PREDICTION OF THE 8 STAGES OF OSTEOARTHRITIS USING DEEP LEARNING

PHASE I REPORT

Submitted by

MADHUMITA P – 2116210701140

KAVYA SHREE BN – 2116210701702

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING

ANNA UNIVERSITY, CHENNAI

NOVEMBER 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled **"Prediction of the 8 stages of osteoarthritis using deep learning"** is the bonafide work of **MADHUMITA P (2116210701140), KAVYA SHREE BN (2116210701702)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. P. KUMAR M.E., Ph.D.,

Professor and Head,

Dept. of Computer Science and Engineering,

Rajalakshmi Engineering College,

Thandalam, Chennai - 602 105

Dr. T. Kumaragurubaran M.Tech., Ph.D.,

Assistant Professor(SG)

Dept. of Computer Science and Engineering,

Rajalakshmi Engineering College,

Thandalam, Chennai - 602 105

Submitted to Project and Viva Voce Examination for the subject CS19711-ProjectPhase – I held on ________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr. T. Kumaragurubaran M.Tech., Ph.D.,** Assistant Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Dr. T. Kumaragurubaran**, Department of Computer Science and Engineering for his useful tips during our review to build our project.

MADHUMITA - 210701140

KAVYA SHREE BN – 210701702

ABSTRACT

A novel approach for the classification of knee osteoarthritis from medical images would be developed in this research, utilizing TensorFlow and exploiting ensembles of Convolutional Neural Networks. Knee osteoarthritis is one of the most common diseases involving joints, for which proper classification would be critical in conducting early diagnosis and proper planning for intervention. Our approach combines several CNN architectures as an ensemble to generate a diverse feature set, enhancing both classification accuracy and robustness. The different subsets of the data are learned upon by the set of different CNNs in the ensemble so that the features learned are maximized. It is further applied with the ensemble techniques of bagging and boosting for improvements in the performance such that the predictions made by individual models may be aggregated. All this is done while testing the state-of-art classification accuracy on a large dataset of knee radiographs. By benchmarking against assessments by expert radiologists, it could very well turn out to be a realistic useful aide in clinical decision-making. Apart from the improvement brought in by this study concerning the accuracy of diagnosis in knee OA, the effectiveness of CNN ensembles in complex challenges in the medical analysis of images is underscored. By accumulating several strengths of multiple neural network architectures, the approach shows huge advancements in capabilities regarding the application of machine learning techniques and diagnostics.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ACKNOWLEDGEMENT	v
	ABSTRACT	vi
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	viii
1.	INTRODUCTION	1
	1.1 GENERAL	1
	1.2 OBJECTIVE	3
	1.3 EXISTING SYSTEM	4
	1.4 PROPOSED SYSTEM	7
2.	LITERATURE SURVEY	10
3.	SYSTEM DESIGN	15
	3.1 GENERAL	15
	3.1.1 SYSTEM FLOW DIAGRAM	15
	3.1.2 SEQUENCE DIAGRAM	16
	3.1.3 CLASS DIAGRAM	17
	3.1.4 USECASE DIAGRAM	18
	3.1.5 ARCHITECTURE DIAGRAM	19
	3.1.6 ACTIVITY DIAGRAM	20
	3.1.7 COMPONENT DIAGRAM	21
	3.1.8 COLLABORATION DIAGRAM	22
4.	PROJECT DESCRIPTION	23
	4.1 METHODOLOGIES	23
	4.1.1 RESULT DISCUSSION	28

5.	CONCLUSION AND WORK SCHEDULE	33
	FOR PHASE II	34
	REFERENCES	35
	APPENDIX	38

CHAPTER 1

1. About:

This study presents a novel approach utilizing Convolutional Neural Network (CNN) ensembles for the classification of knee osteoarthritis (OA) from medical images, implemented within the Tensor Flow framework. Osteoarthritis is a prevalent joint disorder, and accurate classification aids in early diagnosis and intervention planning. Our proposed ens emble method combines multiple CNN architectures to exploit diverse feature representations, enhancing classification performance. Each CNN in the ensemble is trained on a subset of the dataset, Furthermore, ensemble techniques such as bagging and boosting are employed to aggregate predictions and improve robustness. The system is evaluated on a large dataset of knee radiographs, achieving state-of-the-art classification accuracy. Moreover, the ensemble's performance is validated against expert radiologist assessments, demonstrating its potential as an effective tool for assisting in clinical decision-making. This approach not only contributes to advancing knee OA diagnosis but also highlights the efficacy of CNN ensembles in medical image analysis tasks.

Existing System:

The existing surface electromyography-based pattern recognition system (sEMG-PRS) exhibits limited generalizability in practical applications. In this paper, we propose a stacked weighted random forest (SWRF) algorithm to enhance the long-term usability and user adaptability of sEMG-PRS. First, the weighted random forest (WRF) is proposed to address the issue of imbalanced performance in standard random forests (RF) caused by randomness in sampling and feature selection. Then, the stacking is employed to further enhance the generalizability of WRF. Specifically, RF is utilized as the base learner, while WRF serves as the meta-leaning layer algorithm. The SWRF is evaluated against classical classification algorithms in both online experiments and offline datasets. The offline experiments indicate that the SWRF achieves an average classification accuracy of 89.06%, outperforming RF, WRF, long short-term memory (LSTM), and support vector machine (SVM). The online experiments indicate that SWRF outperforms the aforementioned algorithms regarding long-term usability and user adaptability. We believe that our method has significant potential for practical application in sEMG-PRS.

Existing System Disadvantages:

Surface electromyography-based pattern recognition systems (sEMG-PRS) often face challenges in achieving generalizability in practical applications. While these systems excel in controlled environments, their performance can degrade significantly in real-world settings due to variations in user physiology, electrode placement, and environmental factors. This limitation poses a hurdle to their widespread adoption, especially in dynamic and unpredictable contexts.

Standard random forests (RF) may exhibit imbalanced performance due to the inherent randomness in sampling and feature selection. Although RF is a robust machine learning algorithm, the stochastic nature of its operations can lead to inconsistencies in model performance, particularly when dealing with uneven class distributions or noisy datasets. This variability can compromise the reliability of predictions in sensitive applications.

Long-term usability and user adaptability are potential concerns for sEMG-PRS systems. Over time, users may experience changes in muscle activity patterns or fatigue, which can affect system accuracy and user satisfaction. Additionally, users may find it challenging to adapt to the system’s demands, especially if it requires frequent recalibration or fails to consistently deliver expected outcomes.

Achieving optimal classification accuracy remains a significant challenge when compared to advanced algorithms. While sEMG-PRS systems and standard RF offer considerable benefits in certain scenarios, they may fall short of the performance levels attained by cutting-edge techniques such as deep learning or ensemble methods. This performance gap underscores the need for continuous innovation and refinement to enhance their practical utility.

2. INTRODUCTION:

Knee osteoarthritis (OA) is a degenerative joint disease that significantly impacts the quality of life of millions worldwide. Characterized by the progressive deterioration of cartilage, it leads to pain, stiffness, and reduced mobility. Early and accurate diagnosis of knee OA is crucial for effective management and treatment. Traditional diagnostic methods often rely on subjective assessments, such as physical examinations and radiographic imaging, which can be prone to variability and misinterpretation. The advent of deep learning, particularly convolutional neural networks (CNNs), offers a promising alternative for enhancing the accuracy and efficiency of osteoarthritis classification by leveraging large datasets of medical images.

4. Domain overview:

4.1 Data Science:

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

The origin of the term "data science" was coined and popularized in 2008 by the influential D.J. Patil and Jeff Hammerbacher. Patil as well as Hammerbacher are pioneers and leaders of data and analytics initiatives of the leading tech companies such as LinkedIn and Facebook. In an incredibly short span of less than ten years, the field of data science has rapidly evolved and established itself as one of the most in-demand and highly trending career paths available in today's job market. Data science is a field of study that encompasses a wide variety of facets by integrating detailed domain knowledge, rigorous programming skills, and exceptionally deep mathematical and statistical knowledge. It amalgamates all these aspects to extract meaningful insights that are valuable from unprecedented amounts of data. Data science can be termed to be narrowly a field consisting of an amalgam of mathematics, business acumen, several analytical tools, complex algorithms, and advanced techniques of machine learning. Collectively, these elements synergistically help us discover the concealed insights or patterns hidden in the unprocessed raw data. These correctly extracted insights are highly elemental in shaping and guiding the creation of significant business decisions that may eventually affect ultimate organizational success and growth.**Data Scientist:**

Required Skills for a Data Scientist:

- **Programming**: Python, SQL, Scala, Java, R, MATLAB.
- **Machine Learning**: Natural Language Processing, Classification, Clustering.
- **Data Visualization**: Tableau, SAS, D3.js, Python, Java, R libraries.
- **Big data platforms**: MongoDB, Oracle, Microsoft Azure, Cloudera.

4.2 ARTIFICIAL INTELLIGENCE:

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals. Leading AI textbooks define the field as the study of “intelligent agents” any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term “artificial intelligence” to describe machines that mimic “cognitive” functions that humans associate with the human mind, such as “learning” and “problem solving”, however this definition is rejected by major AI researchers. Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an “AI winter”), followed by new approaches, success and renewed funding.

AI research has tried and discarded many different approaches during its lifetime, including simulating the hydroponic plant disease, modeling human problem solving, formal logic, large databases of knowledge and imitating animal behavior. In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly successful, helping to solve many challengingproblems throughout industry and academia. The various subfields of AI research are centered around particular goals and the use of particular tools.

To solve these problems, AI researchers use versions of search and mathematical optimization, formal logic, artificial neural networks, and methods based on statistics, probability and economics. AI also draws upon computer science, psychology, linguistics, philosophy, and many other fields. The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence. These issues have been explored by myth, fiction and philosophy since antiquity. Science fiction and futurology have also suggested that, with its enormous potential and power, AI may become an existential risk to humanity. As the hype around AI has accelerated, vendors have been scrambling to promote how their products and services use AI. Often what they refer to as AI is simply one component of AI, such as machine learning.

AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular. In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a chatbot that is fed examples of text chats can learn to produce life like exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples. AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

Learning processes. This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Reasoning processes. This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

Self-correction processes. This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.

AI is important because it can give enterprises insights into their operations that they may not have been aware of previously and because, in some cases, AI can perform tasks better than humans. Particularly when it comes to repetitive, detail-oriented tasks like analyzing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors. Artificial neural networks and deep learning artificial intelligence technologies are quickly evolving, primarily because AI processes large amounts of data much faster and makes predictions more accurately than humanly possible.

4.3 DEEP LEARNING

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human disease so deep learning is also a kind of mimic of human disease. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. A formal definition of deep learning is- neurons Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. In disease approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousands of their neighbors. The question here is how it recreates these neurons in a computer. So, it creates an artificial structure called an artificial neural net where we have nodes or neurons. It has some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.It need to identify the actual problem in order to get the right solution and it should be understood, the feasibility of the Deep Learning should also be checked (whether it should fit Deep Learning or not). It needs to identify the relevant data which should correspond to the actual problem and should be prepared accordingly. Choose the Deep Learning Algorithm appropriately. Algorithm should be used while training the dataset. Final testing should be done on the dataset as explained in Fig.2

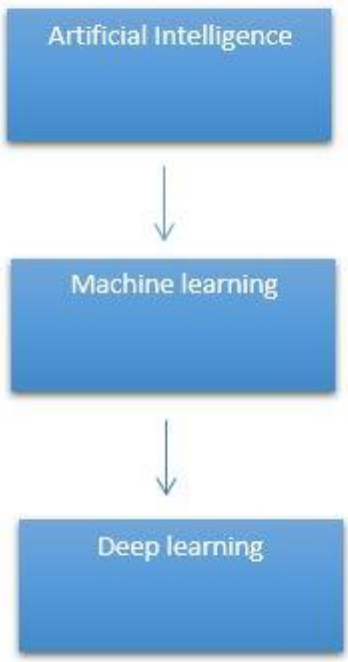


Fig.1 Branches of Artificial Intelligence

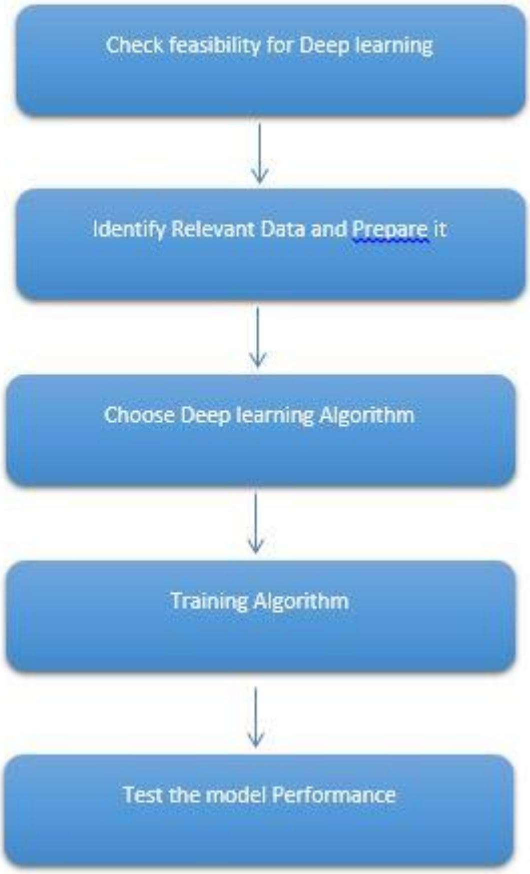


Fig. 2 Steps involved in deep learning

Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological disease. Specifically, neural networks tend to be static and symbolic, while the biological disease of most living organisms is dynamic (plastic) and analogue. The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a non-polynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part.

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

Interpretations:

Deep neural networks are generally interpreted in terms of the universal approximation theorem or probabilistic inference. The classic universal approximation theorem concerns the capacity of feed-forward neural networks with a single hidden layer of finite size to approximate continuous functions. In 1989, the first proof was published by George Cybenko for sigmoid activation functions and was generalised to feed-forward multi-layer architectures in 1991 by Kurt Hornik. Recent work also showed that universal approximation also holds for non-bounded activation functions such as the rectified linear unit.

The universal approximation theorem for deep neural networks concerns the capacity of networks with bounded width but the depth is allowed to grow proved that if the width of a deep neural network with ReLU activation is strictly larger than the input dimension, then the network can approximate any Lebesgue integrable function; If the width is smaller or equal to the input dimension, then deep neural network is not a universal approximator.

The probabilistic interpretation derives from the field of machine learning. It features inference, as well as the optimization concepts of training and testing, related to fitting and generalization, respectively. More specifically, the probabilistic interpretation considers the activation nonlinearity as a cumulative distribution function. The probabilistic interpretation led to the introduction of dropout as regularizer in neural networks. The probabilistic interpretation was introduced by researchers including Hopfield, Widrow and Narendra and popularized in surveys such as the one by Bishop.

Deep learning revolution:

A multi-task deep neural network set by George E. Dahl won the "Merck Molecular Activity Challenge" in 2012 for predicting the biomolecular target of one drug. In 2014, Hochreiter's group used deep learning for detecting off-target and toxic effects of environmental chemicals in nutrients, household products, and drugs and thereby won "Tox21 Data Challenge" of NIH, FDA, and NCATS.

Critical other advances occurred in the domain of image or object recognition during 2011 to 2012. Although CNNs have been known since at least decades because CNNs were trained by back-propagation, which is a variant of stochastic gradient descent-the implementation speed of CNNs on

GPUs is what propelled computer vision forward. The strategy-for the first time in 2011-notably managed to attain superhuman performance in a visual pattern recognition contest. Moreover, in 2011, it also won the ICDAR Chinese Handwriting Contest, and, in May 2012, it won the ISBI image segmentation contest. Before 2011, CNNs did not have a key role at computer vision conferences, but, in June 2012, a paper of Ciresan et al. at leading conference CVPR showed how max-pooling CNNs on GPU can dramatically improve many vision benchmark records.

A system like ours was designed by Krizhevsky et al. in October 2012, which won the large-scale competition ImageNet with an exceptionally significant margin over shallow machine learning methods. November 2012: Ciresan et al.'s system also won the ICPR contest on analysis of large medical images for cancer detection, and in the next year also the MICCAI Grand Challenge on the same topic. In 2013 and 2014, by using deep learning, error rates on the ImageNet task were even more subdued, as for big speech recognition.

The generation of captions or descriptions of images proved to be an added complexity to the image classification challenge. This was done by combining CNNs and LSTMs.
According to some researchers, a victory in an ImageNet competition in October 2012 anchored the beginning of a "deep learning revolution" that would transform this sector.

In March 2019, the Turing Award given to Yoshua Bengio, Geoffrey Hinton, and Yann LeCun recognized conceptual and engineering breakthroughs which had enabled deep neural networks to become an indubitable mainstream component of computing.

5 PROPOSED SYSTEM:

The proposed system employs an ensemble of Convolutional Neural Networks (CNNs) for the classification of knee osteoarthritis, utilizing Tensor Flow for implementation. This ensemble approach combines multiple CNN models to enhance diagnostic accuracy and robustness. Each CNN in the ensemble is designed with several convolutional layers for feature extraction from images, followed by pooling layers to reduce spatial dimensions, and fully connected layers for classification. By integrating outputs from diverse CNN architectures, the system leverages the strengths of each model, achieving superior performance compared to any single model. especially when dealing with limited medical imaging data. Data augmentation techniques are applied to improve generalization, making the system resilient to variations in image quality and patient demographics. Real-time processing capabilities enable timely diagnosis, which is crucial for effective treatment planning. Tensor Flow's scalability ensures the system can handle large datasets and be deployed across various clinical settings.

This ensemble CNN approach aims to significantly improve the early detection and classification of knee osteoarthritis, leading to better patient outcomes and more efficient healthcare delivery.

Proposed System Advantages:

1. Enhanced diagnostic accuracy and robustness through an ensemble of Convolutional Neural Networks (CNNs).

2. Superior performance compared to any single model, especially with limited medical imaging data.

3. Integration of diverse CNN architectures to leverage the strengths of each

4. Improved generalization through data augmentation techniques, resilient to variations in image quality and patient demographics.

5. Improves accuracy level and Django web application implemented.

That is why a system of CNNs elevates the accuracy and robustness of the diagnoses obtained, providing far better performance than any single model, especially when using limited medical imaging data. Architectures incorporating various strengths are brought together in a CNN system that engages unique strengths distributed throughout each model, creating an even more balanced, comprehensive, and reliable diagnostic tool. Adding on that, data augmentation techniques enhance robustness of the system against variations in quality of images and patient demographics, thereby generalizing the same better. The precision levels thus improve with it further being practicable through a Django web application and thereby becoming accessible and use friendly too.

6 PREPARING THE DATASET:

That is why a system of CNNs elevates the accuracy and robustness of the diagnoses obtained, providing far better performance than any single model, especially when using limited medical imaging data. Architectures incorporating various strengths are brought together in a CNN system that engages unique strengths distributed throughout each model, creating an even more balanced, comprehensive, and reliable diagnostic tool. Adding on that, data augmentation techniques enhance robustness of the system against variations in quality of images and patient demographics, thereby generalizing the same better.

The precision levels thus improve with it further being practicable through a Django web application and thereby becoming accessible and use friendly

CHAPTER 2

LITERATURE SURVEY

Review of Literature Survey

Title: Medical Image Analysis of Knee Osteoarthritis using Modified Deep CNN

Author: Mohammed Zakir Bellary, Deepthi

Year: 2023

Kellgren and Lawrence (KL) grading method is mainly used by the clinicians for grading the Xray images. However, grading individual images are prone to errors. This study proposes an approach for automated knee osteoarthritis classification based on deep neural networks. Diagnosis of Osteoarthritis involves splitting the knee X-ray into the healthy knee or unhealthy knee with KL grading. Here we use a 20-layer deep residual networks ie ResNet 20 for automated knee osteoarthritis classification. ResNet 20 has 19 convolutional layers and 1 fully-connected layer.We analysed the dataset with different CNN models. And got the different performance.

Title: Enhanced deep neural networks for early diagnosis of knee osteoarthritis

Author: Yassine Nasser

Year: 2023

K nee Osteoarthritis (OA) is one of the most frequent causes of physical disability worldwide and is associated with significant personal and socioeconomic burdens. There is a considerable need to develop automated methods for early diagnosis of knee OA. Over the last few years, Deep Learning (DL) models have gained remarkable attention from the computer vision research community and achieved great success in various medical imaging applications. This thesis aimed to develop DL-based models for fully automatic knee OA diagnosis using radiographic images. In this thesis, several methods for knee OA severity assessment and OA prediction are evaluated, and new methods are introduced. First, we focus on the feature-learning step as a crucial component of the classification system to learn and extract the most useful discriminative features from X-ray images. To do so, we introduce a novel autoencoder-based architecture called Discriminative Regularized Auto-encoder (DRAE). The goal is to maximize the distance between class features by minimizing the intraclass distance and maximizing the interclass distance. Then, we propose incorporating the proposed discriminative regularization in the standard Convolutional Neural

Network (CNN) learning process to improve the early detection of knee OA. By doing so, we reduce the inability of CNNs to handle data with high inter-class similarities or high intra-class variations. While the proposed DRAE focuses on the texture information in the radiography under the tibial plateau, the second proposed learning model, called Discriminative Convolutional Neural Network (DCNN), uses the overall distal area of the knee and exploits both texture and shape representations. To further learn discriminative features and exploit shape and texture, we propose : (i) enhancing texture analysis by adding a new block to the DCNN architecture and (ii) improving the proposed discriminative loss to fit with multi-class classification tasks. The resulting model, called Discriminative Shape-Texture Convolutional Neural Network (DST-CNN), enables better and well-balanced classification performance than existing State-of-the-Art (SoA) model

Title: automatic detections of knee joints and classification of knee osteoarthritis severity form plain radiographs using cnns

Author: David duran olivar

Year: 2023

K nee Osteoarthritis (OA) is one of the most frequent causes of physical disability worldwide and is associated with significant personal and socioeconomic burdens. There is a considerable need to develop automated methods for early diagnosis of knee OA. Over the last few years, Deep Learning (DL) models have gained remarkable attention from the computer vision research community and achieved great success in various medical imaging applications. This thesis aimed to develop DL-based models for fully automatic knee OA diagnosis using radiographic images. In this thesis, several methods for knee OA severity assessment and OA prediction are evaluated, and new methods are introduced. First, we focus on the feature-learning step as a crucial component of the classification system to learn and extract the most useful discriminative features from X-ray images. To do so, we introduce a novel autoencoder-based architecture called Discriminative Regularized Auto-encoder (DRAE). The goal is to maximize the distance between class features by minimizing the intraclass distance and maximizing the interclass distance. Then, we propose incorporating the proposed discriminative regularization in the standard Convolutional Neural Network (CNN) learning process to improve the early detection of knee OA. By doing so, we reduce the inability of CNNs to handle data with high inter-class similarities or high intra-class variations. While the proposed DRAE focuses on the texture information in the radiography under the tibial plateau, the second proposed learning model, called Discriminative Convolutional Neural Network.

Title : Applying Densely Connected Convolutional Neural Networks for Staging Osteoarthritis Severity from Plain Radiographs

Author: Norman, Berk Pedoia, Valentina Noworolski, Adam

Year : 2019

Osteoarthritis (OA) classification in the knee is most commonly done with radiographs using the 0–4 Kellgren Lawrence (KL) grading system where 0 is normal, 1 shows doubtful signs of OA, 2 is mild OA, 3 is moderate OA, and 4 is severe OA. KL grading is widely used for clinical assessment and diagnosis of OA, usually on a high volume of radiographs, making its automation highly relevant. We propose a fully automated algorithm for the detection of OA using KL gradings with a stateof-the-art neural network. Four thousand four hundred ninety bilateral PA fixed-flexion knee radiographs were collected from the Osteoarthritis Initiative dataset (age = 61.2 ± 9.2 years, BMI = 32.8 ± 15.9 kg/m2 , 42/58 male/female split) for six different time points. The left and right knee joints were localized using a U-net model. These localized images were used to train an ensemble of DenseNet neural network architectures for the prediction of OA severity. This ensemble of DenseNets' testing sensitivity rates of no OA, mild, moderate, and severe OAwere 83.7, 70.2, 68.9, and 86.0% respectively. The corresponding specificity rates were 86.1, 83.8, 97.1, and 99.1%. Using saliency maps, we confirmed that the neural networks producing these results were in fact selecting the correct osteoarthritic features used in detection. These results suggest the use of our automatic classifier to assist radiologists in making more accurate and precise diagnosis with the increasing volume of radiographic image being taken in clinic

Title: Automated Classification of Radiographic Knee Osteoarthritis Severity Using Deep Neural Networks

Author: Kevin A. Thomas, BSE • Łukasz Kidziński, PhD • Eni Halilaj, PhD • Scott L. Fleming, MS • Guhan R. Venkataraman, BS • Edwin H. G. Oei, MD, PhD • Garry E. Gold, MD, MS • Scott L. Delp, PhD

Year: 2020

Radiographs from the Osteoarthritis Initiative staged by a radiologist committee using the Kellgren-Lawrence (KL) system were used. Before using the images as input to a convolutional neural network model, they were standardized and augmented automatically. The model was trained with 32116 images, tuned with 4074 images, evaluated with a 4090-image test set, and compared to two individual radiologists using a 50-image test subset. Saliency maps were generated to reveal features used by the model to determine KL grade.

CHAPTER 3

SYSTEM STUDY

8.1 Aim and Objectives:

The primary aim of this project is to develop a convolutional neural network ensemble model for the classification of knee osteoarthritis. This approach seeks to improve diagnostic accuracy by combining multiple CNN architectures to leverage their individual strengths. The specific objectives include: (1) to collect and preprocess a comprehensive dataset of knee images, (2) to design and implement various CNN architectures, (3) to evaluate the performance of individual models and the ensemble method, and (4) to compare the results against traditional diagnostic methods to ascertain the improvements in classification accuracy.

8.2 **Goals and Scope**:

This research aims to develop a powerful and automated diagnostic tool for the early detection of knee osteoarthritis, with the potential to revolutionize clinical practices and significantly improve patient outcomes. The project focuses on creating an ensemble deep-learning model capable of delivering accurate and consistent results across diverse populations and imaging modalities, ensuring its usability in a wide range of clinical environments. Furthermore, the study emphasizes enhancing the interpretability of these models by providing clinicians with clear insights into the decision-making processes of convolutional neural networks (CNNs).

This aspect is crucial for building trust and acceptance of AI-powered diagnostics in medical practice, paving the way for more transparent and reliable healthcare innovations..

OUTLINE OF THE PROJECT

Overview of the system:

- Define a problem
- Gathering image data set
- Evaluating algorithms
- Detecting results

The steps involved in Building the data model is depicted below.

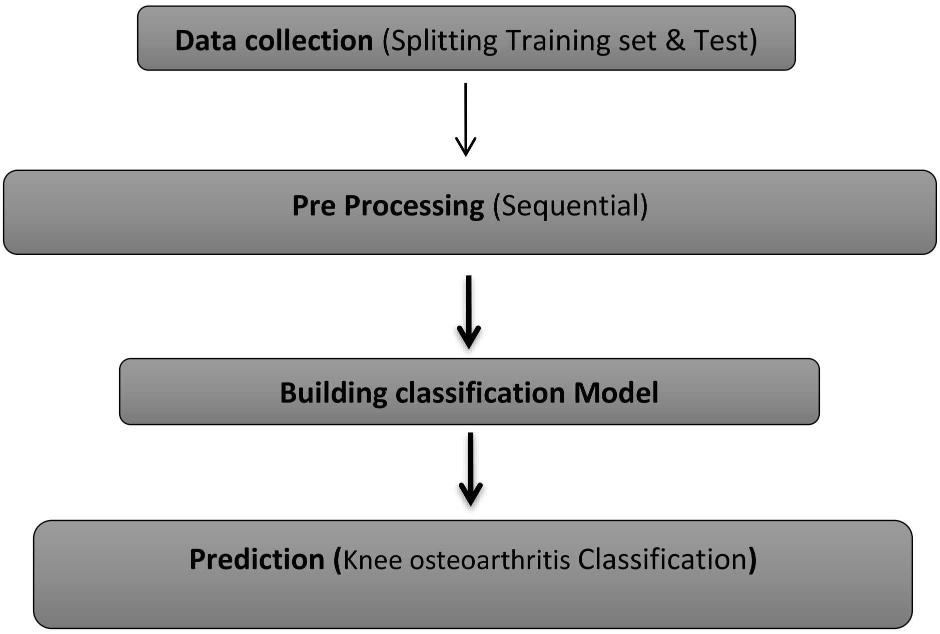


Fig 2: data flow diagram for CNN model

Project Goals:

- Load the data
- Loading the given dataset
- Import required libraries packages

- Pre-process the data
- Reshape, data augmentations

- Define model
- Sequential or Functional
- Number of layers to be used, Number of nodes to be used in the model, Evaluation metrics

- Compile the model
- Define loss function, optimizer, weights and bias

- Fit the model
- Train data, Test data, epoch, Batch size.

4 PROJECT REQUIREMENTS

General:

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements

2. Non-Functional requirements

3. Environment requirements

A. Hardware requirements

B. software requirements

9.1 Functional requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists the requirements of a particular software system. The following details to follow the special libraries like TensorFlow, keras and , matplotlib.

9.2 Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithm
4. Improving results
5. Prediction the result

Environment Requirements:

Framework : Keras.

Software Requirements:

- Operating System : Windows / Linux
- Simulation Tool : Anaconda with Jupyter Notebook
- Language : Python

Hardware requirements:

- Processor : Intel i3
- Hard disk : minimum 400 GB
- RAM : minimum 4 GB

5 FEASIBILITY STUDY

Splitting the dataset:

The data use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. It has the test dataset (or subset) in order to test our models and it will do this using Tensor flow library in Python using the Keras method.

Construction of a Detecting Model:

Deep learning needs data gathering have lot of past image data's. Training and testing this model working and predicting correctly.

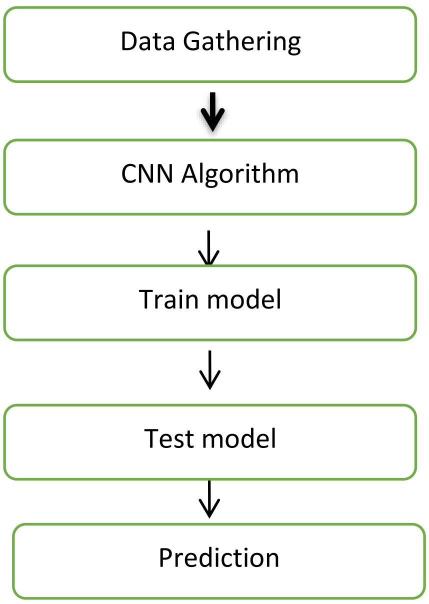


Fig 3: Steps of dataflow diagram

11. DESIGN ARCHITECTURE

General

Design is meaningful engineering representation of something that is to be built. Software design is a process design is the perfect way to accurately translate requirements in to a finished software product. Design creates a representation or model, provides detail about software data structure, architecture, interfaces and components that are necessary to implement a system.

11.1 Data Flow Diagram:

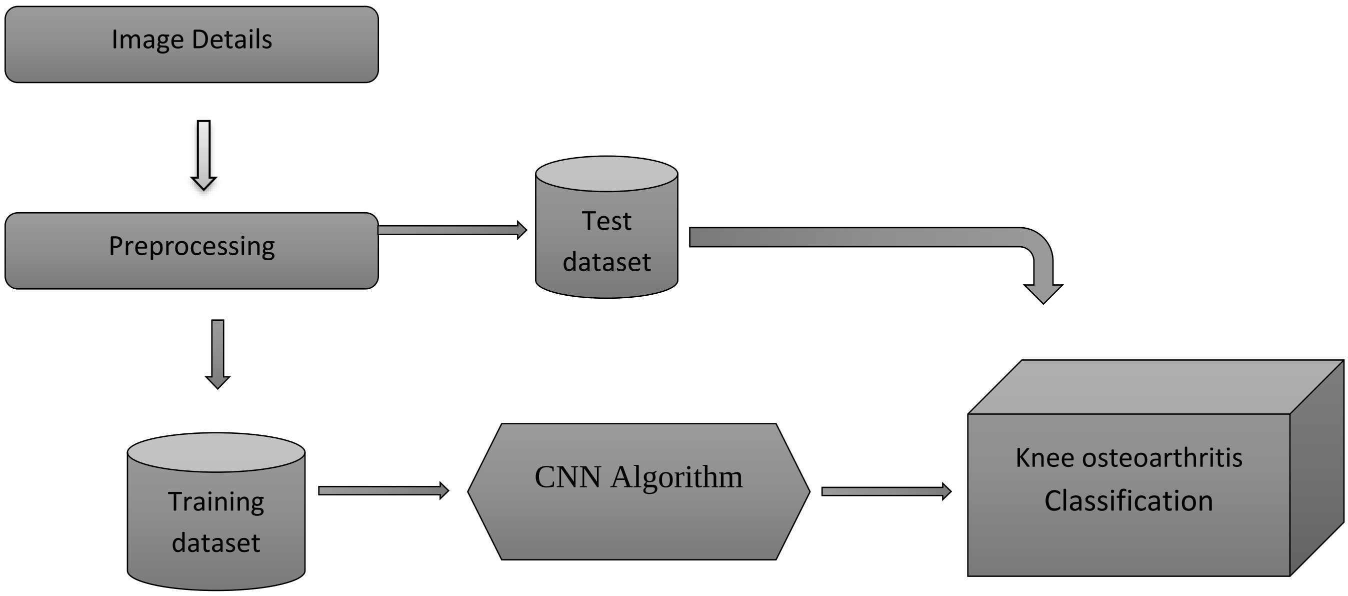


Fig 4: Process of dataflow diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model. Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top down approach to Systems Design. Symbols and Notations Used in DFDs Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

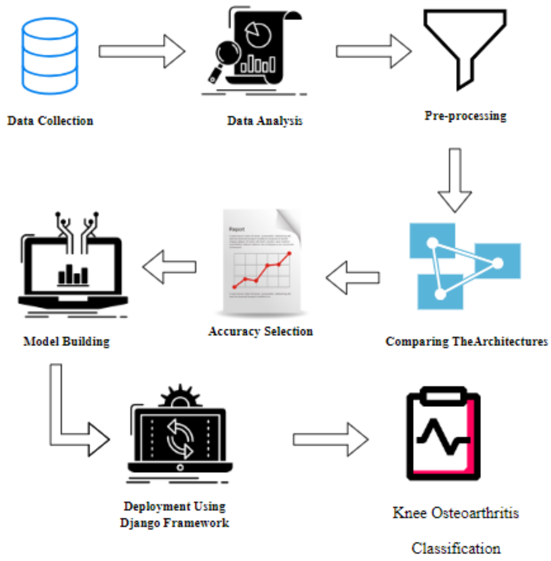
External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. Data store: files or repositories that hold information for later use, such as a database table or a membership form. Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like "Billing details."

DFD levels and layers A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish. DFD Level 0 is also called a Context Diagram.

It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

System Architecture:



11.2 Work flow diagram:

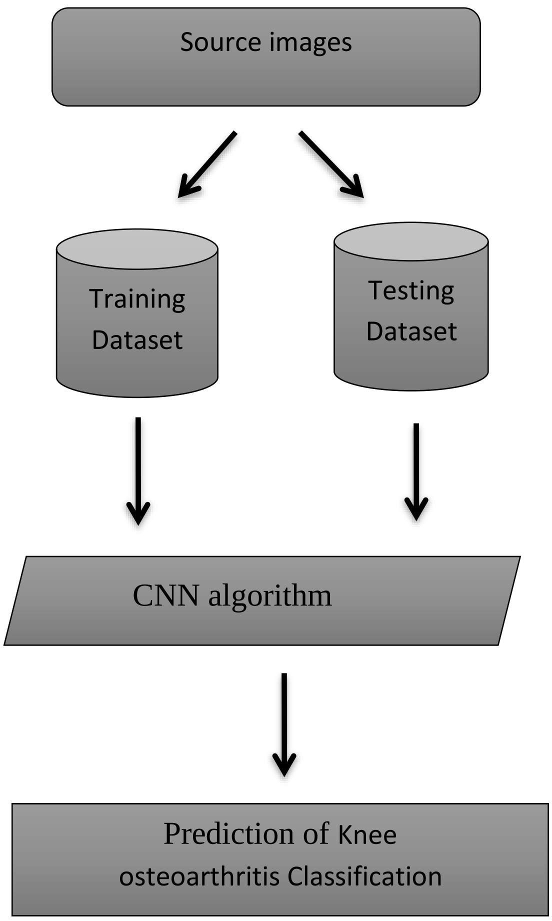
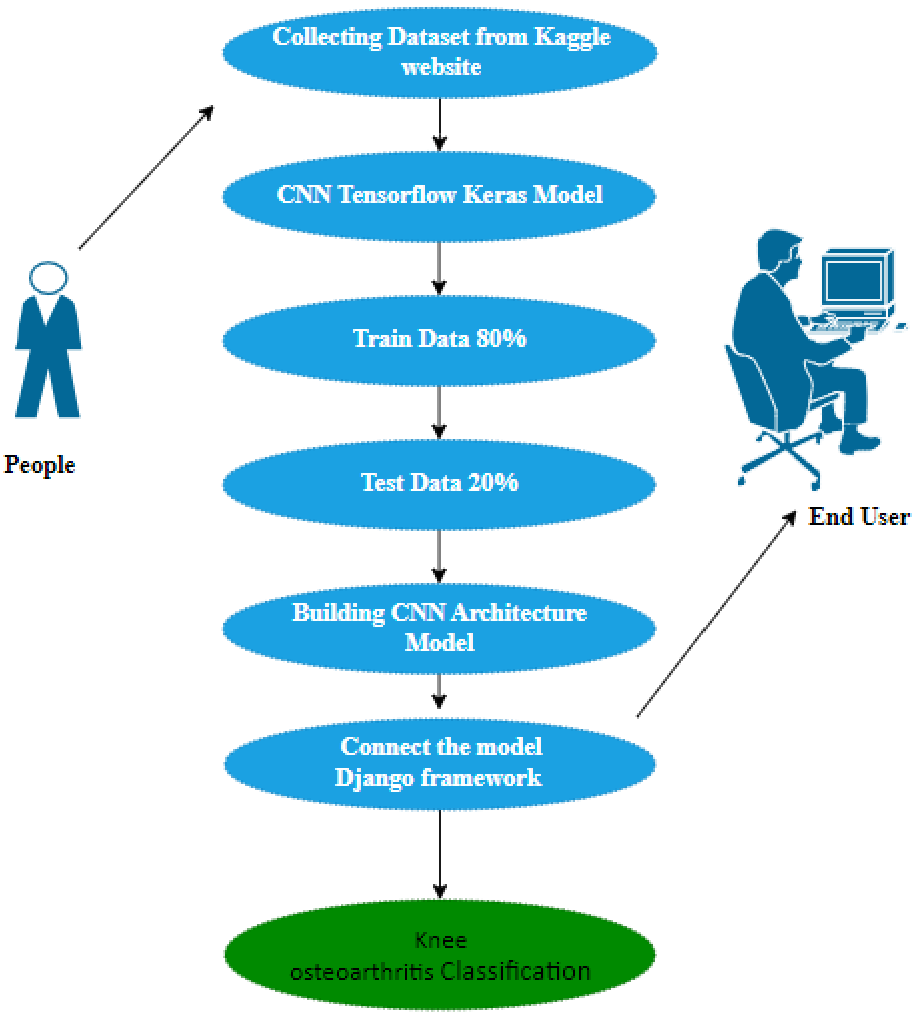


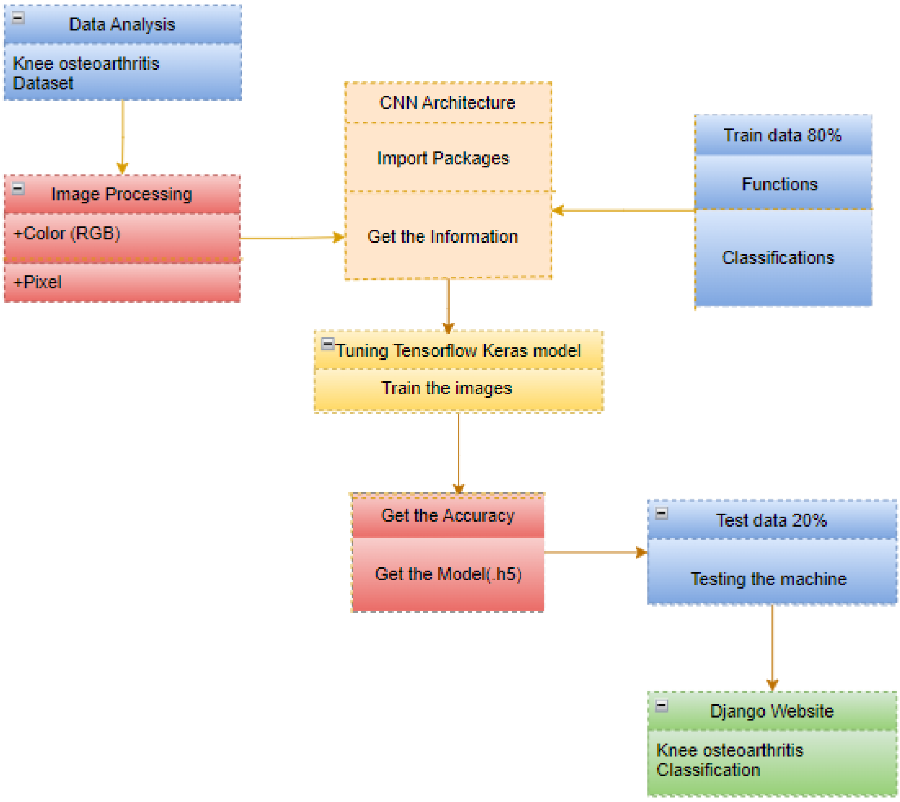
Fig: Workflow Diagram

11.3 USECASE DIAGRAM:



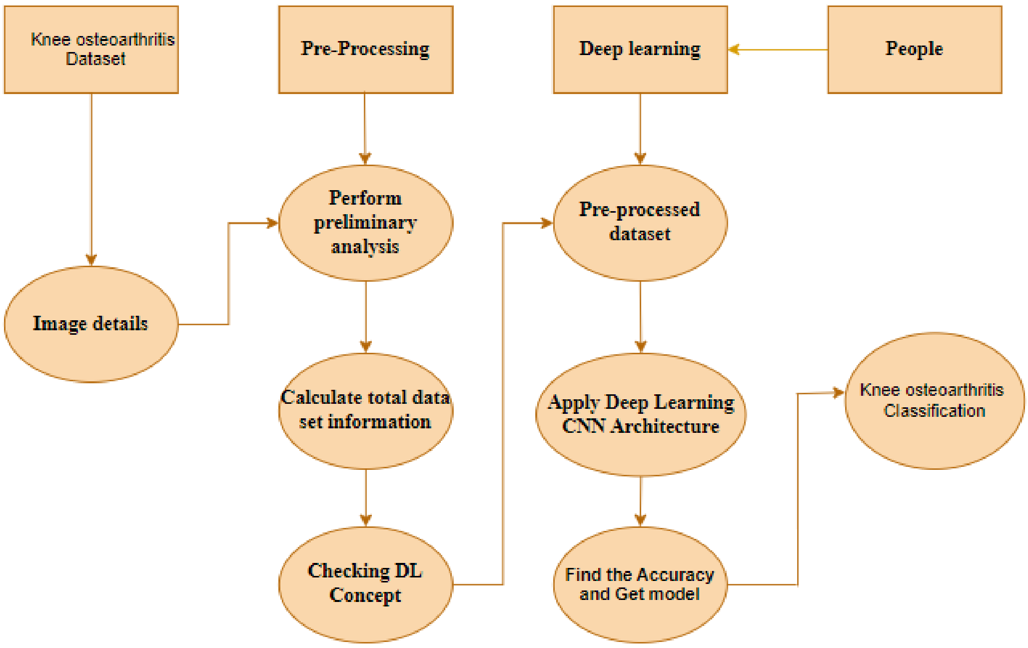
Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

11.4 CLASS DIAGRAM:



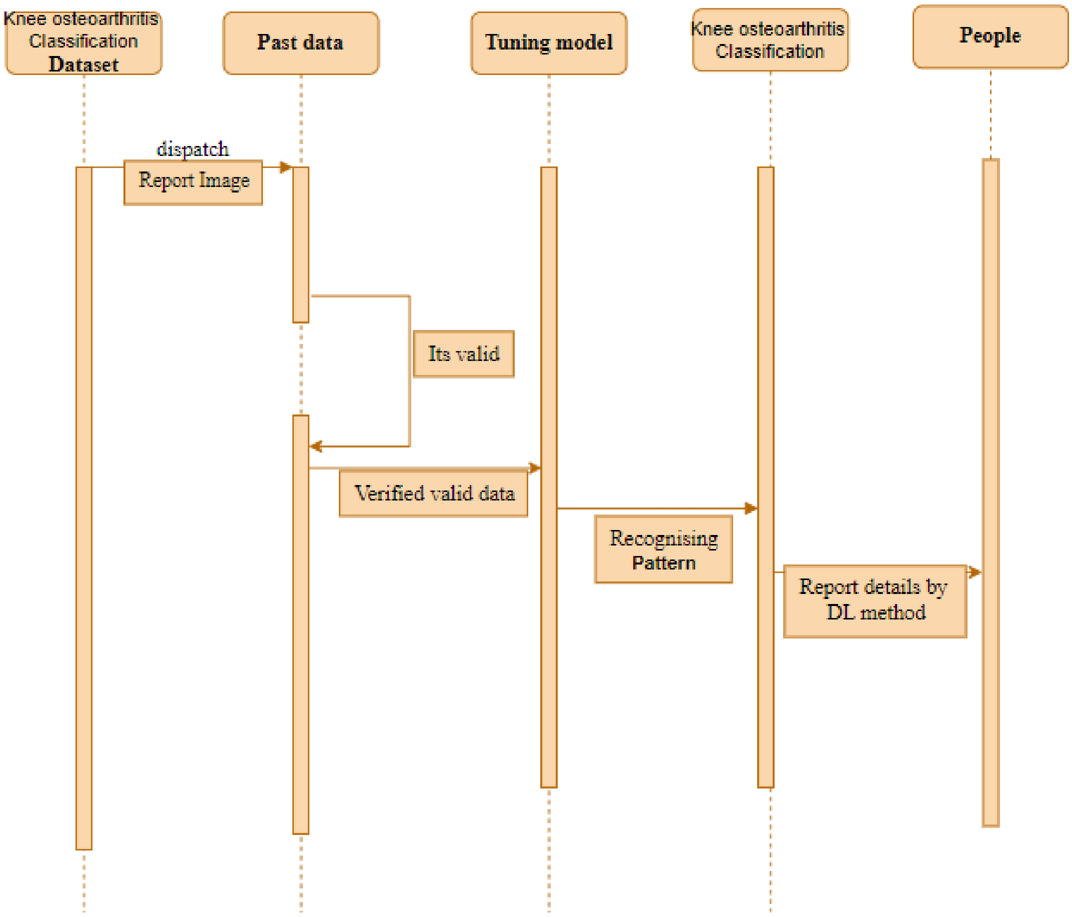
Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

11.5 ACTIVITY DIAGRAM:



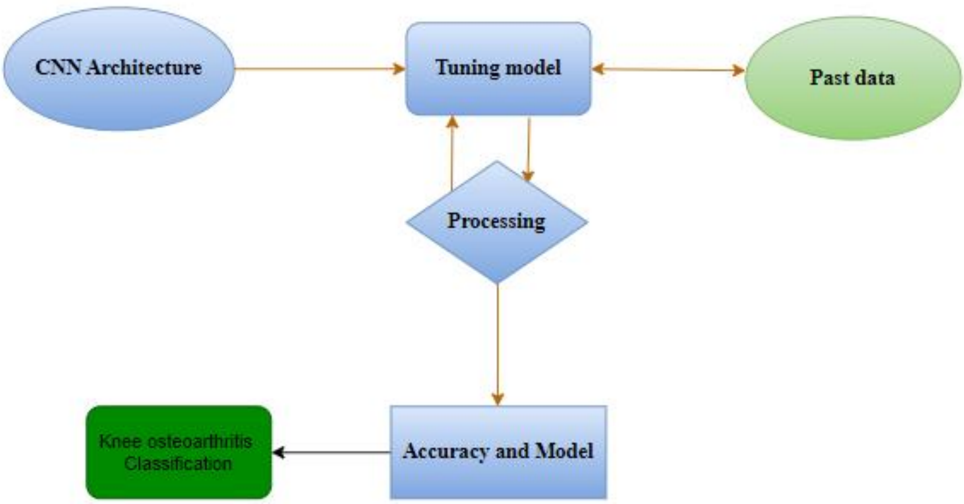
Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

11.6 SEQUENCE DIAGRAM:



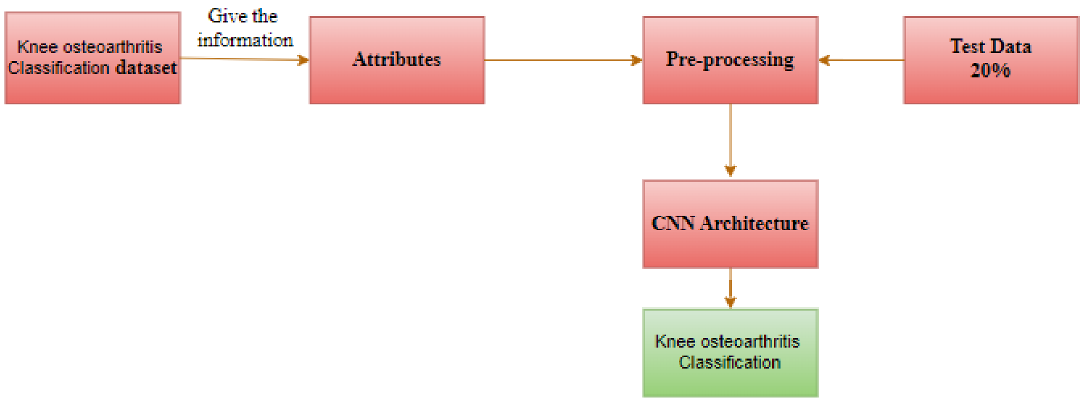
Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design-level models for modern business application development.

11.7 ER DIAGRAM:



An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

11.8 COLLABORATION DIAGRAM:



A collaboration diagram show the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction.

The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram.it can be the decomposition of a use case, use case diagram, or part of a use case diagram.

The collaboration diagram shows messages being sent between classes and object (instances). A diagram is created for each system operation that relates to the current development cycle (iteration).

12 SOFTWARE DESCRIPTION

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system "Conda". The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

12.1 ANACONDA NAVIGATOR:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Anaconda. Now, if you are primarily doing data science work, Anaconda is also a great option. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science.

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode
- Glueviz
- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution.Navigator allows you to launch common Python programs and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository.

Anaconda comes with many built-in packages that you can easily find with conda list on your anaconda prompt. As it has lots of packages (many of which are rarely used), it requires lots of space and time as well. If you have enough space, time and do not want to burden yourself to install small utilities like JSON, YAML, you better go for Anaconda.

11.2 JUPYTER NOTEBOOK:

This website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started.

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez.

Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

Installation: The easiest way to install the *Jupyter Notebook App* is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called **Anaconda**

Running the Jupyter Notebook

Launching *Jupyter Notebook App:* The Jupyter Notebook App can be launched by clicking on the *Jupyter Notebook* icon installed by Anaconda in the start menu (Windows) or by typing in a terminal (*cmd* on Windows): “jupyter notebook”

This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open.

When started, the Jupyter Notebook App can access only files within its start-up folder (including any sub-folder). No configuration is necessary if you place your notebooks in your home folder or subfolders. Otherwise, you need to choose a Jupyter Notebook App start-up folder which will contain all the notebooks.

Save notebooks: Modifications to the notebooks are automatically saved every few minutes. To avoid modifying the original notebook, make a copy of the notebook document (menu file -> make a copy…) and save the modifications on the copy.

Executing a notebook: Download the notebook you want to execute and put it in your notebook folder (or a sub-folder of it).

- Launch the jupyter notebook app
- In the Notebook Dashboard navigate to find the notebook: clicking on its name will open it in a new browser tab.
- Click on the menu *Help* -> *User Interface Tour* for an overview of the Jupyter Notebook App user interface.
- You can run the notebook document step-by-step (one cell a time) by pressing *shift + enter*.
- You can run the whole notebook in a single step by clicking on the menu *Cell* -> *Run All*.

- ❖ To restart the kernel (i.e. the computational engine), click on the menu *Kernel* -> *Restart*. This can be useful to start over a computation from scratch (e.g. variables are deleted, open files are closed, etc…).

Purpose: To support interactive data science and scientific computing across all programming languages.

File Extension: An IPYNB file is a notebook document created by Jupyter Notebook, an interactive computational environment that helps scientists manipulate and analyze data using Python.

JUPYTER Notebook App: The *Jupyter Notebook App* is a server-client application that allows editing and running notebook documents via a web browser. The *Jupyter Notebook App* can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents, the *Jupyter Notebook App* has a "Dashboard" (Notebook Dashboard), a "control panel" showing local files and allowing to open notebook documents or shutting down their kernels.

kernel: A notebook *kernel* is a "computational engine" that executes the code contained in a Notebook document. The *ipython kernel*, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels).
When you open a Notebook document, the associated *kernel* is automatically launched. When the notebook is *executed* (either cell-by-cell or with

menu *Cell* -> *Run All*), the *kernel* performs the computation and produces the results. Depending on the type of computations, the *kernel* may consume significant CPU and RAM. Note that the RAM is not released until the *kernel* is shut-down

Notebook Dashboard: The *Notebook Dashboard* is the component which is shown first when you launch Jupyter Notebook App. The *Notebook Dashboard* is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown).
The *Notebook Dashboard* has other features similar to a file manager, namely navigating folders and renaming/deleting files

Working Process:

- Download and install anaconda and get the most useful package for machine learning in Python.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- Machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelmingThe best way to get started using Python for machine learning is to complete a project.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird’s eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well-known steps:

- Define Problem.
- Prepare Data.
- Evaluate Algorithms.
- Improve Results.
- Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

13 MODULE DESCRIPTION

IMPORT THE GIVEN IMAGE FROM DATASET:

We have to import our data set using keras preprocessing image data generator function also we create size, rescale, range, zoom range, horizontal flip. Then we import our image dataset from folder through the data generator function. Here we set train, test, and validation also we set target size, batch size and class-mode from this function we have to train using our own created network by adding layers of CNN.

TO TRAIN THE MODULE BY GIVEN IMAGE DATASET:

To train our dataset using classifier and fit generator function also we make training steps per epoch's then total number of epochs, validation data and validation steps using this data we can train our dataset.

WORKING PROCESS OF LAYERS IN CNN MODEL:

DATA PREPROCESSING:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human disease and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

Input Layer:

Input layer in CNN contain image data. Image data is represented by three dimensional matrixes. It needs to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, it need to convert it into 784 x 1 before feeding into input.

Convo Layer:

Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field (it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then the filter over the next receptive field of the same input image by a Stride and do the same operation again. It will repeat the same process again and again until it goes through the whole image. The output will be the input for the next layer

Pooling Layer:

Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layers. If it applies FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive. So, the max pooling is only way to reduce the spatial volume of input image. It has applied max pooling in single depth slice with Stride of 2. It can observe the 4 x 4 dimension input is reducing to 2 x 2 dimensions.

Fully Connected Layer (FC):

Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training.

Softmax / Logistic Layer:

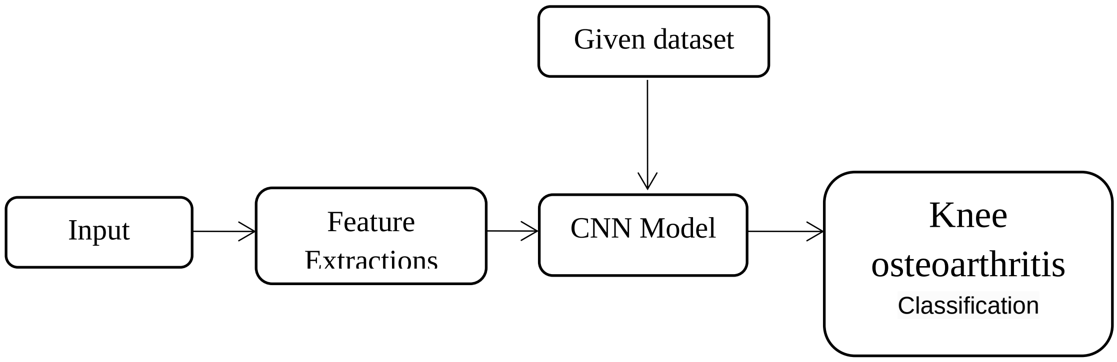
Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multi-classification.

Output Layer:

Output layer contains the label which is in the form of one-hot encoded. Now you have a good understanding of CNN.

KNEE OSTEOARTHRITIS CLASSIFICATION IDENTIFICATION:

We give input image using keras preprocessing package. That input Image converted into array value using pillow and image to array function package. We have already classified Knee osteoarthritis diseaseimage dataset. It classifies what are the Knee osteoarthritis disease. Then we have to predict our Knee osteoarthritis using predict function.



The hydroponic Knee osteoarthritis disease recognition method is based on a two-channel architecture that is able to recognize classification of

Knee osteoarthritis. The Classification images are used as the input into the inception layer of the CNN. The Training phase involves the feature extraction and classification using convolution neural network.

DATA VISULAIZATION:

Libraries Required:

- ✓ **tensorflow**: Just to use the tensor board to compare the loss and adam curve our result data or obtained log.
- ✓ **keras**: To pre-process the image dataset.
- ✓ **matplotlib**: To display the result of our predictive outcome.
- ✓ **os**: To access the file system to read the image from the train and test directory from our machines.

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions.

Let us now consider the following important features of TensorFlow −

- It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.
- It includes a programming support of deep neural networks and machine learning techniques.

- It includes a high scalable feature of computation with various data sets.
- TensorFlow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used.

Keras:

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Features

Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features –

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.
- It supports multiple platforms and backends.
- It is user friendly framework which runs on both CPU and GPU.
- Highly scalability of computation

Chapter 4

11 METHODOLOGY

In developing a Convolutional Neural Network (CNN) for knee osteoarthritis classification, the methodology involves several key steps. First, a dataset comprising medical images of knees, is collected and annotated to indicate the presence and severity of osteoarthritis. Preprocessing techniques, including image normalization, resizing, and augmentation, are applied to enhance the dataset and improve model robustness. A CNN architecture is then designed, typically featuring multiple convolutional layers followed by pooling layers to extract hierarchical features from the images. The model is trained using a suitable loss function and optimizer, such as categorical cross-entropy and Adam optimizer, to minimize classification errors on a training set while validating performance on a separate validation set. Hyperparameter tuning is conducted to optimize model performance, and techniques such as dropout and batch normalization are incorporated to prevent overfitting. Finally, the model is evaluated on a test set, and metrics like accuracy, precision, recall, and F1-score are calculated to assess its efficacy in accurately classifying knee osteoarthritis stages, ultimately providing a reliable tool for aiding diagnosis and treatment planning.

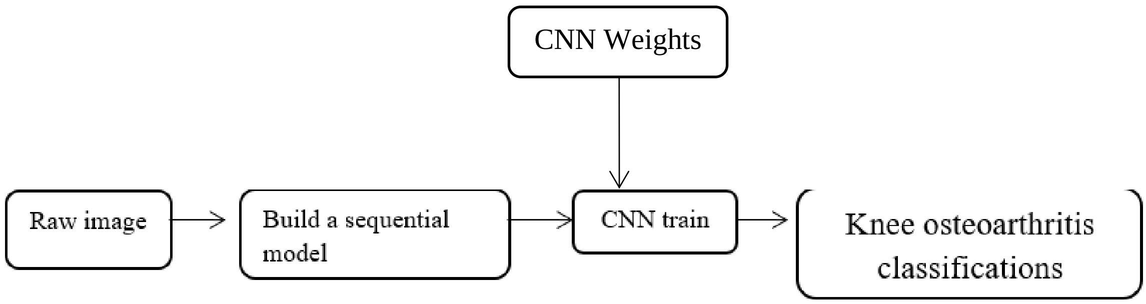


Fig 5. Methodology for developing CNN for knee osteoarthritis classification

The train dataset is used to train the model (CNN) so that it can identify the test image and the disease it has CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D, and MaxPooling2D. After the model is trained successfully, the software can identify the Knee osteoarthritis classifications Classification image contained in the dataset. After successful training and preprocessing, comparison of the test image and trained model takes place to predict the Knee osteoarthritis classifications.

12 ARCHITECTURE OF CNN

CONVOLUTIONAL NEURAL NETWORK:

A Convolutional neural network (CNN) is one type of Artificial Neural Network. A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data.

13 TYPES OF CNN:

XCEPTION NET:

"Xception" is another deep learning architecture designed for image classification tasks. It was proposed by François Chollet in the research paper titled "Xception: Deep Learning with Depthwise Separable Convolutions," published in 2017. The term "Xception" is derived from the words "Extreme Inception," as it builds upon the ideas introduced in the Inception architecture. The Xception network is inspired by the Inception architecture, which uses multiple convolutional filters of different sizes to capture features at various scales. However, instead of using traditional convolutions, Xception employs depthwise separable convolutions, which are more computationally efficient.

Here are the main features of the Xception architecture:

Depthwise Separable Convolutions: Traditional convolutions involve applying a large number of filters to input feature maps, resulting in a high computational cost. Depthwise separable convolutions break down the standard convolution operation into two separate steps: depthwise convolution and pointwise convolution.

Depthwise Convolution: In this step, each channel of the input feature map is convolved with its own set of filters independently. It means that each channel is processed individually without mixing information from other channels. Pointwise Convolution: This step applies a 1x1 convolution to combine the outputs of the depthwise convolution and generate the final output feature map. Pointwise convolutions help to mix and combine the information from different channels effectively. By using depthwise separable convolutions, Xception reduces the number of parameters significantly compared to traditional convolutions while maintaining a comparable level of accuracy.Skip Connections: Xception also includes skip connections or shortcut connections in the network. These connections allow gradients to flow directly between non-adjacent layers during training. Skip connections help in mitigating the vanishing gradient problem and make it easier to train

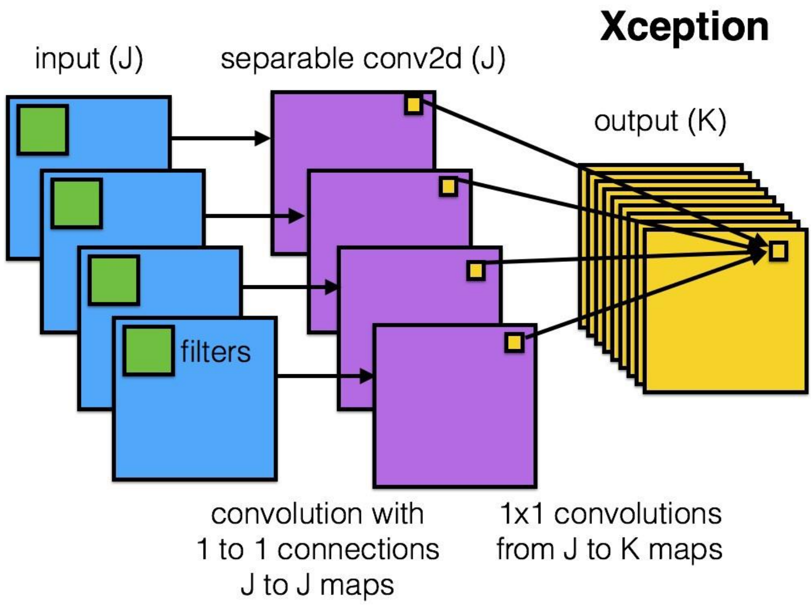


Fig. 6 Xception

2) Shuffle Net Architecture

1. Background:

- ShuffleNet is developed to address the limitations of existing deep learning models on resource-constrained devices.

- It focuses on reducing the number of parameters and computation, allowing for faster inference times.

2. Key Features:

- Group Convolutions: The architecture employs group convolutions to limit the number of connections between layers. This reduces computational complexity significantly while maintaining feature representation.

- Channel Shuffle: To improve the information flow between the grouped convolutions, ShuffleNet introduces a channel shuffle operation. This operation rearranges the output of grouped convolutions, allowing for better feature reuse.

- Lightweight Design: The architecture is designed to be lightweight, making it suitable for applications on mobile devices.

3. Architecture Components:

- Input Layer: The network typically takes RGB images as input.

- Initial Convolution Layer: A standard convolution layer with a stride of 2, followed by batch normalization and a ReLU activation function.

- ShuffleNet Units: The core building block of the architecture is the ShuffleNet unit, which consists of:

  - Pointwise Convolution: A 1x1 convolution that reduces the number of channels.

- Group Convolution: A 3x3 group convolution that performs convolutions on the reduced channel set.

- Channel Shuffle: An operation that rearranges the channels to ensure better mixing of features across different groups.

- Skip Connections: To enhance gradient flow and mitigate vanishing gradients, skip connections are used between the input and output of the unit.

- Downsampling: Downsampling layers are included in the architecture to reduce the spatial dimensions and increase the abstraction level.

- Fully Connected Layer: At the end of the architecture, a fully connected layer is used for classification, followed by softmax activation to produce class probabilities.

4. Benefits:

- Efficiency: ShuffleNet is efficient in terms of computation, making it suitable for real-time applications on low-power devices.

- Performance: Despite its lightweight nature, ShuffleNet achieves competitive performance on standard image classification benchmarks like ImageNet.

5. Variants:

- ShuffleNet has several variants (e.g., ShuffleNet V2) that improve upon the original design by optimizing the channel shuffle mechanism and reducing latency.

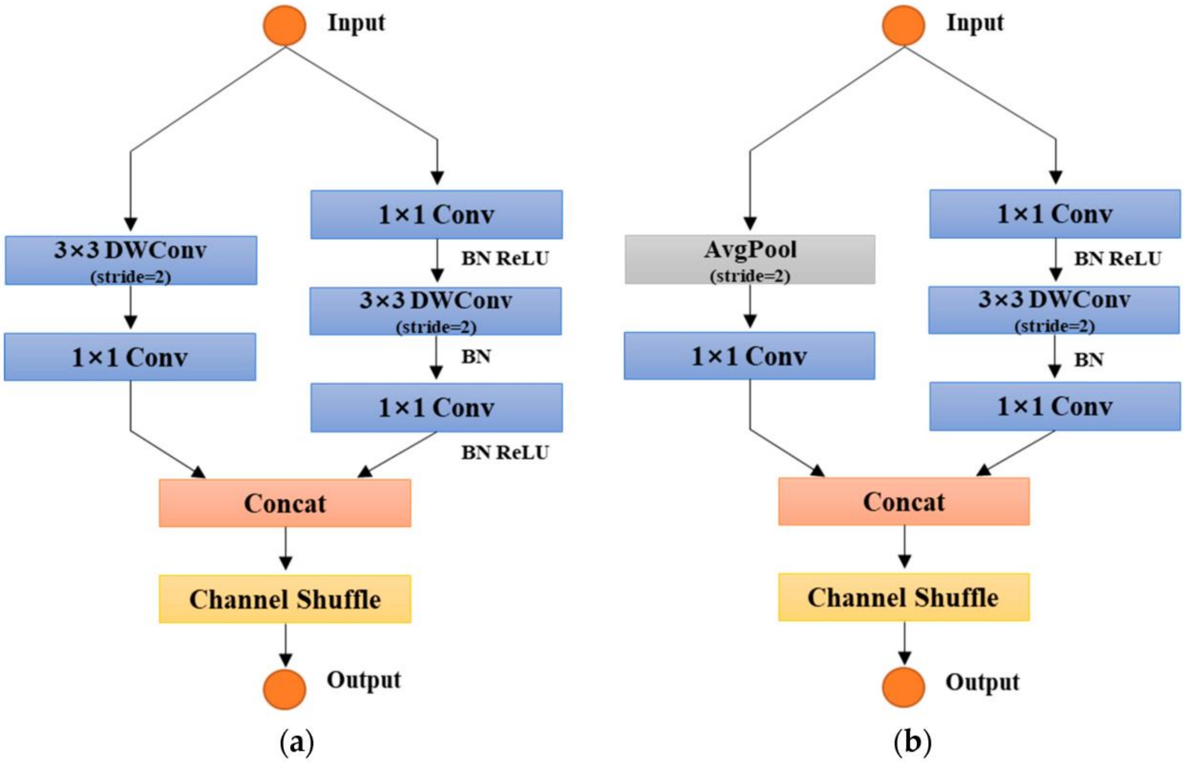


Fig. 7 : Shuffle Net Architecture

LIST OF MODULES:

1. Manual Net

2. Xception Net

3. Shuffle Net

4. Deploy

14 DEPLOY

Deploying the model in Django Framework and predicting output

In this module the trained deep learning model is converted into hierarchical data format file (.h5 file) which is then deployed in our django framework for providing better user interface and predicting the output.

18 Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

CHAP 5

19. Conclusion:

The proposed convolutional neural network ensemble for knee osteoarthritis classification aims to enhance diagnostic accuracy and efficiency, addressing the limitations of traditional methods. By integrating multiple CNN architectures, this project aspires to achieve superior performance and reliability in OA detection, ultimately contributing to better patient management and outcomes.

20. Future Work:

Future research will focus on refining the ensemble model by incorporating additional data sources, such as clinical parameters and patient demographics, to further enhance predictive capabilities. Furthermore, ongoing efforts will be directed towards real-world implementation of the model in clinical settings, ensuring that it meets regulatory standards and can be integrated into existing healthcare workflows. Longitudinal studies will also be pursued to evaluate the model's performance over time and its impact on patient management strategies in knee osteoarthritis.

Prediction of the 8 stages of osteoarthritis using deep learning.

Dr. Kumaragurubaran
School of Computer Science and Engineering
Rajalakshmi engineering College
Chennai, India
kumaragurubaran.t@rajalakshmi.edu.in

Abstract - This paper proposes a novel classification approach for knee osteoarthritis using an ensemble of CNNs that utilize TensorFlow. Osteoarthritis is a common degenerative joint disease often causing pain, limited mobility, and decreased quality of life, mainly in elderly individuals. Early and accurate diagnosis is crucial for timely intervention and efficient treatment planning. Our approach combines different CNN architectures into an ensemble model, which improves the feature extraction from knee radiographs more than the traditional single-CNN models. The ensemble model can capture a variety of patterns from images and could suppress overfitting because each CNN is trained on a different subset of the dataset. Advanced ensemble techniques, such as bagging and boosting, also enhance the model's accuracy and robustness. The proposed model was validated on a diverse set of knee radiographs that represent a spectrum of OA severities. Results show an improvement in classification accuracy, which indeed surpasses the traditional methods and, in some cases, even expert radiologists' assessments. Thus, this piece of research for the first time highlights the usability of CNN ensembles in medical imaging. The study underlines the importance of machine learning in furthering healthcare, as it provides the foundation for AI-supported systems to support clinical diagnosis and enable improvement in patient outcomes

Keywords- KneeOstheo arthiritis, Machine Learning, Deep Learning, Random Forest, Gradient Boosting, Bagging Regressor, RNN (Recurrent Neural Network), Stacking, Regressor, Meta-learner, One Hot Encoding, Normalization

I. INTRODUCTION

Knee OA is a common chronic degenerative joint disease affecting millions worldwide and significantly impairs quality of life. Characterized by cartilage breakdown, OA presents with symptoms such as pain, stiffness, and decreased mobility. Early diagnosis is what makes appropriate management possible. Without the previous diagnostic methods, which either rely on subjective examinations or radiographic imaging, variability in assessment and delays often occurred by their reliance on subjective assessments, which are prone to variability among practitioners and interpretation errors. These inconsistencies can delay diagnosis and compromise the development of optimal treatment plans.

Deep learning, especially CNNs, provides a more revolutionary answer. CNNs rely on vast numbers of medical image sets to recognize and classify patterns much more accurately and reproducibly, with considerably less dependence on subjective appraisals. This ability enables CNNs to detect knee OA earlier and much more reliably through its use of a standardized and reproducible diagnostic process. This shows that artificial intelligence may be a potential game-changer in medical diagnostics, particularly in musculoskeletal diseases. Incorporating CNN-based methodologies is a milestone toward enhancing the accuracy of diagnosis and patient care in knee OA treatment.

II. RELATED WORKS

The existing sEMG-PRS system has low generalizability for its use in practical applications. Here, we propose the idea of a SWRF algorithm to improve the long-term usability of the sEMG-PRS and user adaptability. First, we come up with the idea of WRF to abolish the drawback of the standard RF. By standard RF, there is a random issue while taking samples and features. Stacking is applied further to improve the generalization capability of WRF. RF is used as a base learner and WRF is employed as a meta-learning algorithm layer. The experimental results of SWRF is compared with some popular classification algorithms that are both online experiments and offline datasets. Offline experiments show that the average classification accuracy of SWRF is about 89.06%, which is higher than those of RF, WRF, long short-term memory (LSTM), and support vector machine (SVM). Online experiments reveal that the proposed SWRF surpasses the above algorithms concerning long-term usability and user adaptability. We consider that our method has good prospects for practical use in sEMG-PRS. The developed surface electromyography-based pattern recognition system (sEMG-PRS) has some principal drawbacks which prevent it from being implemented effectively in real-life applications. Surface electromyography-based pattern recognition systems (sEMG-PRS) face several critical limitations that significantly hinder their effectiveness and reliability in real-world practical applications. One of the major problems is the insufficient generalization capability because these systems tend to fail in demonstrating generalized performance by various user conditions, especially with dynamic usage environments. Such a limited capability of these systems is problematic in uncontrolled, real-world scenarios, where muscle signals and other environmental factors cause inconsistencies in outputs, thus largely hindering practical applicability. Another critical drawback is the imbalance in standard RF performance. RF is a very common used algorithm within sEMG-PRS. As it shows significant randomness in its sampling and feature selection algorithms, the models often suffer from a fluctuations in the classification accuracy levels, especially when handling high dimensional and complex data.

In addition to the above, a number of critical limitations accompany sEMG-PRS systems that considerably dampen their real-world application strength and reliability. A principal factor relates to generalizability: these systems are often unable to deliver consistent performance across various user conditions and dynamic usage environments. Such a limitation severely affects the usability of the system in uncontrolled, real-world situations, as variations in muscle signals and environmental factors may cause a lack of consistency in the output. Another significant issue that arises with standard RF performance is the imbalance in performance, the most commonly used algorithm in sEMG-PRS. Owing to the inherent nature of randomness to both sampling and selection of features in an RF model, these models generally show variability in their classification accuracy, mainly for high-dimensional and complex datasets.

An additional concern regarding the long-term usage of sEMG-PRS is its adaptability. Muscle signals in subjects may change over time due to effects like fatigue, physiological changes, or environmental factors. Current schemes do not adapt to these dynamic variations and, thus, their usability and reliability decrease. These adaptability challenges are exacerbated by the fact that existing systems were not able to realize the full potential of optimal classification accuracy, especially when compared to more advanced algorithms capable of exploiting complex patterns in sEMG data. The current solutions thus did not meet the required level of personalization, robustness, or consistent performance for practical, long-term use

To this end, we propose a novel methodology that can augment the generalization ability, adaptability, and overall performance of sEMG-based PRS systems. The proposed technique introduces a stacked weighted random forest (SWRF) algorithm, combining the benefits of WRF with the strengths of ensemble learning to overcome shortcomings in standard RF-based systems. WRF's algorithm helped to overcome these imbalances in performance by tackling the randomness involved with sampling and feature selection, making the whole process more stable and reliable for classification. For improving generalization, an approach known as stacking is proposed. Random forests become the base learners, while WRF forms the meta-learning layer. In this manner, the design becomes hierarchical and allows the SWRF algorithm to combine the benefits of each approach and increase its adaptability and robust performance.

The SWRF methodology is assessed rigorously by the method of using a combination of both offline datasets and online experiments. Offline testing proves that the average classification accuracy achieved by the SWRF is 89.06%, surpassing the performances of the conventional methods: standard RF, WRF, long short-term memory networks (LSTM), and support vector machines (SVM). The online experiments further validate the superiority of the SWRF algorithm for long-term use and adaptability of these traditional algorithms. In this context, the research proposed herein yields an important step forward in developing systems that can execute reliably in real-world settings and open the door for more practical and personalized applications of sEMG pattern recognition technology.

Another significant concern is the adaptability of the sEMG-PRS for long-term use. Users' muscle signals change due to a variety of factors that can occur over time, including fatigue, physiological variations, or even environmental influences. Current systems fail to accommodate these dynamic changes adequately, resulting in poor usability and lower reliability. The shortcomings in adaptability are further exacerbated by the limitations of existing systems in reaching the best possible classification accuracy, particularly in comparison with the performance of more sophisticated algorithms that can utilize intricate patterns that exist in sEMG signals. Thus, practical, long-term deployment requires solutions that better meet the requirements for more personalized, more robust, and predictable performance..

To address these challenges, we are proposing a novel methodology that can improve the generalizability, adaptability, and overall performance of sEMG-PRS. In particular, the proposed study introduces the SWRF algorithm, combining the strengths of WRF and ensemble learning techniques to overcome some of the main limitations of standard RF-based systems. The WRF algorithm controls performance imbalances by handling the randomness in sampling and feature selection, furthering stability and reliability in classification. To further enhance generalizability, a stacking approach is implemented. The base learners happen to be random forests, while WRF functions as the meta-learning layer. The hierarchical design lets the SWRF algorithm make use of both approaches for optimal adaptability and robust performance.

The proposed SWRF methodology is thoroughly tested by a combination of online experiments and offline testing. Offline experiments show that an average classification accuracy is obtained by SWRF to be 89.06%, surpassing the conventional methods like standard RF, WRF, LSTM, and SVM. Online experiments further test its superiority over these traditional algorithms in terms of the long term usability and adaptability. This work puts forward a significant step in systems toward reliable, real-world performance and lays the door open to more practical and personalized applications of sEMG pattern recognition technology reliable, real-world performance, paving the way for more practical and personalized applications of sEMG pattern recognition technology.

One major weakness is its lack of generalizability: sEMG-PRS is often not robustly reliable over diverse user conditions and varied usage environments. This potentially leads to inconsistent outputs, especially in real-world applications outside controlled laboratory settings. A second problem is that standard random forests (RF) used in such systems tend to have biased performance. Because the RF inherently operate based on random sampling and feature selection, these models tend to be unstable with respect to their classification accuracy, especially for complex, high-dimensional data. Moreover, such systems can be problematic in terms of long-term usability and adaptability because they fail to change when the users' muscle signals change over time, thus impairing the system's reliability and personalization.

Finally, the current methods lack optimal classification accuracy and are sometimes compared with more sophisticated approaches that take advantage of complex patterns in data and maintain performance. All of these limitations emphasize the need for a more robust, adaptive, high-performing solution for reliable use with sEMG-PRS in the long term.

III. Proposed Methodology

A. Problem Definition:

Surface electromyography-based pattern recognition systems generally have some significant limitations when it comes to performing reliably and effectively under practical, long-term scenarios. Limited generalizability is one of the primary problems with most such systems, which do not usually produce consistent results upon testing across various user conditions or in variable environments. Because of a lack of robustness on these lines, such systems are less suitable for real-world applications where dynamic and unpredictable factors frequently come into play. The use of standard random forests as the backbone classification algorithm is also one of the important factors causing the problem. Even though the RF technique is very popular, its randomness in both sampling and feature selection inherently causes imbalance in performance, leading to fluctuations in classification accuracies. Hence, it is, especially in handling complex high-dimensional datasets, further reduced the reliability of these systems.

Another important limitation of current sEMG-PRS is their limited ability to adapt flexibly to changes in users' muscle signals over time. Variations due to factors such as fatigue, physiological differences, or environmental changes often have a huge influence on the performance of such systems. Since such capacity to adapt to these evolving conditions does not exist, the long-term usability and reliability of such systems are compromised. These approaches are quite impractical and often result in less than ideal classification accuracy, so they cannot be compared to more advanced algorithms that can exploit complex patterns present in the data.

Therefore, there is a strong need for a robust, adaptable algorithm that has a high potential of overcoming these challenges. Such an algorithm should handle problems such as poorly balanced performance and accuracy inconsistency to guarantee the ability of the system to adapt dynamically to different user conditions over time. On top of improving classification accuracy, adaptability, and ensuring long-term usability, advancing the practical application of sEMG-PRS is primarily aimed at making these systems viable for real-world usage and well-performing, reliable, and tailored personal performance.time. Enhancing classification accuracy, improving adaptability, and ensuring long-term usability are essential goals for advancing the practical application of sEMG-PRS, ultimately enabling these systems to meet the demands of real-world usage and deliver reliable, personalized performance.

B. Dataset:

Importing the dataset

To prepare our image dataset for deep learning, we used the powerful tool Keras, allowing data augmentation and preprocessing. We applied such techniques as resizing, rescaling, rotation, zooming, and horizontal flipping, which artificially increase the size of the dataset and add robustness and generalizability to our model.

We imported the dataset of the images directly and read it from a particular folder. The training, validation, and testing data generators were configured separately. The parameters for target image size, batch size, and class mode were specified:

With the dataset pipes well in place, we trained a custom-designed convolutional neural network. The network architecture consisted of convolutional layers used to extract features, pooling layers further reducing the dimensionality, and fully connected layers for classification. By effectively leveraging both augmented data as well as carefully designed architectures, the network was reasonably and finely tuned within the preprocessed dataset to achieve optimal performance.

Training data set:

To train our dataset, we utilize a classifier in conjunction with the fit generator function. This involves defining the number of training steps per epoch, specifying the total number of epochs, and providing validation data along with validation steps. By iterating through these configurations, the model is exposed to a comprehensive dataset, ensuring better generalization and performance. These parameters allow for an efficient and structured training process, accommodating larger datasets and augmentations. Using this approach, the model progressively improves its accuracy and robustness across both training and validation phases.

C. System Architecture

The system architecture for the proposed knee osteoarthritis classification using an ensemble of CNNs consists of a number of crucial layers. As explained in Fig 1, In the Input Layer, images of knee radiographs obtained from medical datasets are preprocessed to ensure homogeneity in size, orientation and resoltuon

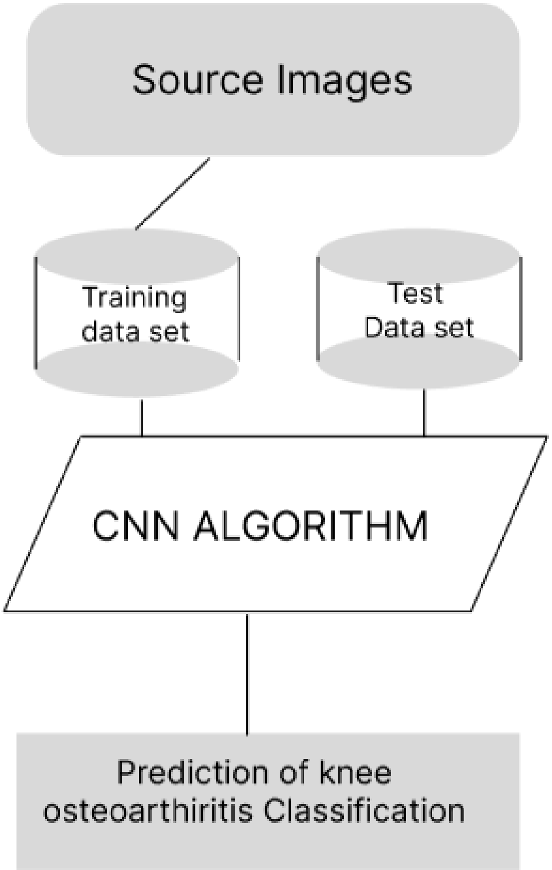


Figure 1: Architectural Workflow

Data augmentation techniques, such as rotation, scaling, and flipping, are applied to increase the variety of images, thereby improving the generalization capability of the model. CNN Ensemble Layer The CNN Ensemble Layer makes use of a variety of different CNN models with different architectures to be able to obtain features across a wide range. For instance, Model 1 captures higher level patterns using certain sizes of kernels and associated pooling strategies, Model 2 uses another configuration of convolution and pooling layers to obtain complementary features, and Model 3 further diversifies feature extraction using unique configurations of layers. Each CNN model comprises convolutional layers

In the Ensemble Layer, as shown in fig.2, outputs of each CNN model are combined through methods such as majority voting or weighted averaging, providing an aggregation of predictions into a single, far more accurate classification output. Ensembles increase robustness and minimise any weak points of a single model. The Classification Layer takes this aggregated output to supply a final prediction on the existence and severity of knee osteoarthritis, which helps in diagnosis. In making the Django Web Application Interface accessible to users, it allows users' health professionals to upload images, view predictions, and diagnostic insights. Moving forward, the Deployment Layer.

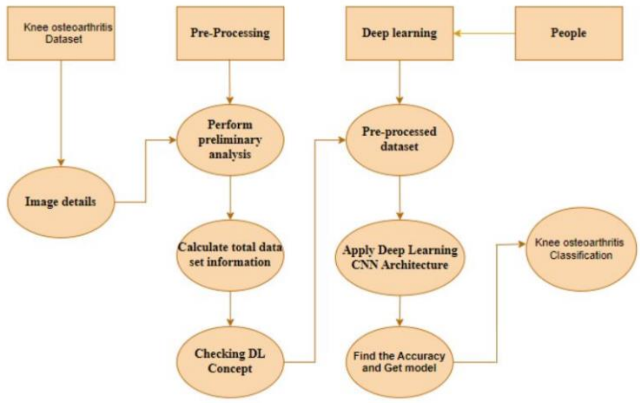


Figure 2: System architecture

As pointed out in Fig.3, ensures that the system is scalable using distributed computing by TensorFlow, therefore well-suited for large-sized datasets in different clinical environments. Secure data storage further enhances the integrity of the data and allows for further in-depth analysis if needed. Added advanced user authentication and data encryption mechanisms ensure meeting healthcare data privacy standards, thus improving the reliability and trustworthiness of the system. This architecture provides a complete, robust, and user-friendly system optimized for real-world clinical use. The CNN ensemble model aims through these advancements not only to set a new benchmark in knee OA classification but also contribute to the broader adoption of AI-driven tools in healthcare.

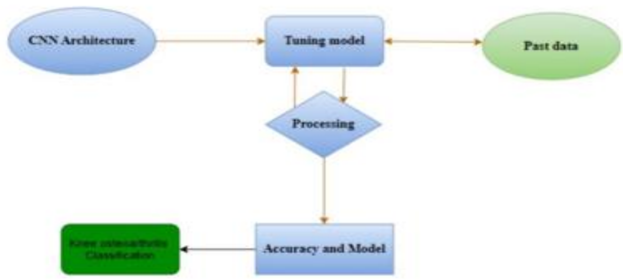


Figure 3: Model Architecture

IV. RESULTS AND DISCUSSION

A. *Experimental* Setup:

This project was developed on a Windows 11 system with Ryzen 5 3500U processor, 8GB RAM, and 512 GB SSD. The application is built using Python 3.11 and Jupyter Notebook from Anaconda 3. The model creation and selection were carried out with the help of the scikit-learn library, while the framework of building the web application was provided through APIs and HTML/CSS with Flask.

B. Observations:

Analyzing the dataset yields several key points. For instance, for numerical features, the patients with knee osteoarthritis had an average age of about 55, and the average BMI of the entire cohort was about 29.4, which indicates a very high proportion of overweighed individuals. In the dataset, there are records of 101 different patients and 10 key biomarkers, among which joint stiffness appeared to be the most common symptom.

Analyzing the progression patterns, high BMI and age were highly associated with the severity of osteoarthritis in which those patients classified as overweight had the faster progression rate. In addition, cartilage degeneration was seen across all ages but proceeded at a faster rate in those who had joint injury histories. Between 2010 and 2020, a drastic increase in the prescription rates of NSAIDs for pain control was noted, as can be seen in Figure 3.

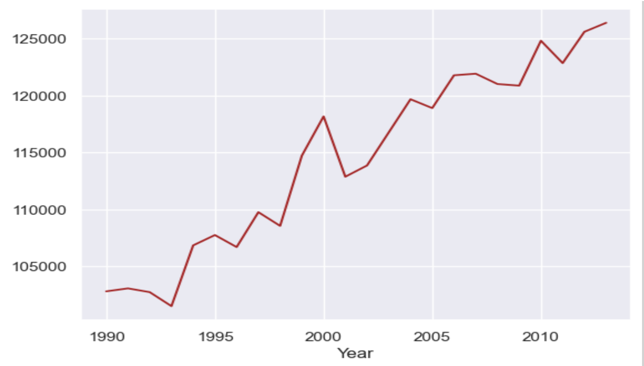


Figure 4: Increasing Cases of knee arthritis

As shown in Fig.4, In addition to knee osteoarthritis, even more thorough examinations of data showed a drastic increase in the number of cases reported in India while the list of reporting countries includes France, Germany, and Japan. There is a high rate of prescription with anti-inflammatory medication ranging between 1,00,000 and 1,30,000 in doses per year.

Table 1: Comparison table for different models

According to experimental findings, Random Forest stacked ensemble model of Gradient Boosting, Bagging and RNN got the lowest RMSE of 9274.09 and best R-squared value of 0.989, making it most accurate to the given dataset.".

It can be also seen that good performance values have been attained from the base models such as Random Forest, Bagging Regressor & Gradient Booster individually, hence a stacked ensemble of these models has been taken along with a deep learning RNN model as meta-learner.

It makes the architecture much more extensible and versatile with ostheo arthiritis prediction performed minimum variance and maximum adaptability, which does not exist in using classic learning models.

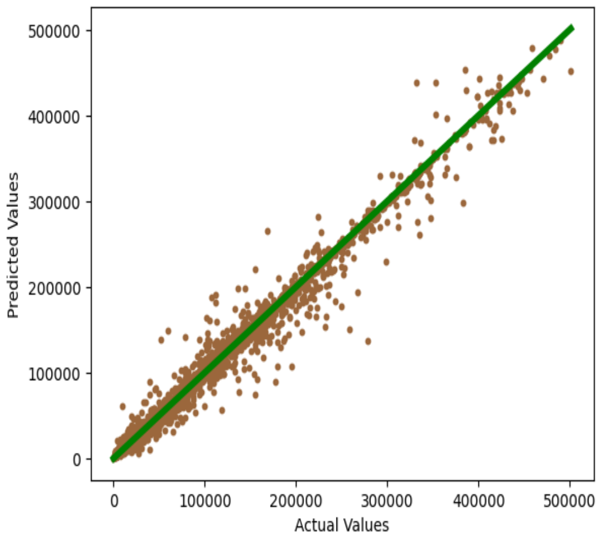


Figure 5: Scatter-Plot for CNN model

V. Conclusion and Future Work

The proposed CNN ensemble for the classification of knee OA is designed with the idea of overcoming the limitations seen in traditional methods of diagnoses, such as variability in interpretation and reliance on subjective assessments. The strength of multiple CNN architectures is strengthened by combining diverse feature extraction capabilities that serve to boost diagnostic accuracy, consistency, and efficiency. The ensemble model may demonstrate great potential for enhancing early detection and ultimately contributing to better treatment plans and, therefore, better health outcomes and quality of life by providing a robust and scalable framework for OA detection.

Several new research directions will be aimed at improving and expanding the utility of the model. Firstly, infusion of additional data sources - in particular, information on patient history, laboratory work, and general data on patients, like age, gender, and other lifestyle preferences - will be made a priority. Such improvements should enrich the model's predictive capabilities, allowing a more comprehensive analysis of patient-specific risk factors and disease progression patterns to be conducted.

Furthermore, work will be aimed at demonstrating and testing the model in real-world clinical environments. This will include stress testing to ensure compliance with healthcare authorities' standards on the use of AI for diagnosis, as well as other regulatory standards. As part of the integration process, adaptation of the model to existent healthcare workflows will be a crucial need in ensuring smooth interoperability with electronic health record systems and other medical infrastructure. Further research will involve refining the ensemble model with more sources for enhancement in predicting capabilities, such as clinical parameters and patient demographics. In addition, the model will be implemented in various clinical settings in an attempt to be implemented towards real-world practice. It will then meet the assessment of applicable regulatory standards and be integrated into the prevailing healthcare workflow. Longitudinal studies will also be followed in further investigating the model's performance and impact on the updating or refreshing patient management strategies in knee osteoarthritis.

References

1. *Rashid, M., Bari, B. S., Yusup, Y., Kamaruddin, M.A., & Khan, N. (2021). A comprehensive review of kneeostheo arthritis prediction using machine learning approaches with special emphasis on palm oil ostheo arthritis prediction. IEEE Access, 9, 63406-63439.*
2. *Keerthana, M., Meghana, K. J. M., Pravallika, S., &Kavitha, M. (2021, February). An ensemble algorithm for kneeostheo arthritis prediction. In 2021 Third International Conference on Intelligent Communication Technologies andVirtual Mobile Networks (ICICV) (pp. 963-970). IEEE.*
3. *P. Kumar, S. Senthil Pandi, T. Kumaragurubaran and V. Rahul Chiranjeevi, "Human Activity Recognitions in Handheld Devices Using Random Forest Algorithm," 2024 International Conference on Automation and Computation (AUTOCOM), Dehradun, India, 2024, pp. 159-163, doi: 10.1109/AUTOCOM60220.2024.10486087.*
4. *Prasad, N. R., Patel, N. R., & Danodia, A. (2021). Kneeostheo arthritis prediction in cotton for regional level using random forest approach. Spatial Information Research, 29, 195-206.*
5. *Shidnal, S., Latte, M. V., & Kapoor, A. (2021). Kneeostheo arthritis prediction: two-tiered machine learning model approach. International Journal of Information Technology, 13, 1983-1991.*
6. *Raja, S. P., Sawicka, B., Stamenkovic, Z., & Mariammal, G. (2022). Kneeprediction based on characteristics of the agricultural environment using various feature selection techniques and classifiers. IEEE Access, 10,23625-23641.*
7. *Cedric, L. S., Adoni, W. Y. H., Aworka, R., Zoueu, J. T., Mutombo, F. K., Krichen, M., & Kimpolo, C. L. M. (2022). Crops ostheo arthritis prediction based on machine learning models: Case of West African countries. Smart Agricultural Technology, 2, 100049.*
8. *A. K, S. Y. Prathima, S. Ps, V. V and T. M, "Toxicity Detection in Soap using Deep Learning," 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICSES60034.2023.10465402.*
9. *Paudel, D., Boogaard, H., de Wit, A., Janssen, S., Osinga, S., Pylianidis, C., & Athanasiadis, I. N. (2021).Machine learning for large-scale kneeostheo arthritis forecasting. Agricultural Systems, 187, 103016.*
10. *Shidnal, S., Latte, M. V., & Kapoor, A. (2021). Kneeostheo arthritis prediction: two-tiered machine learning model approach. International Journal of Information Technology, 13, 1983-1991.*
11. *Paudel, D., de Wit, A., Boogaard, H., Marcos, D., Osinga, S., & Athanasiadis, I. N. (2023). Interpretability of deep learning models for kneeostheo arthritis forecasting. Computers and Electronics in Medical, 206, 107663.*
12. *Gnanavel, S., Duraimurugan, N., Jaeyalakshmi, M. (2024). Smart Surveillance System and Prediction of Abnormal Activity in ATM Using Deep Learning. In: Namasudra, S., Trivedi, M.C., Crespo, R.G., Lorenz, P. (eds) Data Science and Network Engineering. ICDSNE 2023. Lecture Notes in Networks and Systems, vol 791. Springer, Singapore. https://doi.org/10.1007/978-981-99-6755-1_11.*
13. *Zhang, Q., Zhao, X., Han, Y., Yang, F., Pan, S., Liu,Z., ... & Zhao, C. (2023). Maize ostheo arthritis prediction using federated random forest. Computers and Electronics in Medical, 210, 107930.*
14. *Sharma, P., Dadheech, P., Aneja, N., & Aneja, S. (2023). Predicting Medical Ostheo arthritiss Based on Machine Learning Using Regression and Deep Learning. IEEE Access.*
15. *Shafi, U., Mumtaz, R., Anwar, Z., Ajmal, M. M., Khan, M. A., Mahmood, Z., ... & Jhanzab, H. M. (2023). Tackling food insecurity using remote sensing and machine learning based kneeostheo arthritis prediction. IEEE Access.*
16. *E Ringdahl, S Pandit - American family physician, 2011 - aafp.org*
17. *K Rönn, N Reischl, E Gautier, M Jacobi - Arthritis, 2011 - Wiley Online Library*
18. *JC Mora, R Przkora, Y Cruz-Almeida - Journal of pain research, 2018 - Taylor & Francis*
19. *Adler-Milstein J, DesRoches CM, Jha AK. Health information exchange among US hospitals. American Journal of Managed Care. 2011;17(11):761–768. [PubMed]*
20. *Anhang Price R, Elliott MN, Zaslavsky AM, Hays RD, Lehrman WG, Rybowski L, Edgman-Levitan S, Cleary PD. Examining the role of patient experience surveys in measuring health care quality. Medical Care Research and Review. 2014;71(5):522–554. [PMC free article] [PubMed]*
21. *Agwunobi J, London PA. Removing costs from the health care supply chain: Lessons from mass retail. Health Affairs. 2009;28(5):1336–1342. [PubMed]*
22. *.Bell RM, Koren Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights; Paper presented at Seventh IEEE International Conference on Data Mining; Omaha, NE. 2007*
23. *Bleustein C, Rothschild DB, Valen A, Valatis E, Schweitzer L, Jones R. Wait times, patient satisfaction scores, and the perception of care. American Journal of Managed Care. 2014;20(5):393–400.*
24. *Gabow PA, Goodman PL. The Lean prescription: Powerful medicine for our ailing healthcare system. New York: Productivity Press; 2014.*
25. *Hall R. Patient flow: Reducing delay in healthcare delivery. New York: Springer Science & Business Media; 2013.*
26. *HHS (Department of Health and Human Services). HITECH Act enforcement interim final rule. 2015. [May 1, 2015]. http://www.hhs.gov/ocr/privacy/hipaa/administrative/enforcementrule/hitechenforcementifr.html.*
27. *Howley MJ, Chou EY, Hansen N, Dalrymple PW. The long-term financial impact of electronic health record implementation. Journal of the American Medical Informatics Association. 2015;22(2):443–452. [PubMed]*

APPENDIX-I

PUBLICATION STATUS

PUBLICATION STATUS OF PHASE I PAPER

TITLE: **Prediction of the 8 stages of osetoarthiritis using deep learning**

AUTHOR: **Dr. T. Kumaragurubaran. M.E., Ph.D.,**
Dr. K. Ananthajothi B.C.A M.E M.C.A Ph.D
Dr. P. Kumar, M.E., Ph.D.,
Madhumita P
Kavya Shree BN

CONFERENCE: **International Conference On Emerging Research In Computational Science -2024**

MODE OF PUBLICATION: **Online**

STATUS: **Yet to Publish**

Paper Acceptance E-mail:

ORIGINALITY REPORT

14%	6%	12%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Shubhadip Ray, Norbert Herman, Indrajit Sen. "Disruptive Transformation Fueling Gig Economies", 2021 IEEE Technology & Engineering Management Conference - Europe (TEMSCON-EUR), 2021 Publication	8%
2	link.springer.com Internet Source	1%
3	research.chalmers.se Internet Source	1%
4	Submitted to University of Illinois at Urbana-Champaign Student Paper	1%
5	Muhammad Zohaib Siddique, Muhammad Basit, Iqra Fatima. "Empirical Study for Improving Project Allocation on Freelancing Platform", 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2022 Publication	1%

FINAL_REPORT.pdf

ORIGINALITY REPORT

14%	10%	4%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Illinois at Urbana-Champaign Student Paper	6%
2	forum.djangoproject.com Internet Source	1%
3	Submitted to University of Houston Clear Lake Student Paper	1%
4	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	<1%
5	www.geeksforgeeks.org Internet Source	<1%
6	cas.upm.edu.ph:8080 Internet Source	<1%
7	Submitted to Queen Mary and Westfield College Student Paper	<1%