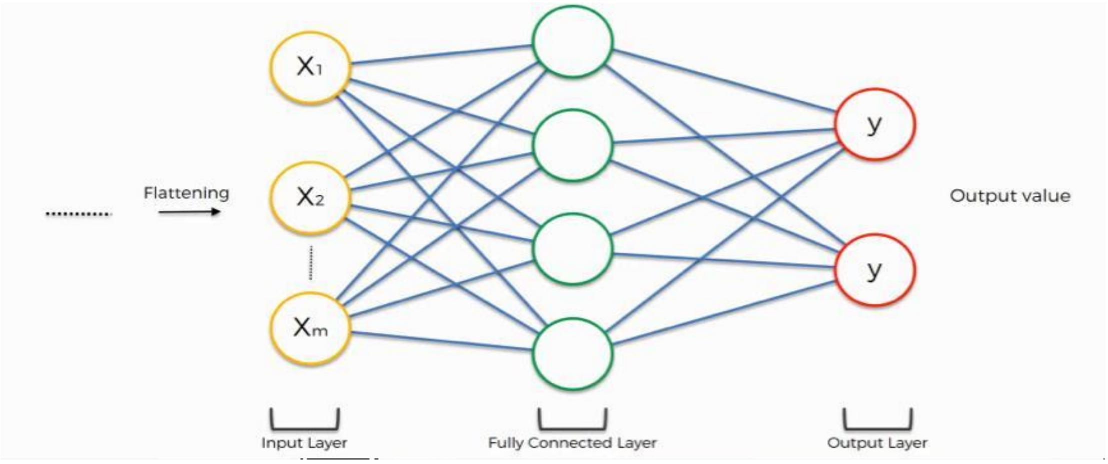


Figure 3.18 Fully Connected Layer in CNN

3.4.4 TRANSFER LEARNING

Transfer Learning is a machine learning technique in which a previously trained model is reused as the basis for a new model on a different task. This optimization method allows rapid progress while modelling the new task. As such, most researchers and data scientists prefer to begin with a pre-trained model that already knows how to identify objects and has learned general features such as edges.

Since the model will be trained with datasets, a significant amount of time and resources will be required only to train the massive parameters. As a result, transfer learning has become one of the most popular methods for building a deep learning model in the real world, as it only requires training a few final classification layers on top of a base pre-trained model, saving time and data. Hence, transfer learning is said to be a great technique for achieving cutting-edge results and are needed in many applications nowadays.

Traditional Machine Learning vs Transfer Learning

Deep learning experts introduced transfer learning to overcome the limitations of traditional machine learning models. Let's have a look at the differences between the two types of learning. TL attempts to provide an efficient manner of learning and communication between the source task and the target task[11].

- Traditional machine learning models require training from scratch, which is computationally expensive and requires a large amount of data to achieve high performance. On the other hand, transfer learning is computationally efficient and helps achieve better results using a small data set.

- Traditional ML has an isolated training approach where each model is independently trained for a specific purpose, without any dependency on past knowledge. Contrary to that, transfer learning uses knowledge acquired from the pre-trained model to proceed with the task. To paint a better picture of it: One can not use the pre-trained model of ImageNet with biomedical images because ImageNet does not contain images belonging to the biomedical field.
- Transfer learning models achieve optimal performance faster than the traditional ML models. It is because the models that leverage knowledge (features, weights, etc.) from previously trained models already understand the features. It makes it faster than training neural networks from scratch.

Pre-Trained CNN Models

A pre-trained model is one that has already been trained to handle a similar problem by someone else. Instead of starting from the beginning to tackle a comparable problem, a model that has previously been trained on another problem might be utilised to begin the learning process. Even if a pre-trained model is not 100% correct, it saves time and effort over having to start from scratch (Analytics Vidhya, 2017).

There are lots of network models to be used to train the computer to identify objects in an image. There is a competition held every year from 2010 until 2016, ImageNet Large Scale Visual Recognition Challenge (ILSVRC), hosted by ImageNet. The purpose of the competition is to challenge the network models from any organization. The winner of the competition will be announced as the best network model for that year without any argument [12].

For instance, by comparing the pre-trained deep learning models with custom built CNN, employing a pre-trained model for various picture recognition tasks is advantageous. By adopting a pre-trained model, less training and work are required in constructing the model's architecture. Another advantage is the precision, using a pretrained model rather than a custom-built convolutional neural network greatly improves the accuracy.

There are several pre-trained models to be selected such as LeNet, AlexNet, VGG, GoogLeNet, Inception V3, Inception BN and so on in the ImageNet Large

Scale Visual Recognition Competition.

3.4.4.1 INCEPTIONV3

Inception v3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for GoogLeNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge.

The design of Inceptionv3 was intended to allow deeper networks while also keeping the number of parameters from growing too large: it has "under 25 million parameters", compared against 60 million for AlexNet. Just as ImageNet can be thought of as a database of classified visual objects, Inception helps classification of objects in the world of computer vision. All our evaluations are done on the 48238 non-blacklisted examples on the ILSVRC-2012 validation set, as suggested by Imagenet[13].

The Inceptionv3 architecture has been reused in many different applications, often used "pre-trained" from ImageNet. One such use is in life sciences. Inception v3 is an image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model is the culmination of many ideas developed by multiple researchers over the years which does not provide a clear description about the contributing factors that lead to the various design decisions of the GoogLeNet architecture. This makes it much harder to adapt it to new use-cases while maintaining its efficiency[14].

The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using Softmax.

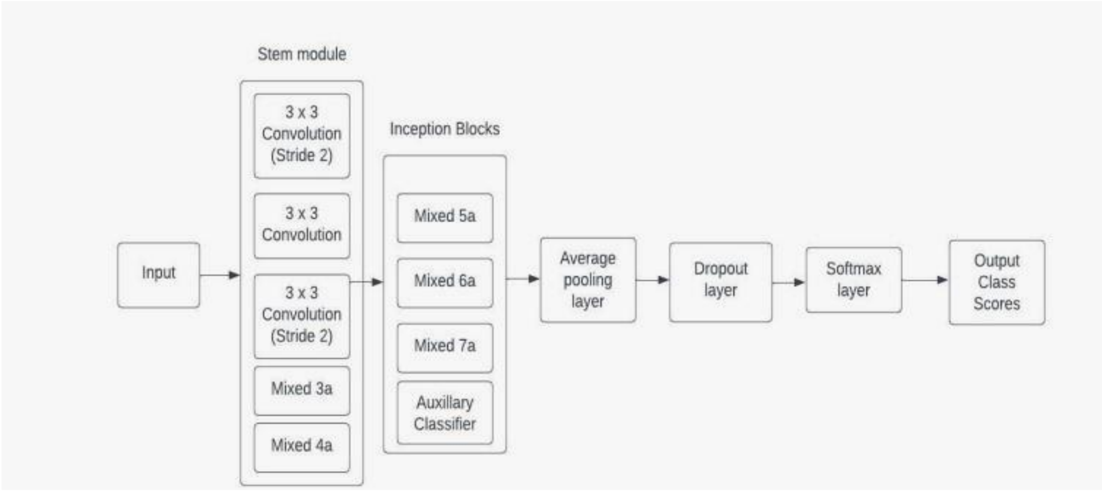


Figure 3.19 Architecture of InceptionV3

ARCHITECTURE OF INCEPTIONV3

- **Input layer**: The layer that receives and transmits the raw image data to the following layer.
- **Factorized convolutions**:InceptionV3 substitutes more computationally effective factorized convolutions for conventional convolutions. The typical convolution is divided into two smaller convolutions by factorized convolutions, one along the input’s width and the other along its height. As a result, the network’s parameters are reduced, which speeds up network training and inference.
- **Inception Modules**: The foundation of the InceptionV3 architecture is made up of Inception modules. They include a number of 1x1 convolutional layers, pooling layers, and parallel convolutional layers of various sizes. These modules are designed to collect features at various scales and merge them into a single layer.
- **Better pooling**: InceptionV3 substitutes ”global average pooling” for the more conventional ”max pooling” method. Global average pooling minimizes overfitting and is more resistant to slight input fluctuations. Moreover , number od parameters in network is reduced.
- **Auxiliary classifiers**: InceptionV3 has two auxiliary classifiers that are coupled to the network’s intermediate layers. In order to increase overall accuracy and decrease overfitting, these auxiliary classifiers give the network more training signals during training.

- **Batch normalization**: The batch normalization layers in InceptionV3 serve to stabilize the learning process and lessen the network's sensitivity to the initial weight values.
- **stem network**: The architecture of InceptionV3 includes a "stem network" that is intended to preprocess input images and extract basic features. This aids in decreasing the amount of parameters and enhancing network performance.
- **Reduction layers**: InceptionV3 has "reduction layers" that are used to increase the number of channels while decreasing the spatial dimensions of the feature maps. This aids in lowering the network's computational complexity and raising performance.
- **Fully connected layers**: The final classification of the input image is carried out by the fully connected layers at the end of the network. Two fully connected layers, one with 2048 units and the other with 1000 units are present.
- **Softmax layer**: The softmax layer creates a probability distribution over the classes using the output of the final fully linked layer.

3.4.4.2 XCEPTION

Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. It was developed by Google researchers. Google presented an interpretation of Inception modules in convolutional neural networks as being an intermediate step in-between regular convolution and the depthwise separable convolution operation (a depthwise convolution followed by a pointwise convolution).

In this light, a depthwise separable convolution can be understood as an Inception module with a maximally large number of towers. This observation leads them to propose a novel deep convolutional neural network architecture inspired by Inception, where modules have been replaced with depthwise separable convolutions.

We first go through the Inception V3:

The basic idea of Inception V3 is: input the data into different convolution structures at the same time to extract features and then concat. The Inception V3 changes the 5*5 kernel into two 3*3 kernels. This neural network can be simplified. In figure 2, using a 1*1 kernel as the first layer and then connect to 3*3 kernels. Let's consider an extreme condition. When the number of 3*3 convolution equals to the output channel of the 1*1 layer.

XCeption is an efficient architecture that relies on two main points :

- Depthwise Separable Convolution
- Shortcuts between Convolution blocks as in ResNet

The depthwise separable convolution is so named because it deals not just with the spatial dimensions, but with the depth dimension — the number of channels — as well. An input image may have 3 channels: RGB. After a few convolutions, an image may have multiple channels. You can image each channel as a particular interpretation of that image; in for example, the “red” channel interprets the “redness” of each pixel, the “blue” channel interprets the “blueness” of each pixel, and the “green” channel interprets the “greenness” of each pixel. An image with 64 channels has 64 different interpretations of that image.

The primary process of depthwise separable convolution is: divide the traditional convolution neural network into two steps. Assume the original kernel is 3*3. Then for depthwise separable convolution, it will first use M (the number of input channels) 3*3 kernels to filter each input channel. It gets M output. Next, using N (the number of input) 1*1 kernels to filter the M outputs and get N outputs. These two steps are called depthwise convolution and pointwise convolution, respectively. Thus, the structure of the Xception . The sparsableConv is the depthwise separable convolution. The mark ‘+’ means residual connection. A Strictly Equivalent Reformulation of the Simplified Inception Module.

XCEPTION ARCHITECTURE

The data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Note that all Convolution and Separable Convolution layers are followed by batch normalization.

Xception is based entirely on depthwise separable convolution layers. The Xception architecture has 36 convolutional layers forming the feature extraction base of the network.). The 36 convolutional layers are structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules. In short, the Xception architecture is a linear stack of depthwise separable convolution layers with residual connections. This makes the architecture very easy to define and modify[15].

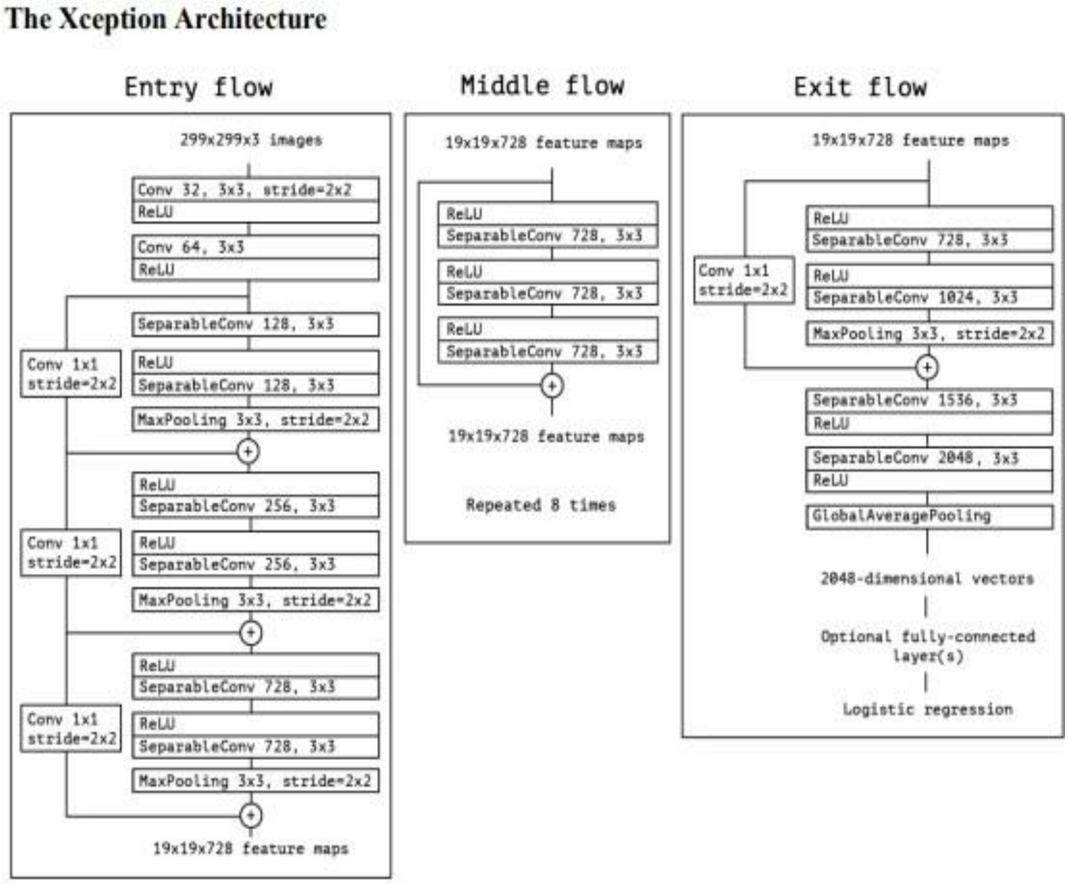


Figure 3.20 Architecture of Xception

The Xception is a very deep and complicated CNN with 36 convolutional layers. The input should be 299*299 images. To run it fully, we need a laptop with a great GPU. And finally, we chose to use Google Colab, a browser that can run Python code. It has a powerful graphics processing unit (GPU) and significantly improves the program speed.

Methodology

This chapter describes the methodology that we have used for the classification and detection of OA affected region in knee X-ray. 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 post processing works for the development of this project are addressed in order in this chapter.

Classification Methodology

In this project we are solving an image classification problem using deep learning 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 X-ray and OA affected knee X-ray where the OA affected knee X-ray are further classified into multi stage classification. The way we are going to achieve this classification is by training an convolution 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 or a OA affected knee .

A simple work flow of classification model is shown in the prediction of input X-ray image after training the neural network on the image datasets of OA affected knee X-ray and normal knee X-ray.

In this project we are using TensorFlow, an open source framework for image classification related tasks, for training and testing with Inception V3 and Xception- models. The model is tested as pre-trained and with fine-tuning with a dataset consisting of OA knee X-ray images.

TensorFlow is an open source framework that aims to make it easy to design, train and implement the images. To achieve this, TensorFlow provides the user with multiple pre-trained models with instructions and example codes for fine-tuning and using the models for image classification tasks. TensorFlow can be used with different pre-trained models.

Setup

The workstation included an Intel i7 processor, 8GB of RAM and the computer ran Ubuntu 16.04 As the primary programming language Python 3 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 Google Colab 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. SikitLearn is a Python libraries. It includes many tools for image processing and is especially popular in the medical field. Other libraries were also used for smaller tasks.

Data set

The data set is a collection of Knee X-ray. The number of available samples grew during the project. For most of the development time, it included data samples that came from multiple X-ray sources. And also downloaded from Kaggle.. All images were provided in the png file format, which is very common in the medical field. It features technical meta information about each image. For our classification problem we have used 8260 samples of knee x-ray and for our OA Binary classification we have used 3836 samples.

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.

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). 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

Augmentation

Image augmentation is a popular approach to virtually increase the size of the dataset. The general idea is that a neural network will overfit more when learning onesample 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. Common choices include:

- Horizontal shifts
- Width shifts
- Rotations
- Shear mapping

- Brightness adjustments
- Contrast 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 24pixels 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.

Binary Classification Work Flow

The work flow of the model for classification of Knee Images is given.First we have prepared the data sets by selecting the X-ray of OA affected knee and X-ray 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 × n pixel and read as grayscale.

The data sets are splitted as 2350 images for training purpose and 845 images for testing and then data sets are fine tuned to build the model. After building the model, the loss function is calculated generally we take the binary cross entropy loss function for training deep learning models which as optimizer sets learning rate to 2e-5. The model is then trained for n epochs and finally the prediction of the input image is achieved as a OA affected knee X-ray or a normal knee X-ray. The X-ray datasets are labelled based on two different classes. One class contains the normal knee X-ray images and the other class contains the OA affected knee X-ray images.

Multi stage Classification Work Flow

The work flow of the model for classification of Knee Images is given . First we have prepared the data sets by selecting the X-ray of OA affected knee with 4 stages and X-ray 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 × n pixel and read as grayscale. The data sets are splitted as 5605 images for training purpose and 1017 images for testing and then data sets are fine tuned to build the model. After building the model, the loss function is calculated generally we take the categorial cross entropy loss function 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 X-ray with 4 stages or a normal knee X-ray.

The X-ray datasets are labelled based on five different classes. One class contains the normal knee X-ray images and the other classes contains the OA affected knee X-ray images.

Image Transformation

All the Knee X-ray images of the datasets are in different pixel values. So, the image datasets with different pixel values are resized to n × n pixel values and made all images with similar pixel values.

CNN based modern neural network model

A simple deep learning based classification model, 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 with pre trained models like Inception V3 and Xception. 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 X-ray and OA affected knee X-ray and also multi stage classification of Knee Degenerative Arthritis.

The way we are going to achieve this classification by training a convolution 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 any one of the stage.

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.

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.

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.

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 =T P + T N / (T P + T N + F P + F N)

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.

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.

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.

GRAD-CAM Visualization

Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. For CNNs, a well-established ex-post hoc explainable technique is used. Furthermore, the Grad-CAM method passes an independent sanity check [16].

The technique is an improvement over previous approaches in versatility and accuracy . From a high-level, we take an image as input and create a model that is cut off at the layer for which we want to create a Grad-CAM heat-map. We attach the fully-connected layers for prediction. We then run the input through the model, grab the layer output, and loss. Next, we find the gradient of the output of our desired model layer w.r.t. the model loss. From there, we take sections of the gradient which contribute to the prediction, reduce, resize, and rescale so that the heat-map can be overlaid with the original image.

We recommend using optimized CNN models to include Grad-CAM visualization into the knee OA severity classification process. Following the CNN model's training on knee images labeled with severity grades, we use Grad-CAM to produce heatmaps that highlight the input images with particular regions.

3.5 ALGORITHM STEPS IN PSEUDO CODE:

1. Input: Images = {I1, I2, I3, ..Ik}
2. Output: Classified Images
3. Start
4. data(i) ← 1. ... k
5. While (data(i)!= eof)
6. Begin
7. BalancedDataset ← UnbalancedDataset D Data Augmentation
8. Images Pre-Processing Ik (alter format, downscaling, negativity)
9. Transfer Learning ← Extraction of Features D Fine Tuning
10. Binary Classification ← Inception V3
11. 5 stage Classification ← Xception
12. Grade wise Knee Degenerative Arthritis classification
13. Generate Reports
14. End
15. Finish

3.6 FLOW CHART OF ALGORITHM

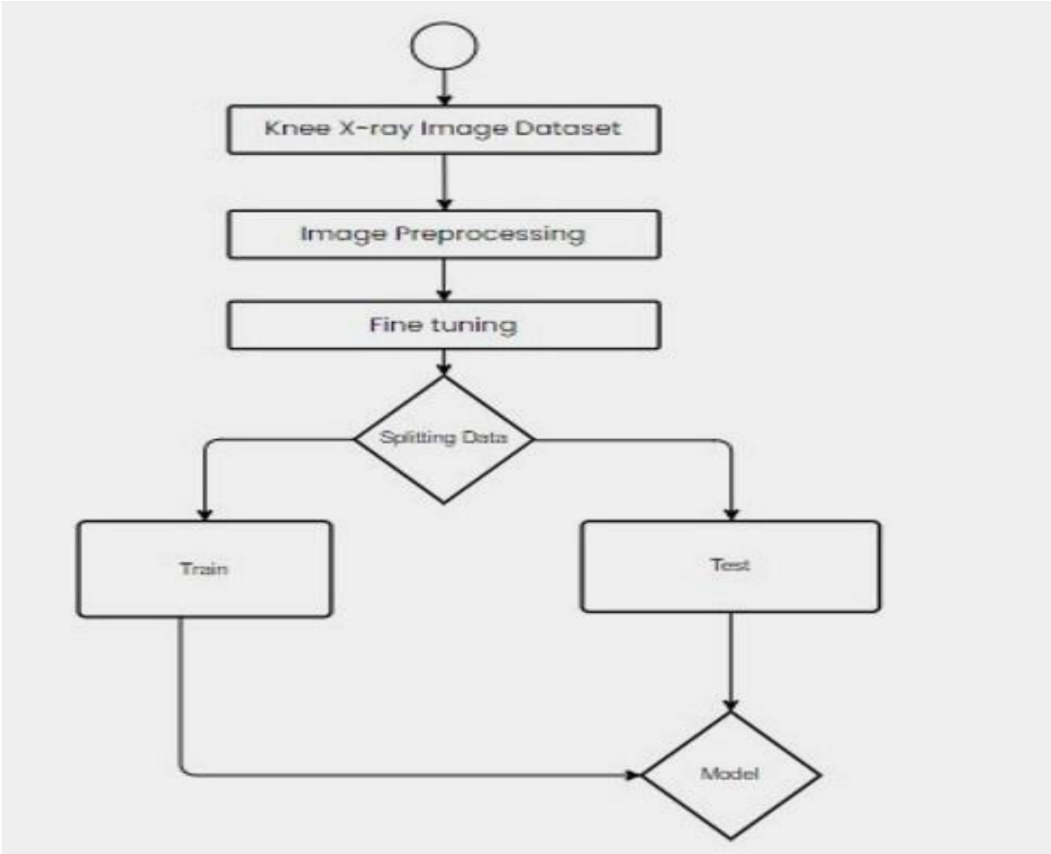


Figure 3.21 Flow Chart of Algorithm

3.7 SOFTWARE ENVIRONMENT:

The software environment for knee degenerative arthritis system scans typically involves a combination of programming languages, libraries, and frameworks. Here's an overview of the software environment commonly used for this purpose:

Python: Python is a versatile programming language widely used in machine learning, image processing, and medical imaging applications. It offers a rich ecosystem of libraries and frameworks.

NumPy: NumPy is a fundamental library for scientific computing in Python. They provide support for numerical operations, array manipulation, and mathematical functions essential for processing data.

Pandas: Pandas is a data manipulation and analysis library in Python. It offers data structures and functions for handling structured data, such as loading datasets, cleaning data, and performing exploratory data analysis.

Scikit-learn: Scikit-learn is a machine-learning library in Python that offers tools for classification, regression, clustering, and model evaluation.

TensorFlow: TensorFlow is a deep learning framework commonly used for building and training convolutional neural networks (CNNs). These frameworks provide high-level APIs for constructing neural network architectures and optimizing model parameters using gradient descent.

3.8 STEPS FOR EXECUTING THE PROJECT:

Prerequisites

To successfully complete the execution of the project, you must do the following:

1. Install the Python extension.
2. Install a version of Python 3 (for which this tutorial is written). Options include:
    i. (All operating systems) A download from python.org; typically use the Download button that appears first on the page.
    ii. (All operating systems) A download from Anaconda (for data science purposes).
3. On Windows, make sure the location of your Python interpreter is included in your PATH environment variable. You can check the location by running path at the command prompt. If the Python interpreter's folder isn't included, open Windows Settings, search for "environment", select Edit environment variables for your account, then edit the Path variable to include that folder.

Using VS Code:

1. Install Visual Studio Code (VS Code): If you haven't already installed VS Code, you can download it from the official website (https://code.visualstudio.com/)

2. Open VS Code: Launch Visual Studio Code on your system.

3. Open the Project Folder: Use the "File" menu in VS Code to open the folder containing the Python script and related files for the project.

4. Set Up Python Environment: Make sure you have Python installed on your system. You can check this by opening a terminal within VS Code and running the command `python --version`. If Python is not installed, you can download and install it from the official Python website (https://www.python.org/).

5. Install Required Python Packages: Open a terminal in VS Code and use pip to install the required Python packages. You can do this by running the following command:

pip install tensorflow opencv-python matplotlib numpy pandas scikit-learn flask PIL .This command will install the necessary packages for running the code.

6. Open the Python Script: In the Explorer pane of VS Code, navigate to the Python script file (usually named like `Main.py`) that you want to execute.

7. Run the Script: There are several ways to run the Python script in VS Code:

Press F5 to run the script in debug mode.

Use the "Run Python File in Terminal" option from the context menu (right-click on the script file).

Open a terminal in VS Code and run the script manually using command:

Python main.py

8. Follow On-Screen Instructions: Depending on the functionality of the script, you need to provide input data, interact with the program, or wait for the script to process. Review Output: After the script has finished executing, review the output in the terminal and in browser.

STEPS FOR EXECUTING PROJECT

Navigate to home page by clicking on Run Bash File.

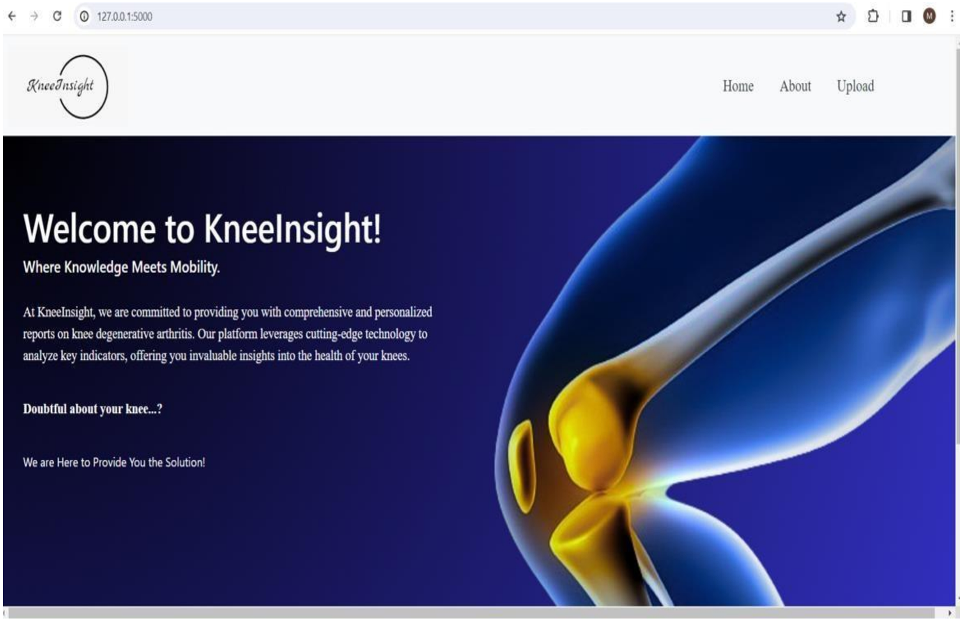


Figure 3.22: Home Page

Click on Upload option present on top right corner of web page.

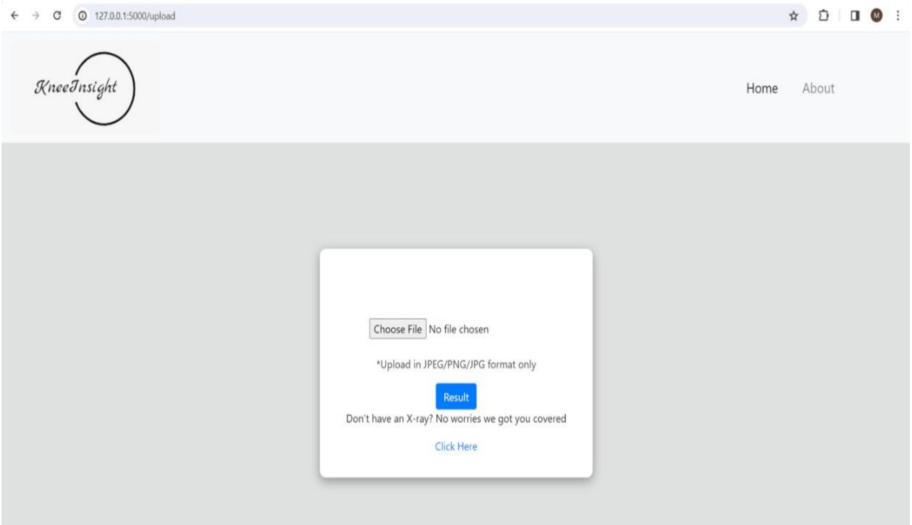


Figure 3.23: Upload Page

TESTING

4. TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are several types of tests. Each test type addresses a specific testing requirement.

4.1 TYPES OF TESTS

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It evaluates a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a test in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to evaluate areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under the test is treated as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

4.2 LEVELS OF TESTING

4.2.1 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is common for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually, and functional tests will be written in detail. Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be evaluated.

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Test Case No	Test Cases	Input	Expected O/T	Actual O/T	P/F
1	A knee X ray image with classification based on 2 grades	The model accurately identifies the knee image	The image is classified into normal stage.	It produce Pass. If not Fail	Pass
2	A knee X ray image with classification based on 2 grades	The model accurately identifies the knee image	The image is classified into osteoarthritis stage	It produce Pass. If not Fail	Pass

Table 1: Unit Testing Table for Inception V3 Model

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test Case No	Test Cases	Input	Expected O/T	Actual O/T	P/F
1	A knee X ray image with classification based on 5 grades	The model accurately identifies the knee image	The image is classified into normal stage [class_0]	It produce Pass. If not Fail	Pass
2	A knee X ray image with classification based on 5 grades	The model accurately identifies the knee image	The image is classified into doubtful stage [class_1]	It produce Pass. If not Fail	Pass
3	A knee X ray image with classification based on 5 grades	The model accurately identifies the knee image	The image is classified into minimal stage [class_2]	It produce Pass. If not Fail	Pass
4	A knee X ray image with classification based on 5 grades	The model accurately identifies the knee image	The image is classified into moderate stage [class_3]	It produce Pass. If not Fail	Pass
5	A knee X ray image with classification based on 5 grades	The model accurately identifies the knee image	The image is classified into severe stage [class_4]	It produce Pass. If not Fail	Pass

Table 2: Unit Testing Table for Xception Model

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

4.2.2 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Case No	Test Cases	Input	Expected O/T	Actual O/T	P/F
1	Read the input image.	Input image taken from user.	Image to be retrieved without errors.	Image will be retrieved without errors.	Pass

Table 3: Integration Testing Table

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

4.2.3 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Case No	Test Cases	Input	Expected O/T	Actual O/T	P/F
1	Valid Image	The model accurately identifies the knee image	The image is classified into one of the stages	It produce Pass. If not Fail	Pass
2	Invalid Image	The model accurately identifies the knee image	Prompt for Insert Correct Knee Image	Focus on image	Pass

Table 4: Acceptance Testing Table

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

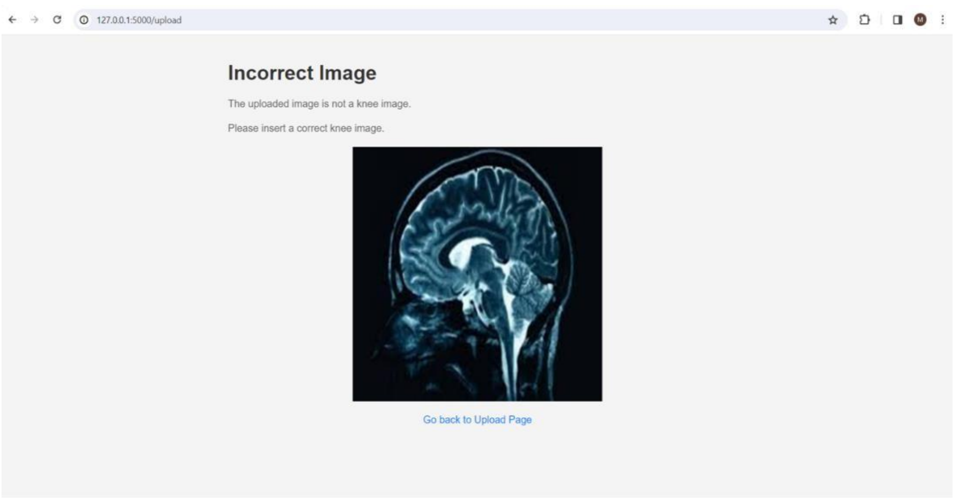


Figure 4.1: Acceptance Testing Image

RESULTS

5. RESULTS

5.1 OUTPUT SCREENS

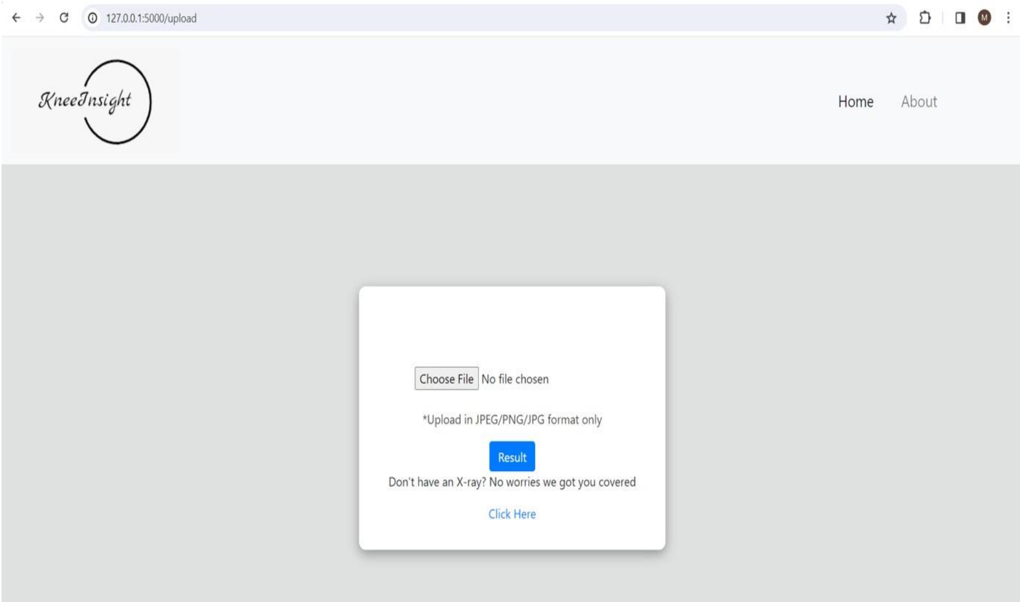


Figure 5.1: Upload Page

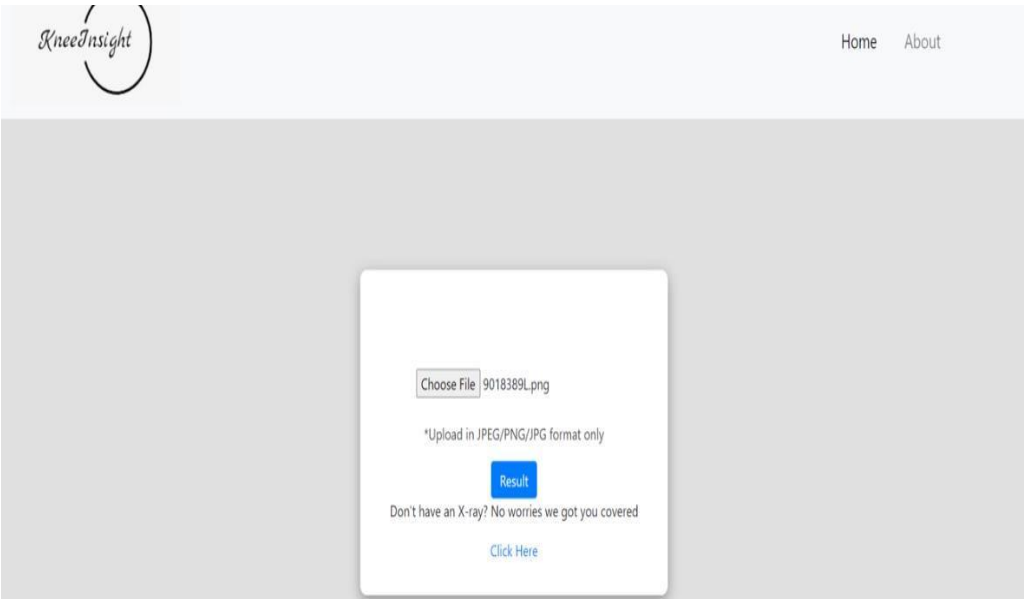


Figure 5.2: After Image Upload

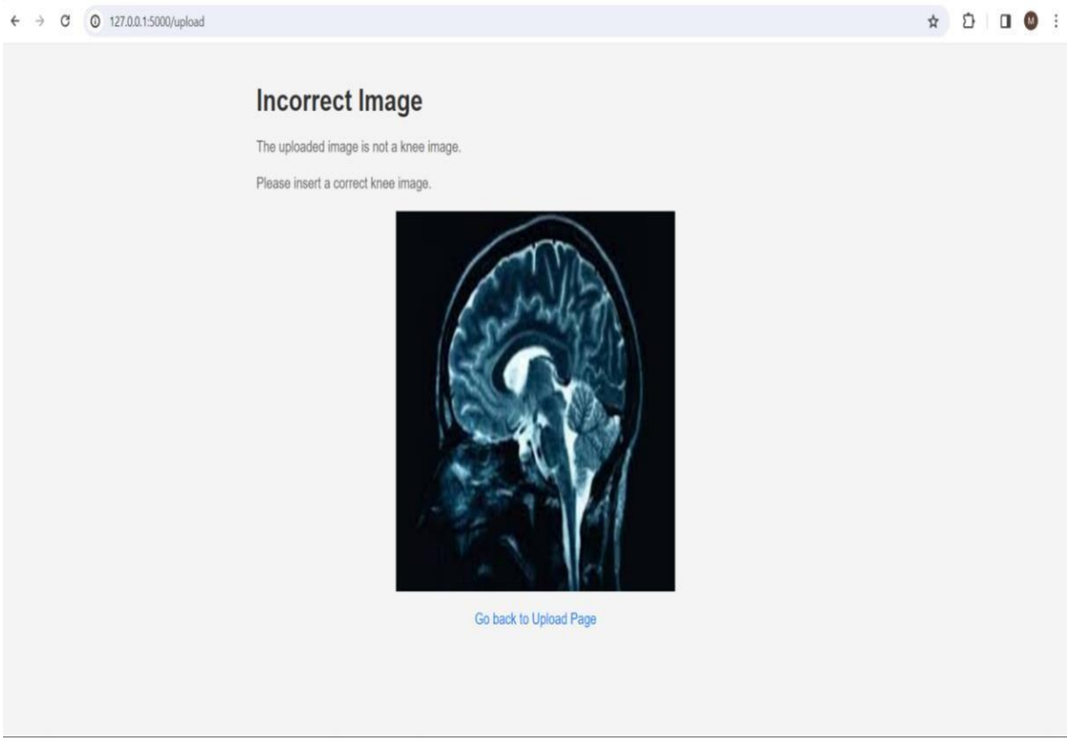


Figure 5.3: Invalid Image

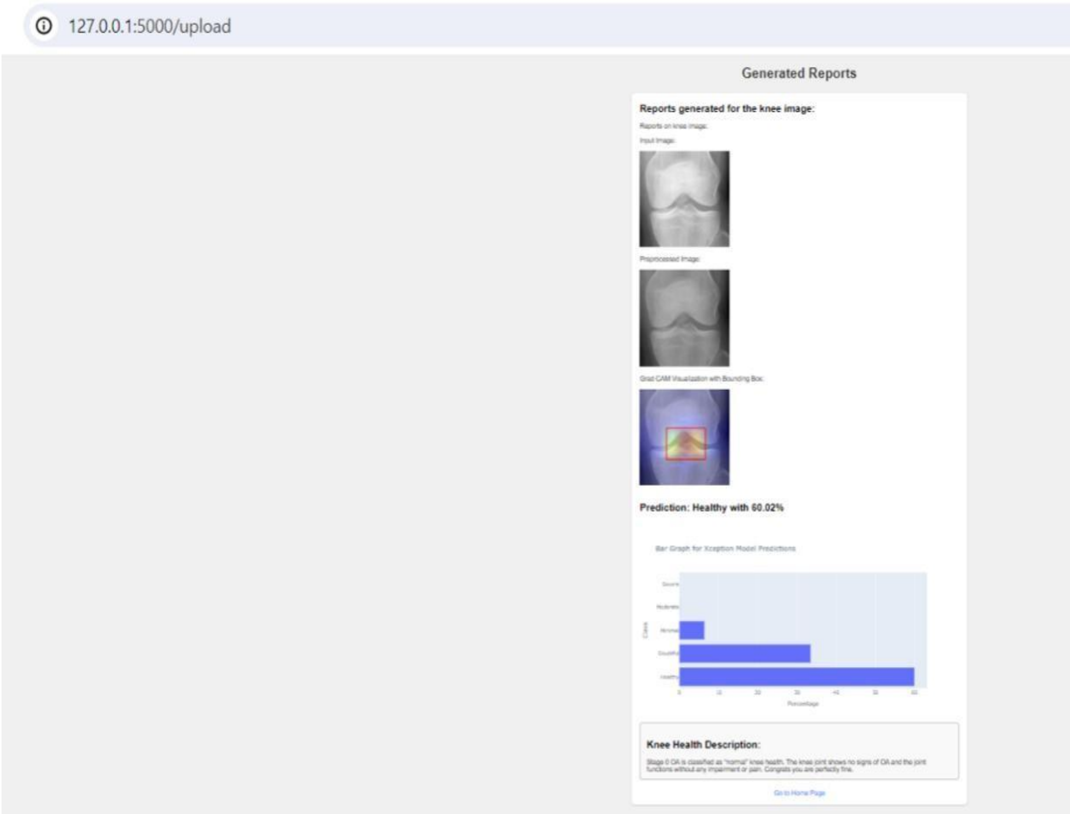


Figure 5.4: Prediction for Healthy Knee

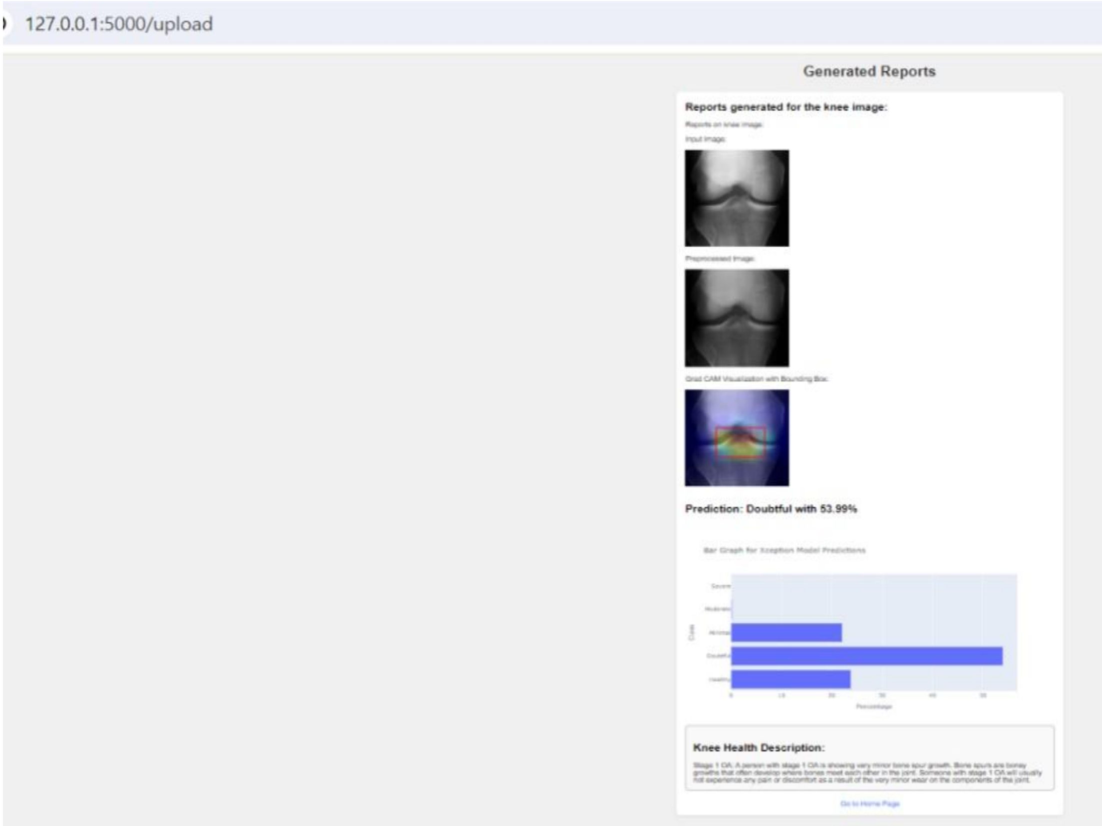


Figure 5.5: Prediction for Doubtful Knee

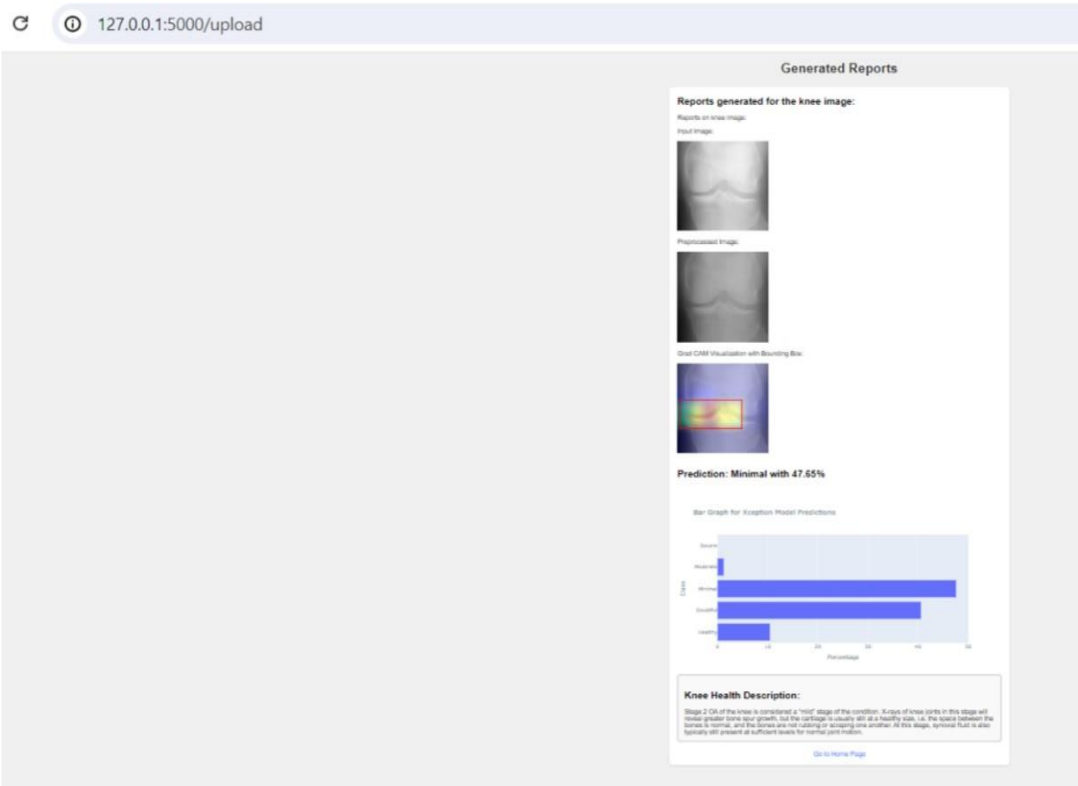


Figure 5.6: Prediction for Minimal Level Arthritis

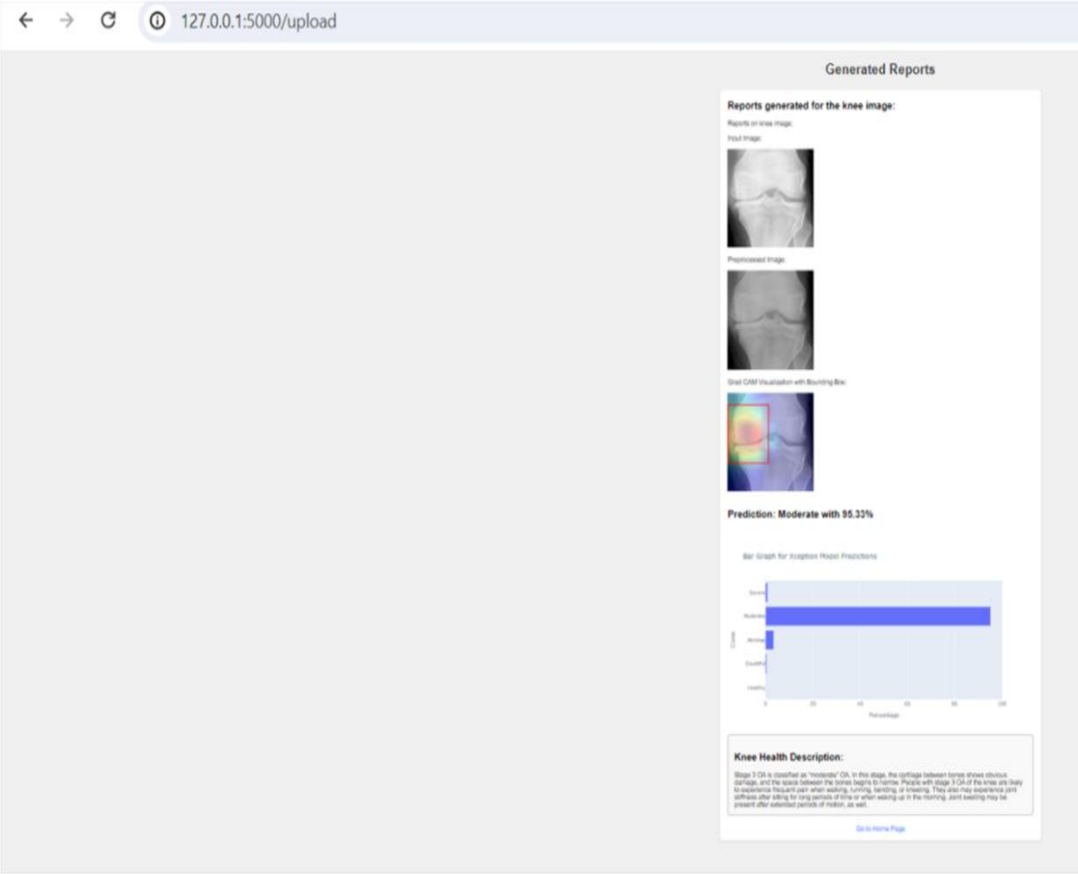


Figure 5.7: Prediction for Moderate level Arthritis

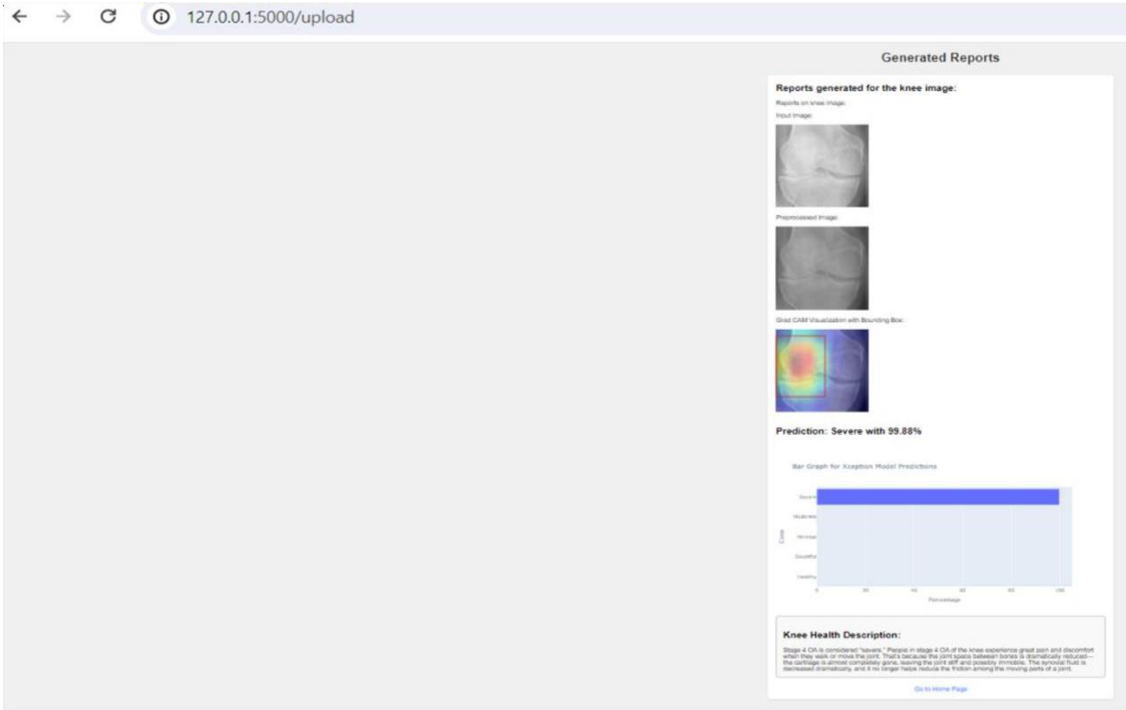


Figure 5.8: Prediction for Severe level Arthritis

5.2 METRICS

CONFUSION MATRIX:

It is a matrix of size 2×2 for binary classification with actual values on one axis and predicted on another. A confusion matrix is a table used to evaluate the performance of a classification model. It compares the predicted values from the model with the actual values in the test data and counts the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each class.

The four terms represent the following:

1. True Positive (TP): The number of correctly predicted positive instances (i.e., the model predicted positive, and the actual value was positive).

2. True Negative (TN): The number of correctly predicted negative instances (i.e., the model predicted negative, and the actual value was negative).

3. False Positive (FP): The number of incorrectly predicted positive instances (i.e., the model predicted positive, but the actual value was negative).

4. False Negative (FN): The number of incorrectly predicted negative instances (i.e., themodel predicted negative, but the actual value was positive).

A confusion matrix can be used to calculate various evaluation metrics such as precision, recall, accuracy, and F1-score. Here is an example confusion matrix:

		ACTUAL	
		Negative	Positive
PREDICTION	Negative	TRUE NEGATIVE	FALSE NEGATIVE
	Positive	FALSE POSITIVE	TRUE POSITIVE

Figure 5.9 Confusion Matrix

To calculate the various evaluation metrics, you can use the following formulas:

1. Accuracy = (TP + TN) / (TP + TN + FP + FN)*100

2. Precision = TP / (TP + FP)*100

3. Recall = TP / (TP + FN)*100

4. F1-score = 2 * ((Precision * Recall) / (Precision + Recall))

Classification Report for Xception Model

Class	Precision	Recall	F1-score	Support
0	0.73	0.80	0.76	328
1	0.36	0.36	0.36	153
2	0.69	0.56	0.61	212
3	0.70	0.74	0.72	106
4	0.81	0.96	0.88	27

Table 5: Classification Report for Xception Model

We evaluate the fine-tuned CNNs for knee OA severity classification using popular evaluation requirements such as accuracy, precision, recall, and F1-score.

Precision quantifies the percentage of true positive predictions among all positive forecasts, whereas accuracy quantifies the percentage of correctly classified occurrences.

Recall, which is a sensitivity measure as well, counts the number of true positive predictions out of all the real positive cases.

The harmonic mean of precision and recall, or F1-score, offers a fair evaluation of a classifier's performance.

Accuracy represents the number of correctly classified data instances over the total number of data instances.

The accuracy of Xception model is 65% for multi stage classification whereas Inception V3 has provided the model evaluation with accuracy of 83% for binary classification of knee images.

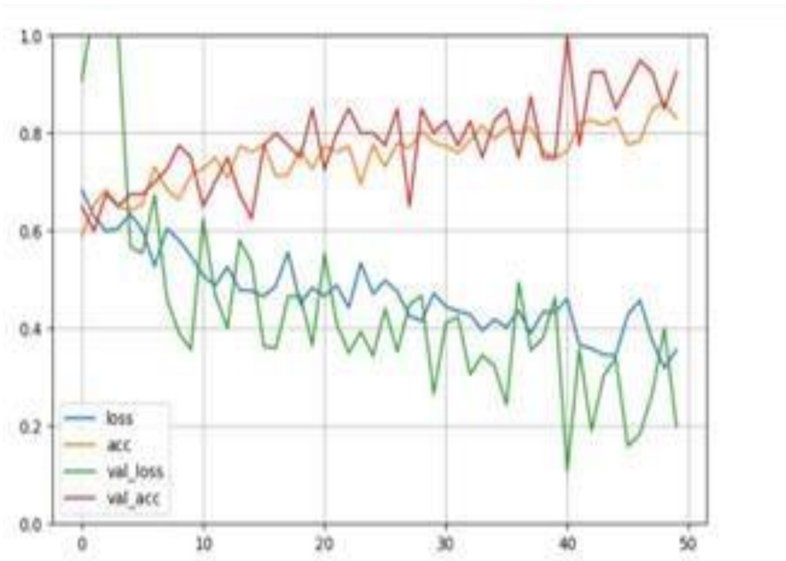


Figure 5.10: Model Evaluation for Inception V3

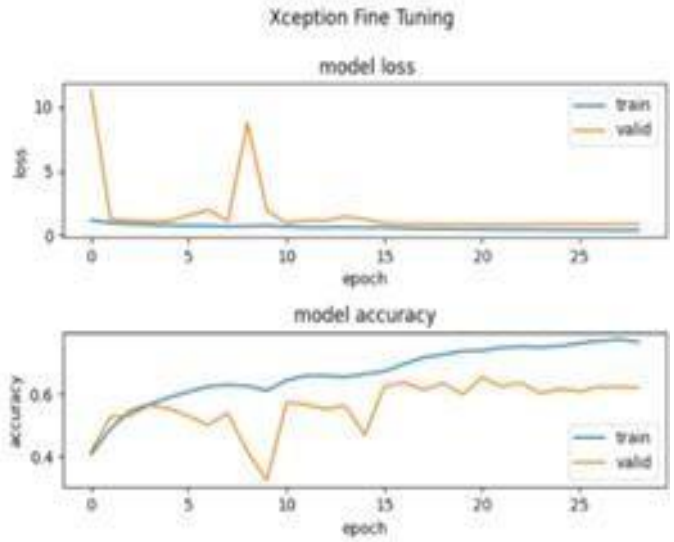


Figure 5.11: Xception model Fine Tuning

CONCLUSIONS

6. CONCLUSION

In Conclusion, the application of the Inception v3 architecture to knee image identification is an important improvement in the area of medical imaging analysis. Our research has demonstrated that the model can distinguish knee images from non-knee images with impressive accuracy through testing and validation. This capacity has significant implications for clinical practice and offers an achievable way to speed up diagnostic procedures and improve the effectiveness of medical image analysis. The Inception v3 model has been successfully implemented, demonstrating its ability to transform healthcare delivery and give users useful tools to improve decision-making and enhance patient outcomes.

Also, the use of the Xception model to provide reports for knee images that are categorized into five severity levels and combined with Grad-CAM visualization is an important step in the area of medical imaging analysis. Our research demonstrated the extent to which the model performs in classifying knee images and offers helpful details about the severity of osteoarthritis. Clinicians may enhance trust and confidence in diagnostic assessments by obtaining clarity into the model's decision-making process through the use of Grad-CAM visualization.

FUTURE SCOPE

7. FUTURE SCOPE

The Previous studies have assessed their algorithms using binary and multi-class classification metrics. We propose that it is more suitable to treat KL grades with Grad-cam analysis to generate reports on knee degenerative arthritis. Future work will focus on Simplifying the integration of AI-driven diagnostic tools into clinical workflows might be achieved by integrating the established models into the current EHR platforms. The project's future scope involves examining multimodal learning strategies, federated learning frameworks, ensemble learning methodologies, and advanced deep learning algorithms to improve knee image analysis systems' scalability, interpretability, and accuracy. The effort can further advance the study of medical imaging and the delivery of more efficient and personalized healthcare services by adopting such opportunities for research and development.

PUBLISHED PAPER

8. REFERENCES

[1] "Osteoarthritis Fact Sheet." Arthritis Foundation. 2005.

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

[3] Hunter D.J., Bierma-Zeinstra S. Osteoarthritis. Lancet. 2019.

[4] Dissectors Answers – Joints of the Upper and Lower Limbs." University of Michigan Medical School. 2000.

[5] Madry H, Kon E, Condello V, Peretti GM, Steinwachs M, Seil R, Berruto M, Engebretsen L, Filardo G and Angele P: Early osteoarthritis of the knee. Knee Surg Sports Traumatol Arthrosc. 24:1753–1762. 2016.

[6] Incze MA: I have arthritis of the knees: What Should I Do? JAMA Intern Med. 179(736)2019.

[7] Rosenblatt F. The Perceptron. A probabilistic model for information storage and organization in the brain. Psychol Rev 1958.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, "You Only Look Once: Unified, Real-Time Object Detection", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016.

[9] Ammar Anuar, Khairul Muzzammil Saipullah, Nurul Atiqah Ismail,Yewguan Soo "OpenCV Based Real-Time Video Processing UsingAndroid Smartphone" , IJCTEE, Volume 1, Issue 3.

[10] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, vol. 1. Cambridge: MIT press; 2016.

[11] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2009;22(10):1345–59.

[12] Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging, 35(5), 1285–1298.

[13]O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh,S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,et al. Imagenet large scale visual recognition challenge.2014.

[14]C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[15]N. Silberman and S. Guadarrama. Tf-slim, 2016.

[16]Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; Kim, B. Sanity checks for saliency maps.In Advances in Neural Information Processing Systems; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018.

REFERENCES

Available online at http://www.journalijdr.com

IJDR

ISSN: 2230-9926

International Journal of Development Research
Vol. 14, Issue, 02, pp. 64977-64981, February, 2024
https://doi.org/10.37118/ijdr.27897.02.2024

RESEARCH ARTICLE OPEN ACCESS

AUTOMATED KNEE DEGENERATIVE ARTHRITIS REPORTS GENERATION THROUGH VISUAL DATA ANALYSIS

Naga Malleswara Rao, V.*[1], Naga Govinda Siva Sai Adithya[2], Krishna Mahitha Nekkalapudi[2], B.N.Lokesh[2], Sri Chaitanya Deepika Loya[2] and Golkonda Meenakshi[2]

[1]Assitant Professor, Dhanekula Institute of Engineering and Technology, Ganguru, India
[2]Computer Science and Engineering, Dhanekula Institute of Engineering and Technology, Ganguru, India

ARTICLE INFO

Article History:
Received 17th January, 2024
Received in revised form
19th January, 2024
Accepted 01st February, 2024
Published online 28th February, 2024

Key Words:
Knee Degenerative Arthritis, X-rays, CNN, Computer Vision.

***Corresponding author:** *Naga Malleswara Rao, V.*

ABSTRACT

Knee degenerative arthritis is a common medical condition that affects the knee joint.Knee Degenerative arthritis causes major disability in patients all over the world. The computerized reporting procedure requires effort and expertise. Manual diagnosis, segmentation of knee joints are still used in clinical practice. Manual diagnosis of this disease involves observing X-ray images of the knee area and classifying it under five grades using the Kellgren – Lawrence (KL) system. Despite the fact that they are time-consuming and sensitive to user variance. As a result, we have the proposed system employing the CNN model with Computer Vision to increase the clinical workflow efficiency and overcome the constraints of the generally used method. We can also implement the report generation system that generates medical reports based on X-ray image features. By extracting the relevant features such as joint space narrowing, and cartilage degeneration, the system generates detailed and objective reports.

Citation: *Naga Malleswara Rao, V, Naga Govinda Siva Sai Adithya, Krishna Mahitha Nekkalapudi, B.N.Lokesh, Sri Chaitanya Deepika Loya and Golkonda Meenakshi, 2024. "Automated knee degenerative arthritis reports generation through visual data analysis". International Journal of Development Research*, 14, (02), 64977-64981.

INTRODUCTION

This study explores the creation of an automated system using visual data analysis to generate reports for patients with degenerative arthritis in the knee. Our method uses Convolutional Neural Network (CNN) methods to classify knee images into different stages of degeneration with great accuracy. Specifically, we use pre-trained models that include Inception. Additionally, we present a novel framework that utilizes exception reporting and CNN-based classification to generate detailed and informative diagnostic reports based on the stages of knee degeneration that have been observed. CNN algorithms have become more common for medical image analysis because of the ability to automatically extract complex information from images, allowing for accurate diagnosis and classification. Our method takes use of the abundance of learnt information by using pre-trained models like Inception, which are trained on large datasets, improving its ability to discern between normal and degenerative knee joints. The suggested model uses a multi-stage classification method to group knee photos into different phases of degeneration, which makes it easier to comprehend how a disease develops over time. Despite the fact that they are time-consuming and sensitive to user variance. As a result, we have the proposed system employing the CNN model with Computer Vision to increase the clinical workflow efficiency and overcome the constraints of the generally used method. We can also implement the report generation system that generates medical reports based on X-ray image features. By extracting the relevant features such as joint space narrowing, and cartilage degeneration, the system generates detailed and objective reports. CNN algorithms are growing increasingly popular for medical image analysis because of their ability to automatically extract complex information from images, enabling accurate diagnosis and classification. Our method takes advantage of a great deal of gained data by using pre-trained models like Inception, which are trained on large datasets, increasing its ability to differentiate between normal and degenerative knee joints. In addition, the suggested framework uses a multi-stage classification method to organize knee photos into various stages of degeneration, which makes it easier to understand as a disease progresses over time. Our method offers providers with a significant insight into the severity and degree of knee degeneration.Also, the efficiency and relevance of reports are enhanced by the addition of exception reporting in the report generating process. Our approach focuses on abnormalities in knee joint morphology and identifies deviations from the norm to produce brief overviews that prioritize discoveries that are clinically significant. This helps to improve diagnostic accuracy and simplify the interpretation of images.

Figure 1. Classification of Knee Image

Using CNN algorithms for image classification and exception reporting based on multi-stage knee degeneration evaluation, this research study concludes with an innovative approach to automated knee degenerative arthritis reports creation using visual data analysis. Our approach provides an achievable way of quickly and accurately detecting knee arthritis through the use of deep learning and computer vision technologies, which will ultimately lead to improved patient outcomes and healthcare delivery.

MATERIALS AND METHODS

Problem Statement: Knee degenerative arthritis is common and has an enormous adverse effect on people's quality of life, radiologists' personal assessments and manual analysis of medical images are still an important part of the diagnostic process. This method takes a long time and is prone to errors and inconsistencies, which might delay diagnosis and result in poor treatment outcomes. The lack of standardized reporting systems makes it more difficult for healthcare providers to communicate with one another making it more difficult to trace the progression of condition over time.

Objective of the Project: The objective of this research project is to utilize Convolutional Neural Network (CNN) methods, particularly leveraging pre-trained models like Inception for image classification, to develop and evaluate an automated system for knee degenerative arthritis identification and reporting. By developing multi-stage classification with exception reporting methods, this system aims to accurately classify knee images into various stages of degeneration and produce detailed diagnostic reports. In order to improve patient care and healthcare delivery, the research aims to assess the performance of the proposed system in terms of classification accuracy, report generation efficiency, and clinical utility. Ultimately, the goal is to enhance the efficiency, consistency, and reliability of knee arthritis diagnosis and management.

Scope of the Project: The scope of the project involves developing, implementing, and assessing an automated system for reporting and detecting knee degenerative arthritis, with a special focus on visual data analysis through CNN algorithms. The study includes the investigation of pre-trained models such as Inception for imageclassification, with a focus on the classification of knee pictures into various phases of degeneration. The scope also includes creating a system that combines exception reporting and multi-stage classification to produce meaningful diagnostic reports.

Dataset: This dataset contains knee X-ray data for both knee joint detection and knee KL grading. The Grade descriptions are as follows: Grade 0: Healthy knee image.Grade 1 (Doubtful): Doubtful joint narrowing with possible osteophytic lipping.Grade 2 (Minimal): Definite presence of osteophytes and possible joint space narrowing. Grade 3 (Moderate): Multiple osteophytes, definite joint space narrowing, with mild sclerosis.Grade 4 (Severe): Large osteophytes, significant joint narrowing, and severe sclerosis.

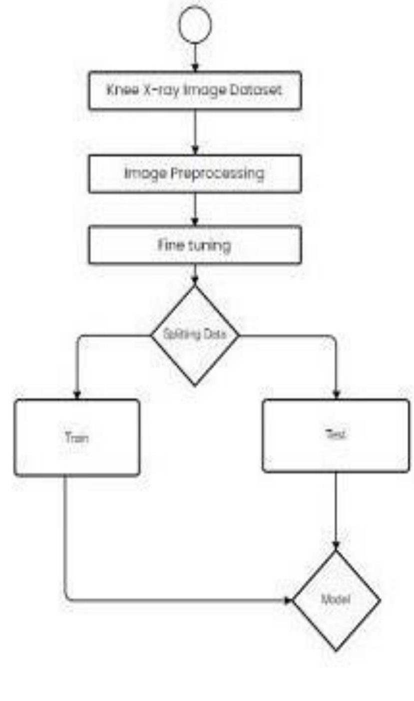
Literature Survey: Kotti et al. [1] proposed a framework to compute the likelihood and degree to which a subject may have knee OA focusing on generic subject attributes (like age, sex, assessment of the Knee Injury, Osteoarthritic Outcome Score (KOOS)) and kinematic data derived during a gait cycle to automatically classify and diagnose knee.

- Brahim et al. [2] proposed a computer aided diagnosis (CAD) system for early knee OA detection using knee X-ray imaging and machine learning algorithms.
- Dempster–Shafer theory of evidence, linear discriminant analysis and nearest neighbor classifiers.[3] proposed a fuzzy decision tree-based SVM (FDT-SVM) classifier to distinguish NL from OA knee gait patterns and investigate OA severity.
- Jose G et., al. [4] considered the case analysis and carried out on the basis of public available data from participants in the Osteoarthritis initiative analysis. There were two kinds of radiological grades used in this case study, the first is a quantitative score and the second is a semi-quantitative score, with radiological results evaluated by two classes of radiologists. To determine the future pain associated variables, A univariate logistic regression test was performed.
- Pedoia *et. al.* [5] used MRI and biomechanics multidimensional with aim to set up a multidimensional platform for improving OA outcome prediction and patient sub stratification. This approach was the first, which provided large-scale integration of compositional imaging and skeletal biomechanics.

METHODOLOGY

Proposed System: The proposed system uses computer vision and Convolutional Neural Networks (CNN) for image analysis and for report generation. In this system, CNNs are used to extract features from knee joint images, enabling the detection and classification of arthritic abnormalities. These extracted features are then fed into models that generate detailed medical reports in a contextually relevant manner. The computer vision and CNN technology not only accelerates the diagnosis process but also reduces the potential for human error, ultimately enhancing the quality of care and improving patient outcomes in the management of knee degenerative arthritis. The generated reports are not only accurate but also follow a logical flow.

Workflow of Proposed System



Data Collection: Collect an extensive set of knee joint images from include various stages of degenerative arthritis. To enable supervised learning for model training, ensure that the dataset has annotated labels that defines each knee image's stage of degeneration.

Image Preprocessing

- To enhance the quality and consistency, organize and preprocess the collected knee images using methods including noise reduction, and normalization.
- Use augmentation methods like rotation, flipping, and scaling to expand the dataset's uniqueness and stability.

Model Selection:

a) ***Inception –V3*** is a architecture of convolutional neural network (CNN) that was developed by Google researchers by combining convolutional layers with various filter sizes to collect features at different scales, it is intended to increase the efficiency and accuracy of image classification tasks. Each module in the Inception v3 architecture is made up of a number of convolutional layers, pooling layers, and other types of layers. These modules are set up in a hierarchy, with lower-level modules concentrating on the capture of low-level features like edges and textures and higher-level modules concentrating on the capture of more complex features like object components and complete objects.

b) ***Xception:*** Xception, short for "Extreme Inception," is a convolutional neural network architecture proposed by François Chollet in 2017 as an advancement over the Inception architecture. It is designed to achieve better performance and efficiency by introducing a novel approach called depthwise separable convolutions. Xception follows the same fundamental idea as the Inception architecture, which involves extracting features at multiple spatial scales using convolutional filters of different sizes. However, instead of using traditional convolutional layers, Xception employs depthwise separable convolutions.

Determining the knee severity using CNN: Using Convolutional Neural Networks (CNNs), a sort of deep learning algorithm which can automatically extract information from medical images, where we provide a novel method to determine the severity of Degenerative arthritis in the knee. We train and evaluate a CNN model for automated knee severity classification using a dataset of knee radiographs annotated with OA severity levels. By providing an efficient and objective method to determine the severity of knee degenerative arthritis, the suggested approach expects to improve patient outcomes and healthcare efficiency.

Classification of Knee using features extracted from pre-trained CNNs: Using features obtained from pre-trained CNNs, particularly Inception v3 and Xception, we provide an innovative strategy for knee OA severity classification in this research work. Our method begins by fine-tuning the images of OA severity using a dataset of knee images labeled with severity grades. From which, we extract meaningful representations of knee joints by extracting high-level features from the trained CNNs.

Fine-tuning the CNNs for classification and report generation: We introduce an approach for fine-tuning CNNs using Inception v3 and Xception for knee OA severity classification and automated report generation. The first step in our process is to improve the pre-trained CNNs using a collection of knee images that have been labeled with severity grades. We modify the CNN models to efficiently divide knee images into severity categories by iterative optimization, using transfer learning. Inception V3 is used to classify the Knee image and Xception is used to generate reports based on 5 stages of knee degenerative Arthritis.

Model Evaluation: We evaluate the effectiveness of the fine-tuned CNNs for knee OA severity classification using popular evaluation requirements such as accuracy, precision, recall, and F1-score. Precision quantifies the percentage of true positive predictions among all positive forecasts, whereas accuracy quantifies the percentage of correctly classified occurrences. Recall, which is a sensitivity measure as well, counts the number of true positive predictions out of all the real positive cases. The harmonic mean of precision and recall, or F1-score, offers a fair evaluation of a classifier's performance. Inception V3 has provided the model evaluation with accuracy of 83% which Classifies the knee image with other images.

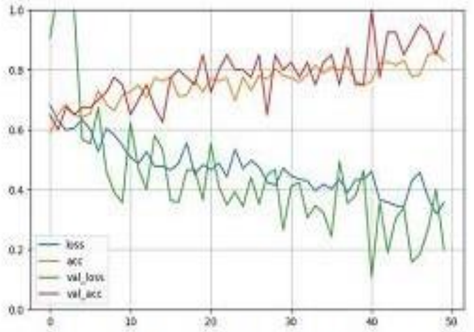
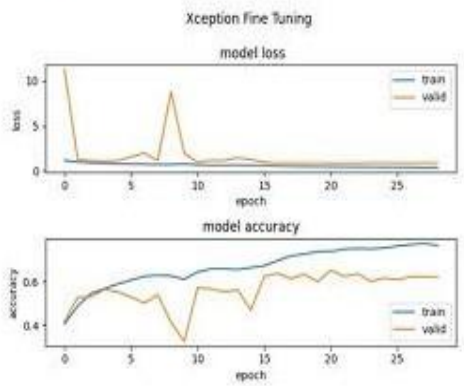


Figure 1. Model Evaluation for Classification of knee Images using Inception V3

Model loss and Model accuracy which are two fundamental metrics used to evaluate the performance of a trained model, using Convolutional Neural Network (CNN), for tasks like knee degenerative arthritis severity classification.

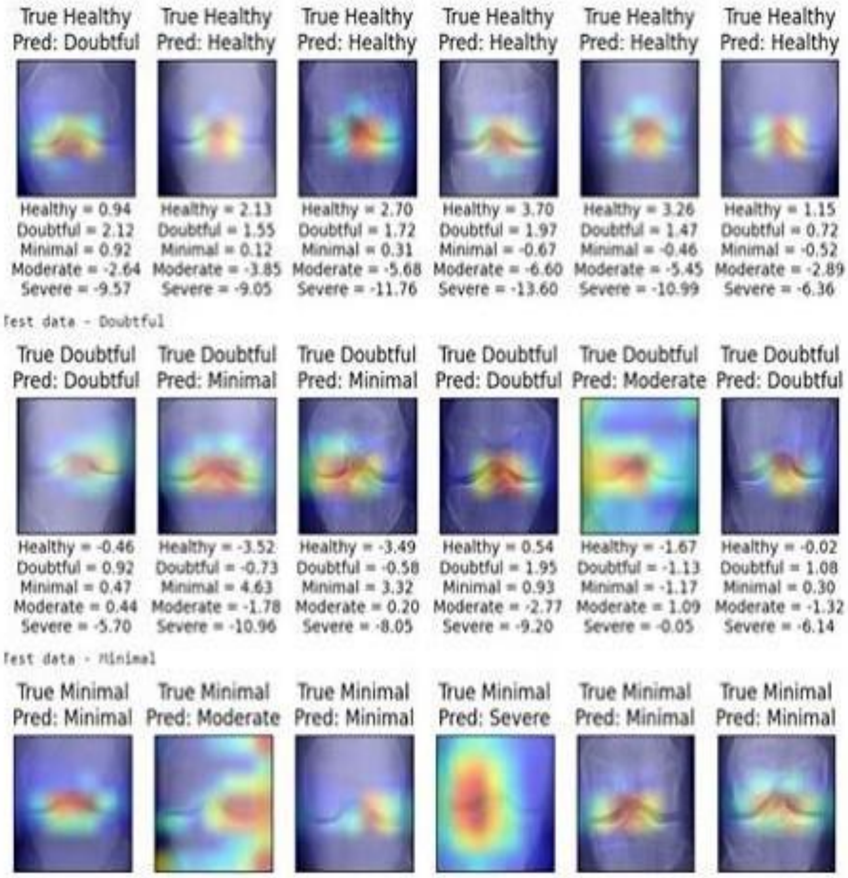


Categorical cross-entropy loss is a popular loss function for classifying the severity of degenerative arthritis. The difference between the true distribution of class labels and the expected probability distribution over classes is measured by this loss function. When data with high confidence are misclassified, the model is penalized more severely by the categorical cross-entropy loss.

	precision	recall	f1-score	support
0	0.73	0.80	0.76	328
1	0.36	0.36	0.36	153
2	0.69	0.56	0.61	212
3	0.70	0.74	0.72	106
4	0.81	0.96	0.88	27
accuracy			0.65	826
macro avg	0.66	0.68	0.67	826
weighted avg	0.65	0.65	0.65	826

The Model Evaluation using Xception pre trained model provide the accuracy of 65% with classification of knee images into 5 stages and also generates the reports based on the severity of knee and by using grad cam visualization the knee degenerative arthritis is observed and evaluated.

GRAD-CAM Visualization: Gradient-weighted Class Activation Mapping (Grad-CAM) is a visualization technique that highlights the regions of input images most relevant to the model's predictions, generating heatmaps that improve the understanding of CNN models. Users are able to better understand and have belief in the model's decision-making process by using Grad-CAM. Furthermore, the understanding of the model's predictions is evaluated by qualitative analysis of Grad-CAM heatmaps. We verify the indicated regions of interest's clinical importance by comparing the heatmaps with ground truth labels. We recommend using optimized CNN models to include Grad-CAM visualization into the knee OA severity classification process.



RESULTS

A.Detection of Knee Images

Test case no	Test Data	Expected outcome	Actual Outcome	Pass/Fail
1	A set of knee X ray images with Normal or degenerative arthritis	The model accurately identifies the knee images	If the image is not Knee image then it classified the image into not knee.	It produce Pass. If not Fail
2	A set of knee X ray images with Normal or degenerative arthritis	The model accurately identifies the knee images	If the image is Knee image then it classified the image into knee.	It produce Pass. If not Fail

B. Reports Generation Based on Classification of Knee Images into 5 grades

Test case no	Test Data	Expected outcome	Actual Outcome	Pass/Fail
1	A set of knee X ray images with classification based on 5 grades	The model accurately identifies the knee images	The image is classified into normal stage[class_0]	It produce Pass. If not Fail
2	A set of knee X ray images with classification based on 5 grades	The model accurately identifies the knee images	The image is classified into doubtful stage[class_1]	It produce Pass. If not Fail
3	A set of knee X ray images with classification based on 5 grades	The model accurately identifies the knee images	The image is classified into minimal stage[class_2]	It produce Pass. If not Fail
4	A set of knee X ray images with classification based on 5 grades	The model accurately identifies the knee images	The image is classified into moderate stage [class_3]	It produce Pass. If not Fail
5	A set of knee X ray images with classification based on 5 grades	The model accurately identifies the knee images	The image is classified into severe stage[class_4]	It produce Pass. If not Fail

Following the CNN model's training on knee images labeled with severity grades, we use Grad-CAM to produce heatmaps that highlight the input images with particular regions. By using Grad Cam Visualization we have evaluated and provides the regions where the knee has effected and produced reports based on the images. THE Report Generation is based on the Grad-Cam Analysis. By using Grad cam Visualization the Knee images categorized into 5 stages where the area of knee is effected and also identified.

CONCLUSION

In Conclusion, the application of the Inception v3 architecture to knee image identification is an important improvement in the area of medical imaging analysis. Our research has demonstrated that the model can distinguish knee images from non-knee images with impressive accuracy through testing and validation. This capacity has significant implications for clinical practice and offers an achievable way to speed up diagnostic procedures and improve the effectiveness of medical image analysis. The Inception v3 model has been successfully implemented, demonstrating its ability to transform healthcare delivery and give physicians useful tools to improve decision-making and enhance patient outcomes. Also,the use of the Xception model to provide reports for knee images that are categorized into five severity levels and combined with Grad-CAM visualization is an important step in the area of medical imaging analysis. Our research demonstrated the extent to which the model performs in classifying knee images and offers helpful details about the severity of osteoarthritis. Clinicians may enhance trust and confidence in diagnostic assessments by obtaining clarity into the model's decision-making process through the use of Grad-CAM visualization.

Future Scope: The Previous studies have assessed their algorithms using binary and multi-class classification metrics. We propose that it is more suitable to treat KL grades with Grad-cam analysis to generate reports on knee degenerative arthritis. Future work will focus on Simplifying the integration of AI-driven diagnostic tools into clinical workflows might be achieved by integrating the established models into the current EHR platforms. The project's future scope involves examining multimodal learning strategies, federated learning frameworks, ensemble learning methodologies, and advanced deep learning algorithms to improve knee image analysis systems' scalability, interpretability, and accuracy. The effort can further advance the study of medical imaging and the delivery of more efficient and personalized healthcare services by adopting such opportunities for research and development.

REFERENCES

Biau G. and E. Scornet, “A random forest guided tour,” Test, vol. 25, no. 2, pp. 197–227, 2016.

Breiman, L. “Random forests,” Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

Cortes C. and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

Friedman, J. H. “Greedy function approximation: a gradient boosting machine,” Annals of statistics, pp. 1189–1232, 2001.

Hastie, T. R. Tibshirani, and J. Friedman, “Boosting and additive trees,” in The elements of statistical learning, Springer, 2009, pp. 337– 387.

Huang, S. N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, “Applications of support vector machine (SVM) learning in cancer genomics,” Cancer Genomics-Proteomics, vol. 15, no. 1, pp. 41–51, 2018.

Jahromi A. H. and Taheri, M. “A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features,” in 2017 Artificial Intelligence and Signal Processing Conference (AISP), 2017, pp. 209–212.

Johnson, S. C. “Hierarchical clustering schemes,” Psychometrika, vol. 32, no. 3, pp. 241–254, 1967.

Kleinbaum D. G. and M. Klein, “Logistic regression, statistics for biology and health,” Retrieved from DOI, vol. 10, pp. 978–1, 2010.

Perez, A. P. Larranaga, and I. Inza, “Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naive Bayes,” International Journal of Approximate Reasoning,vol. 43, no. 1, pp. 1–25, 2006.

Reynolds, D. A. “Gaussian Mixture Models.,” Encyclopedia of biometrics, vol. 741, 2009.

Taud H. and Mas, J. “Multilayer perceptron (MLP),” in Geomatic Approaches for Modeling Land Change Scenarios, Springer, 2018, pp. 451–455.

Vapnik, V. The nature of statistical learning theory. Springer science & business media, 2013.

Yu, Z. X. Zhu, H.-S. Wong, J. You, J. Zhang, and G. Han, “Distribution-based cluster structure selection,” IEEE transactions on cybernetics, vol. 47, no. 11, pp. 3554–3567, 2016.

****