DESIGN AND DEVELOPMENT OF A NOVEL MEDICAL EXPERT SYSTEM FOR LUMBAR DISC DISORDER

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

By Ruchi

Registration Number: 11914889

Supervised By

Dr. Dalwinder Singh (11265)

School of Computer Science and Engineering (Associate Professor)

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

Declaration of Authorship

I, hereby declared that the presented work in the thesis entitled "Design and Development of a Novel Medical Expert System for Lumbar disc disorder" in fulfillment of the degree of **Doctor of Philosophy (Ph.D.)** is the outcome of research work carried out by me under the supervision Dr. Dalwinder Singh, working as Associate Professor, in the Department of Computer Science and Engineering of Lovely Professional University, Punjab, India. In keeping with the general prac- tice of reporting scientific observations, due acknowledgments have been made whenever the work described here has been based on the findings of another inves- tigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

Ruchi

(Signature of Scholar) Name

of Scholar: Ruchi

Registration No: 119 14 8 89

Department: Department of Computer Science and Engineering

University: Lovely Professional University, Phagwara, Punjab, India

Date: March 2025

Certificate

This is to certify that the work reported in the Ph.D. thesis entitled "Design and Development of a Novel Medical Expert System for Lumbar Disc Disorder" submitted in fulfillment of the requirement for the reward of the degree of **Doctor of Philosophy (Ph.D.)** in the Department of Computer Science and Engineering, is a research work carried out by Ms. Ruchi, Registration No. 11914889, is a bonafide record of his/her original work carried out under my supervision and that no part of the thesis has been submitted for any other degree, diploma or equivalent course.

(Signature of Supervisor)

Name of Supervisor: Dr. Dalwinder Singh

Designation: Associate Professor

Department: Department of Computer Science and Engineering

University: Lovely Professional University, Phagwara, Punjab, India

Date: March 2025

Abstract

The lumbar spine can impact a person's life adversely. The painful nature of the spine can make the sufferer's life harder. Back pain is an issue affecting millions of people throughout the world. Research on back pain root cause detection is immense. The lumbar spine is a lower back region of the backbone that could also be responsible for back pain. Research on issues related to lumbar spine disease (LSD) is limited. The lumbar spine is critical for the human body as it supports the weight of the body. Many diseases can affect the lumbar spine adversely.

To overcome the issue, a technology-driven approach is required. This research is directed towards the detection of LSDs using optimized feature extraction and selection phases. Furthermore, the linearity-based model is used for feature selection, selecting only the best possible features to reduce the missed classification degree. The flow of the proposed research work is divided into phases. In the first phase, data collection is performed. Data collection in the proposed work includes both real-time and benchmark magnetic resonance imaging (MRI) datasets. The MRI dataset collected from the hospital is validated by medical experts. For the purposes of making a diagnosis of a condition affecting the lower spine, the collection of data is an obligatory requirement. It's possible that the data that was obtained was in the form of text, but it may also have been an image. The gathering of data in the work that is being suggested is, to a primary extent, dependent upon picture datasets. MRI image dataset compiled by travelling to a number of different labs in Punjab (Pathankot) and making use of the internet as a method of gathering. [101-103] The utilization of correlation analysis is what enables the validation of datasets to become a reality. The data that has been obtained has inconsistencies that need to be fixed in order to be useful After data collection, pre-processing is applied. This step removes the noise from the collected dataset. The pre-processing mechanism is performed using histogram equalization, median filtering, validation, and normalization.

After the pre-processing phase, background subtraction and region of interest (ROI) detection are performed using the region-cut mechanism. Optimal feature extraction is achieved using a differential spider monkey optimization (SMO), and feature selection is performed using a linearity-based convolutional neural network (CNN) model. In the end, ensemble-based classification is used for disease prediction. The validation of the result is conducted through classification accuracy, specificity, sensitivity, and F- Score. The high classification accuracy of 96% is achieved with multi support vector machine (MSVM), 94% with random forest (RF), 93.5% with a decision tree (DT), and91% with the Naïve Bayes (NB) approach, proving the validity of the proposed approach.

Our innovative system represents a breakthrough in lumbar spine detection, leveraging cutting-edge technologies to enhance accuracy and efficiency in diagnosing lumbar spine conditions. By combining advanced medical imaging techniques with artificial intelligence (AI), our system offers a novel approach to detecting abnormalities and anomalies within the lumbar spine region.

Our system seamlessly integrates with existing medical imaging technologies, streamlining the diagnostic process for healthcare professionals. Radiologists and clinicians can benefit from a user-friendly interface that provides detailed insights into lumbar spine conditions, aiding in prompt and informed decision-making. The system's ability to rapidly process large volumes of medical imaging data accelerates the diagnostic workflow, enabling healthcare providers to offer timely and effective interventions.

Moreover, our system prioritizes patient privacy and data security, adhering to the highest standards of compliance with healthcare regulations. It ensures the confidentiality of sensitive medical information, instilling confidence in both healthcare professionals and patients.

Acknowledgements

First and foremost, I want to give praise and gratitude to God, the Almighty, for His gifts that helped me finish my research successfully.

I want to express my sincere appreciation to Dr. Dalwinder Singh for all of his assistance with this thesis and study, including his guidance, advice, patience, and supervision. Despite his busy schedule, he has provided time, effort, and much-needed guidance and assistance at every stage. This made it possible for me to go through the challenging periods and finish this tremendous undertaking.

I would like to thank the research project committee members for their insightful remarks and conversations. A special thanks to the management of Lovely Professional University for their support in academic concerns and for letting me involve in my research study. The doctoral program of LPU has made it possible for me to pursue my dream of research and upgrade my knowledge.

I am deeply grateful to my hubby Er. Shubham Raina for offering full supportto me during the entire period of my research work. My heartfelt gratitude and indebtedness go to my loving parents and my brother Mr. Hitesh for their blessings, encouragement, and assistance.

Moreover, I would also like to thank my colleague Simran Kaur whom I met during my Ph.D. for all the moral support and for helping me at every stage of the research project.

Finally, I would like to thank each and every person who has, directly and indi- rectly, helped and motivated me in this journey.

Date: March 2025

Contents

List of Figures

List of Tables

Abbreviations

KNN	K Nearest Neighbour
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ReLu	Rectified Linear Unit
LBP	Lower Back Pain
GUI	Graphical User Interface
AI	Artificial Intelligence
ML	Machine Learning
T	Throughput
SMO	Spider Monkey Optimization
IVD	Intervertebral Disc
ML	Deep Learning
DL	Machine Learning
LSD	Lumbar Spine Disease

Dedicated to my beloved family. . .

Chapter 1

Introduction

1.1 Introduction

This work consists of identifying the Lumbar Spine diseases using benchmarked dataset. The benchmarked dataset will be derived from Kaggle. The deep learning-based mechanism using convolution neural network is applied for categorizing the data into healthy and persons having Lumbar Spine. The automated approach will help in early diagnosis of the disease and take necessary action for avoiding severe consequences [78]. This section provides the introduction about the Lumbar Spine disease, along with preventive measures

The lumbar spine is at the lower end of the spinal cord and is required for balancing the weight of the human body, as discussed in [1]. The spinal cord area is shielded by five durable and flexible vertebrae that ensure the dispersion of axial forces, as discussed in [2]. The spinal cord goes through the vertebrae and terminates at the level of the L1 and L2 vertebrae. The cauda equina begins at the termination of the spinal cord and descends through the remainder of the canal, as discussed in [81]. The lumbar spine is made up of bones, cartilage, nerves, and muscles. Each of these components plays a critical role in the formation and operation of the lumbar spine [82]. The vital functions of the lumbar spine are shown in figure 1.1

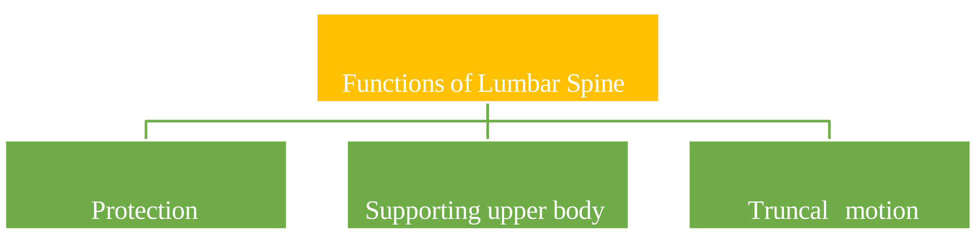


Figure 1.1 The vital Functions of Lumbar Spine

There are three different vital functions associated with the lumbar spine. The primary one is the protection of the spinal cord and nerves [83]. Besides, it bears the upper body weight and provides support to the neck, head, and trunk [59]. The next one is truncal motion, including extension, side bending, flexion, and rotation [7]. Any issue associated with the lumbar spine can cause painful disorders including spinal stenosis, herniated disk, cervical spondylitis, kyphosis, and many more

Lumbar Spine disease is common among individuals having issues with Lumbar Spine gland. [25] discussed clinical applications in the detection of Lumbar Spine related diseases. As per the research [36], thyrotropin can be used for the prevention of excessive release of hormones. [9] discussed the applications of clinical tests for the detection of ankylosing spondylitis Lumbar Spine. [79] discussed the diseases associated with Spinal stenosis. Ankylosing spondylitis and Spinal stenosis related diseases are equivalently dangerous. The ankylosing spondylitis Lumbar Spine causes hormones to decay, and Spinal stenosis causes hormones to increase excessively [95].

In general, Lumbar Spine related diseases can be generally divided into two different categories including ankylosing spondylitis Lumbar Spine and Spinal Stenosis Lumbar Spine. Lumbar spinal stenosis (LSS) is a condition marked by the constriction of the spinal canal within the lower back's lumbar region. This constriction applies pressure on both the spinal cord and the nerves extending from the lower back into the legs, resulting in a range of symptoms and Ankylosing spondylitis is a type of inflammatory arthritis that primarily affects the spine, particularly the sacroiliac joints and the vertebrae of the lower back (lumbar spine). This chronic condition causes inflammation, pain, and stiffness in the affected areas, leading to fusion of the spinal vertebrae over time. Moreover, the lumbar spine bears the weight of the upper body and offers support to the neck, head, and trunk.
[59] Furthermore, we can categorize Lumbar Spine and related disorder on

specific terms as well. The categorization of Lumbar Spine and corresponding disorder caused is given in Table 1.1.

Table 1.1 Lumbar Spine Disease Categorization

Disorder	Characteristics
Ankylosing Spondylitis	Initially no or few symptoms appear In case not detected for longer period of times, it may result in health issues like obesity, joint pain etc.
Spinal stenosis	This disorder has initial symptoms and causes weight loss and abnormal heartbeat.

In ankylosing spondylitis, Lumbar Spine gland does not produce enough hormones and in case of Spinal Stenosis Lumbar Spine, hormones are produced in excess. In both the situations, adverse body conditions may appear

Most common diseases with the Ankylosing Spondylitis Lumbar Spine Are Rheumatoid arthritis, Lupus, and type 1 diabetics. Most common diseases that are caused by Spinal Stenosis Lumbar Spine are graves' disease, overactive Lumbar Spine nodules and Lumbar Spine [31].

Any issue associated with the lumbar spine can cause painful disorders including spinal stenosis, herniated disk, cervical spondylitis, kyphosis, and many more [57]. An automated approach using machine learning (ML) and neural networks (NN) has been researched [9– 13]. Both ankylosing spondylitis as well as Spinal stenosis required to be detected at early stage. To this end, convolution neural network-based mechanism is proposed using dataset derived from Kaggle. The text-based dataset is used for evaluation in this case. The validity of the approach is expressed in the form of metrics including classification accuracy, specificity, and sensitivity.

Figure 1.2 highlight some of the commonly occurring disorder including

Hashimoto, Graves disease, Papillary Carcinoma. These are just a few of the illnesses that result from issues in the lumbar spine. More adverse and life-threatening disease can be caused with Lumbar Spine. Thus, using clinical tests and through technology, implications and adverse effects of Lumbar Spine must be minimized.

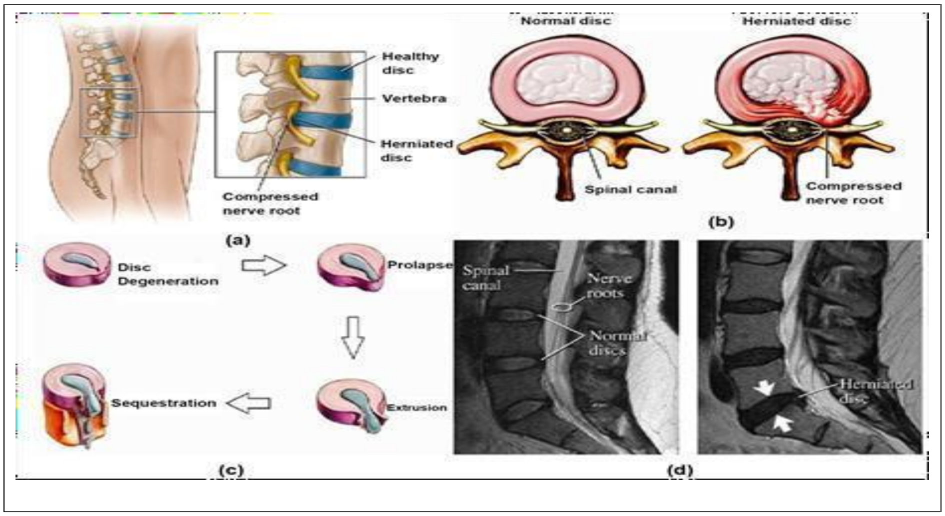


FIGURE 1.2: Lumbar Spine disease categorization

1.2 Clinical Diagnosis

To detect Lumbar Spine and related diseases, doctor may prescribe blood and imaging related tests for checking Lumbar Spine function.

1.2.1 Blood Test

It may include as follows:

1. TSH Test
2. T4
3. T3
4. Lumbar Spine antibody test

Healthcare providers typically begin by obtaining the level of TSH in a person's blood. TSH, a hormone produced in the hypophysis, regulates the production of T4 and T3 in the lumbar spine. Elevated TSH levels typically indicate the presence of Ankylosing Spondylitis in the lumbar spine or an underactive lumbar spine. This indicates that the individual's lumbar spine is not producing sufficient amount of hormone. Consequently, the hypophysis continues to produce and release TSH into the individual's bloodstream. A low TSH level typically suggests the presence of spinal stenosis in the lumbar spine or an overactive lumbar spine. This implies that the individual's lumbar spine is producing an excessive amount of hormone, leading the hypophysis to cease production and release of TSH into the individual's bloodstream. Should the TSH test results deviate from the norm, the individual will require at the minimum one additional test to aid in identifying the underlying origin of the issue.

An elevated T4 concentration in the blood could indicate person has Spinal Stenosis Lumbar Spine. A reduced level of T4 could indicate person has Anki Spondylitis Lumbar Spine. In certain instances, elevated or diminished T4 levels may not necessarily signify person has Lumbar Spine problems. If an individual is pregnant or using oral contraceptives, their lumbar spine hormone levels will be elevated. Serious illnesses or the use of corticosteroids, which are medications for conditions like asthma, arthritis, and skin disorders, can reduce T4 levels. These medications and conditions alter the concentration of proteins in the bloodstream of an individual that bind to T4. Bound T4 remains stored in the bloodstream till it's required. “Free” T4 isn't attached to these proteins and can freely enter body tissues. Since alterations in binding protein levels don't impact the availability of free T4, numerous healthcare providers opt to assess free T4 levels. If an individual health care expert believes he or she might be having Spinal Stenosis Lumbar Spine despite the individual's T4 level being within the normal range, person may have a T3 test to confirm the diagnosis. On occasion, T4 levels may appear normal while T3 levels are elevated, hence evaluating both T4 and T3 levels can be useful in detecting lumbar spinal stenosis Testing Lumbar Spine antibody levels can aid in diagnosing an autoimmune Lumbar Spine disorder such as Graves’ disease—the most common cause of Spinal stenosis—and Hashimoto’s disease—the primary reason for ankylosing spondylitis Lumbar Spine. Lumbar Spine antibodies develop when an individual's immune system mistakenly targets the Lumbar Spine gland. Person health care expert may request Lumbar Spine antibody tests if the results of other blood tests suggest Lumbar Spine disease. In addition to the blood test, imaging tests are also prescribed by doctors for diagnosing the Lumbar Spine diseases.

A healthcare provider may request multiple imaging examinations to diagnose and determine the root cause of lumbar spine disease. Several diagnostic procedures are commonly employed to evaluate the condition of the lumbar spine, depending on the particular symptoms and suspected ailments. These examinations encompass: X-rays, MRI (Magnetic Resonance Imaging), CT (Computing Tomography) Scan, Bone Scan etc.

Typically, a skilled technologist performs these examinations conducted at the physician's clinic, outpatient facility, or hospital. Subsequently, a radiologist, a specialist in medical imaging, analyzes the images and generates a document for the healthcare provider to discuss with the individual.

Lumbar Spine blood test is given with the following hierarchical diagram

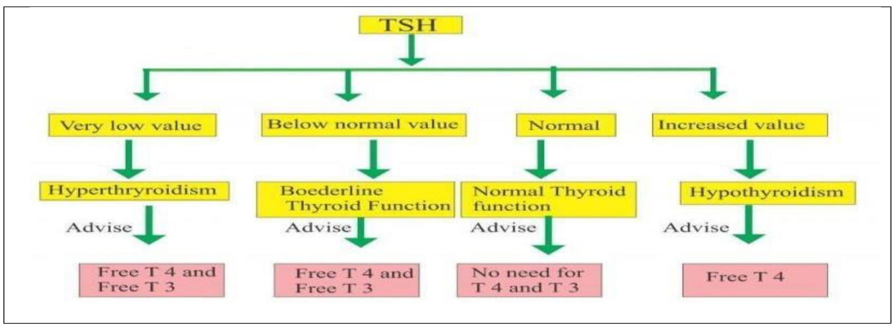


FIGURE 1.3: Blood test for Lumbar Spine

1.2.2 Ultrasound

Ultrasound imaging of the lumbar spine is primarily utilized to examine or scrutinize nodules within this region. These nodules, which are lumps found in the individual's neck, can be assessed more effectively through ultrasound. This imaging technique aids the physician in determining the likelihood of these nodules

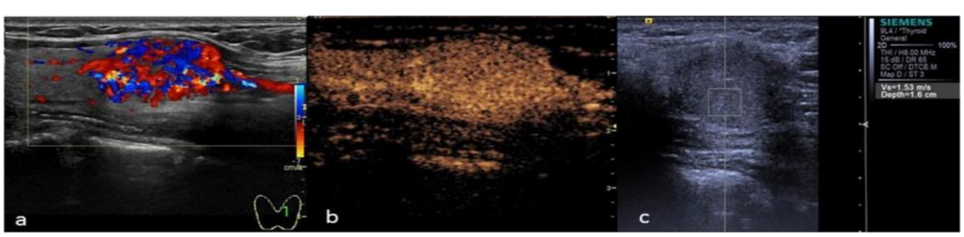


FIGURE 1.4: Ultrasound images

Being cancerous. During an ultrasound procedure, the individual will recline on an examination table while an expert passes an instrument known as a transducer over their neck. This transducer emits harmless sound waves that bounce off the neck, generating images of the lumbar spine. Typically, the ultrasound process lasts approximately 30 minutes.

1.2.3 Lumbar Spine Scan

Medical professionals utilize a lumbar spine scan to examine the dimensions, configuration, and positioning of the lumbar spine gland. This diagnostic procedure involves the administration of a minimal quantity of radioactive iodine to assist in identifying the underlying reason of spinal stenosis and to assess for lumbar spine nodules. Your healthcare provider might recommend refraining from consuming iodine-rich foods like kelp or medications that are having iodine for one-week prior undergoing the test.

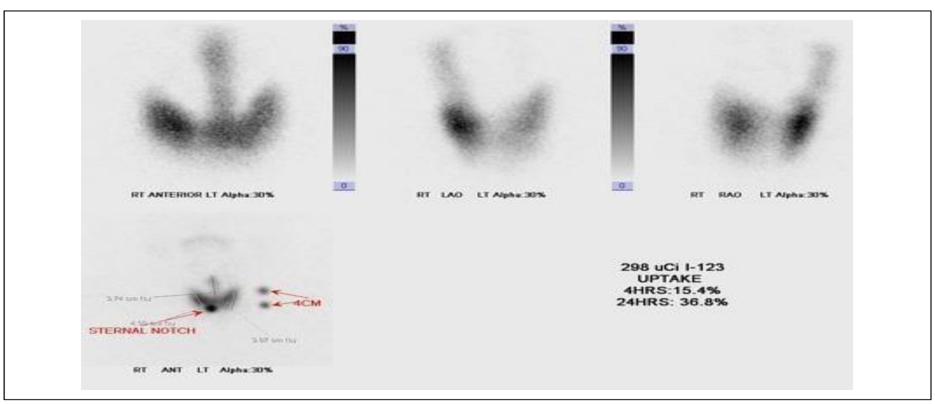


FIGURE 1.5: Lumbar Spine Scan

During the scan, an expert administers a minor dose of radioactive iodine or a comparable substance intravenously. Alternatively, one can also ingest the substance either as a liquid or in capsule form. The scan occurs either 30 minutes following injection or up to 24 hours after ingestion, allowing adequate time for your lumbar spine to assimilate it.

Throughout the scan, you'll be positioned on an examination table as a specialized camera captures images of your lumbar spine. Typically, this procedure lasts for 30 minutes or less. Lumbar spine nodules exhibiting excessive lumbar spine hormone secretion are distinctly visible in the images. If radioactive iodine appears throughout the entire lumbar spine, it could indicate the presence of Graves' disease.

Despite requiring just a minimal dose of radiation, a lumbar spine scan is generally considered safe. However, it is not advisable to undergo this test if you are expecting a baby or breastfeeding.

1.2.4 Radioactive Iodine Uptake Test

A radioactive iodine uptake test, alternatively known as a lumbar spine uptake test, aids in evaluating lumbar spine function and deciding the underlying origin of lumbar spine disorders. The term "uptake" refers to the lumbar spine's absorption of iodine from the bloodstream to produce lumbar spine hormones, hence the name of the test. Prior to the test, your healthcare provider may advise avoiding iodine-rich foods for example kelp or medications consume iodine for a week.

During this examination, you'll ingest a minor dose of radioactive iodine either as a liquid form or as capsules. Throughout the procedure, you'll be positioned in a chair while a technician positions a device known as a gamma probe in proximity to your neck, near the lumbar spine gland. This probe gauges the lumbar spine's uptake of radioactive iodine from your bloodstream. Typically, measurements are obtained 4 to 6 hours' post-ingestion of the radioactive iodine, with additional readings taken at the 24-hour mark. The entire test usually lasts just a couple of minutes.

If your lumbar spine accumulates a significant quantity of radioactive iodine, it could indicate Graves' disease or the presence of one nodule or multiple nodules overproducing lumbar spine hormone. This test may be conducted concurrently with a lumbar spine scan.

1.3 Technology in the field of Lumbar Spine Disease Detection

Use of technology can influence early detection of the lower back pain using artificial intelligence, machine learning and deep learning-based mechanisms. Different mechanisms that can play a critical role in the detection of Lumbar Spine related disorder are described as under:

1.3.1 AI and Machine Learning based approach

The artificial intelligence plays a critical role in the detection and prediction of disease at early stage. Users with AI and machine learning applications installed can directly input the symptoms and predictions will be generated. The application of such sort includes "Food and Symptom tracker", "Meditation Timer" and many more. The overall process followed within AI and Machine learning based mechanism includes pre-processing, segmentation, and classification. The AI and machine learning based mechanism operates on the real time data and match the results with the pre-built trained model. The issue however is the execution time when symptoms other than trained model appears. thus, machine learning based models may not be suitable for large datasets. Using Pre- processing, noise from within dataset is handled. For this purpose, number of filters are used. these filters include median filtering, Gaussian filtering, Most probable value substitution etc. After pre-processing, data is segmented into critical and non-critical parts. Critical parts are retained for comparison against the pre-trained model. This phase is crucial as degree of misclassification is reduced by the use of segmentation. At the end, classification phase is performed. Classification is the mean of identifying the disease from the symptoms entered by the user. There are number of classifiers that can be used for the prediction purpose. Some of the classifiers includes kNN, Random Forest, decision tree, Naïve Bayes and many more.

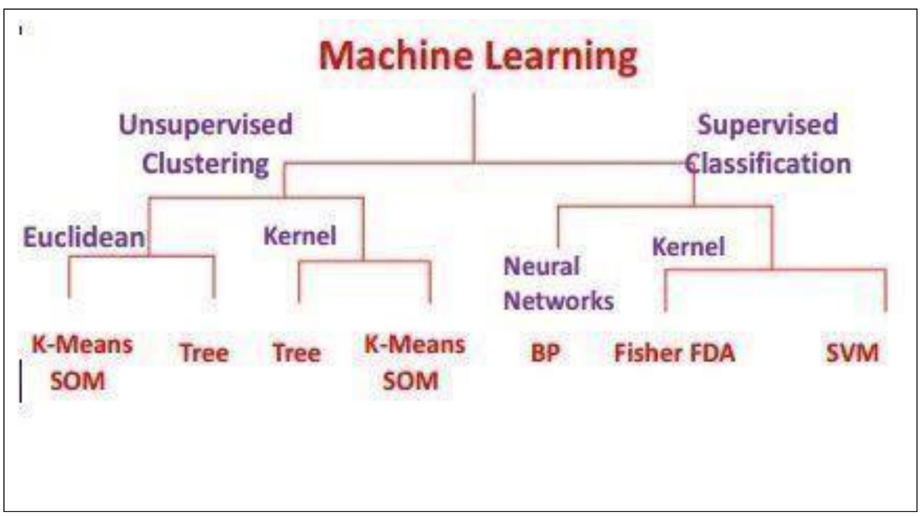


FIGURE 1.6: Machine learning with Types

Artificial intelligence and machine learning commonly used two different types of learning mechanisms for training. These mechanisms include

- Supervised learning
- Unsupervised learning

Supervised learning indicates that model is required to be trained every time prediction is to be generated. As the training is required to be done at every prediction, thus, it may not be suitable for prediction corresponding to multiple classes.

Unsupervised learning on the other hand is trained once and model acquire additional knowledge in case features other than hold by model appears. Thus, this model is suitable for large dataset. The proposed model perform training each time a new prediction is to be generated thus, supervised learning is used.

1.3.2 Deep Learning Models

The deep learning-based model can be used for handling large dataset. The overall execution speed improves using deep learning-based mechanism. The process flow of the deep learning-based framework is given within figure 1.7

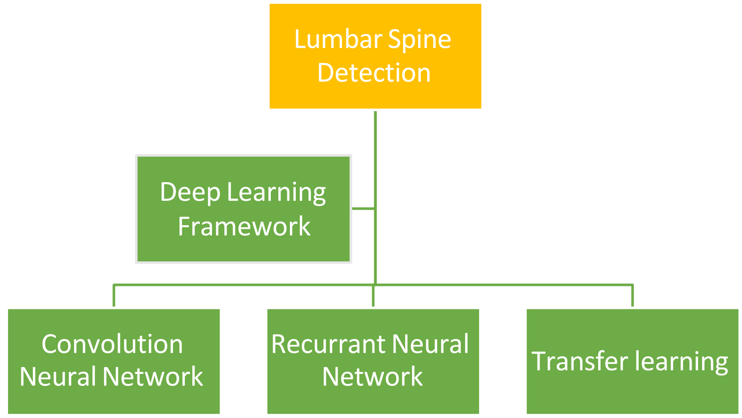
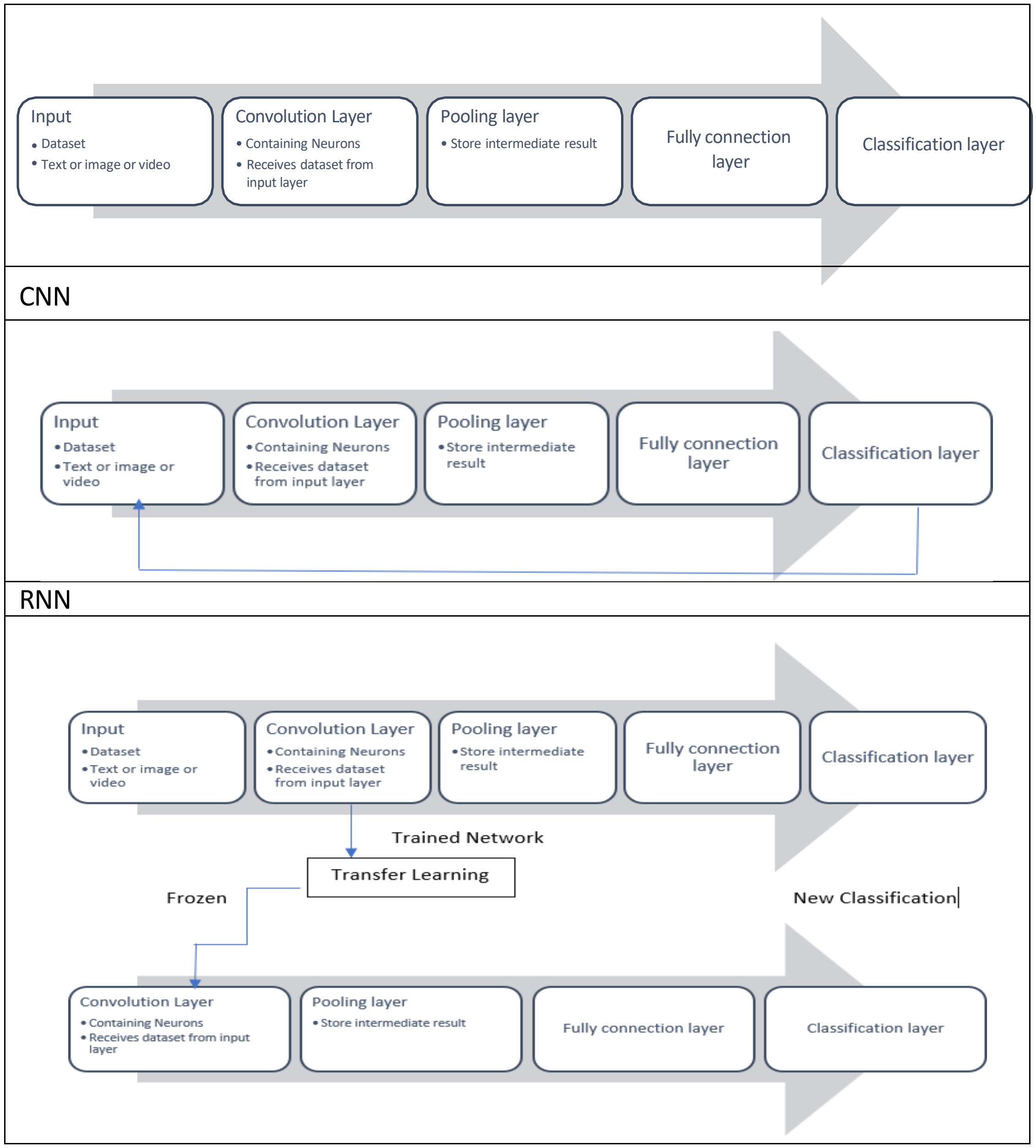


FIGURE 1.7: Process flow of deep leaning for Lumbar Spine detection

The CNN for correct identification of Lumbar Spine disorder as predicted in [35]. Input layer in this case was used to receive the text data set. Convolution layer was used in this case to receive the input and perform filtering. The fully connected layer in this case a received the preprocessed elements from the convolution layer and apply the processing strategies. The outcome was produced on the output layer. As the overall process is divided hence the complexity has been reduced. Thus, the model based on Convolutional Neural Networks (CNNs) is capable of handling larger datasets. [54] [87] suggested deep learning-based model for the identification of Lumbar Spine disorder. The layered based approach was applied on the benchmarked dataset. The implication of the approach detects 12% of the persons suffering from Lumbar Spine with the deep learning-based approach. [43] explored the utilization of convolutional neural networks for detecting Lumbar Spine disorders. The Lumbar Spine dataset was used for the demonstration of the Lumbar Spine detection. [41] discussed the application of transfer learning in the form of discriminant analysis to detect Lumbar Spine disorder within person. The dataset pertaining to lumbar spine disorders was obtained from Kaggle and utilized to examine such disorders within a dataset of children.

A k-nearest neighbors (kNN) based classification approach was employed, achieving a high classification accuracy of approximately 90% [45]. Additionally, [45] proposed a deep learning-based mechanism grounded in detecting diseases related to the lumbar spine in children. Moreover, disabilities of each individual were identified using a CNN-based mechanism. The CNN achieves high classification accuracy and F-score. [26] introduced a recurrent neural network (RNN) for detecting lumbar spine disorders. A network based on RNN enables layers to establishing links with cycle or analogous characteristics, which are typically restricted in convolutional neural networks. This RNN-based model aids in enhancing classification accuracy, especially with complex datasets.

The structural difference between CNN, RNN and Transfer learning is given within figure 1.8



Transfer Learning

Figure 1.8: Structure of CNN, RNN and Transfer learning

Medical imaging has a pivotal role in the assessment and treatment of various musculoskeletal disorders, with lumbar spine detection being a critical component of this diagnostic process. The lumbar spine, comprising the lower back's five vertebrae (L1-L5), is prone to a myriad of pathologies, making its accurate detection imperative for early intervention and effective management. This essay delves into the rationale behind lumbar spine detection in medical imaging, exploring the significance, challenges, and technological advancements driving this essential diagnostic process.

Significance of Lumbar Spine Detection:

1. **Prevalence of Lumbar Spine Disorders:** Lumbar spine disorders, such as herniated discs, degenerative disc disease, and spinal stenosis, are widespread and can significantly impact an individual's quality of life. Early detection allows for timely intervention, preventing further deterioration and improving patient outcomes.
2. **Diagnostic Precision:** The lumbar spine's complex anatomy necessitates precise detection methods to identify abnormalities accurately. Medical imaging techniques, including X-rays, CT scans, and MRI, provide detailed images essential for diagnosing conditions affecting the lumbar spine.
3. **Informed Decision-Making:** Accurate lumbar spine detection empowers healthcare professionals to make informed decisions regarding treatment plans. Whether recommending conservative measures, physical therapy, or surgical intervention, the detection of lumbar spine abnormalities guides clinicians in providing tailored and effective care.

Challenges in Lumbar Spine Detection:

1. **Anatomic Complexity:** The intricate structure of the lumbar spine, with its intervertebral discs, facet joints, and nerve roots, poses

challenges in differentiating between normal variations and pathological conditions. Advanced imaging technologies are essential to overcome this complexity.

2. **Variability in Patient Anatomy:** Individuals exhibit variations in lumbar spine anatomy, and pathology may manifest differently in each case. This variability necessitates adaptive detection algorithms capable of accommodating diverse anatomical presentations.
3. **Integration of Multimodal Imaging:** To enhance diagnostic accuracy, clinicians often rely on a combination of imaging modalities. Integrating information from X-rays, CT scans, and MRI requires sophisticated algorithms and seamless interoperability between different imaging technologies.

Technological Advancements Driving Lumbar Spine Detection:

1. **Artificial Intelligence (AI) and Machine Learning(ML):** AI and machine learning algorithms are transforming lumbar spine detection by automatizing the analysis of medical images. These technologies can rapidly process vast amounts of data, assisting radiologists in identifying subtle abnormalities and streamlining the diagnostic workflow.
2. **Three-Dimensional (3D) Imaging:** Traditional two-dimensional imaging has limitations in representing the spatial complexity of the lumbar spine. The advent of 3D imaging techniques provides clinicians with a more comprehensive view, enabling better visualization of anatomical structures and abnormalities.
3. **Quantitative Imaging Biomarkers:** Quantitative imaging biomarkers, such as measurements of disc height, angles, and signal intensity, contribute to a more objective assessment of lumbar spine health. These biomarkers aid in the early detection and monitoring of degenerative

changes.

In conclusion, the rationale for lumbar spine detection in medical imaging is grounded in the prevalence of disorders affecting this region, the need for diagnostic precision, and the potential for informed decision-making in patient care. Despite challenges arising from anatomical complexity and patient variability, ongoing technological advancements, particularly in AI, 3D imaging, and quantitative biomarkers, are transforming lumbar spine detection. As medical imaging continues to evolve, the accurate and early detection of lumbar spine abnormalities remains paramount for improving patient outcomes and advancing musculoskeletal healthcare.

Why Machine learning for Lumbar Spine?

The field of medical imaging has witnessed a paradigm shift with the integration of machine learning (ML) techniques, particularly in the realm of lumbar spine prediction. As a critical component of the musculoskeletal system, the lumbar spine is susceptible to various disorders, necessitating accurate and timely prediction methods. This essay explores the multifaceted role of machine learning in lumbar spine prediction, examining the significance, challenges, and future prospects of leveraging these advanced technologies for enhanced diagnostics and treatment planning.

Significance of Lumbar Spine Prediction:

Early Detection and Intervention:

Machine learning algorithms offer the potential for early detection of lumbar spine abnormalities, enabling healthcare professionals to intervene before conditions progress to advanced stages. Early intervention can mitigate the severity of disorders, improving patient outcomes and reducing the economic burden associated with prolonged treatments.

Personalized Treatment Plans:

Lumbar spine prediction through machine learning enables the development of personalized treatment plans. By analyzing patient-specific data, ML algorithms can identify patterns and correlations that inform tailored interventions, considering factors such as anatomy, lifestyle, and genetic predispositions.

Efficient Resource Utilization:

Predictive models can optimize the allocation of healthcare resources by identifying high- risk individuals who may require more frequent monitoring or early intervention. This targeted approach enhances the efficiency of healthcare delivery, ensuring that resources are directed where they are most needed.

Challenges in Lumbar Spine Prediction using ML-

Data Quality and Quantity:

The success of machine learning models heavily depends on the accessibility of high-quality, labeled datasets. In the context of lumbar spine prediction, obtaining diverse and comprehensive datasets that capture various pathologies and patient profiles can be challenging.

Inter-Model Variability:

Inter-model variability may lead to differences in predictions, necessitating efforts to establish consensus and reliability in the field.

Clinical Interpretability:

Machine learning models often operate as "black boxes," creating difficulties for health care professionals to interpret their decisions. Understanding the rationale behind predictions is crucial for gaining trust and widespread acceptance of these technologies in clinical settings.

Ethical Considerations:

The use of machine learning in lumbar spine prediction raises ethical concerns related to patient privacy, data security, and the potential for bias in algorithms. Addressing these prioritizing ethical considerations is crucial for ensuring the responsible and equitable deployment of predictive models in healthcare.

Role of Machine Learning in Lumbar Spine Prediction:

Automated Image Analysis:

Machine learning algorithms excel in automated image analysis, offering the ability to process vast amounts of medical imaging data quickly and accurately. In lumbar spine prediction, these algorithms can identify subtle patterns indicative of abnormalities, aiding radiologists in their diagnostic assessments.

Feature Extraction and Selection:

ML techniques enable the extraction and selection of relevant features from complex datasets. In lumbar spine prediction, these features may include measurements of disc height, angles, and signal intensities, providing quantitative insights that contribute to more accurate predictions.

Integration of Multimodal Data:

Lumbar spine prediction benefits from the integration of multimodal data, combining information from X-rays, CT scans, MRI, and patient history. Machine learning facilitates the seamless integration of diverse datasets, enhancing the comprehensiveness and accuracy of predictive models.

Clinical Decision Support Systems:

ML-based clinical decision support systems (CDSS) aid healthcare professionals by providing real-time predictions and recommendations based on patient data. In lumbar spine prediction, CDSS can assist radiologists in interpreting imaging results, leading to more confident and accurate diagnoses.

Future Prospects and Considerations:

Advancements in Explainable AI:

Efforts to enhance the explain ability of machine learning models are underway, with researchers focusing on developing explainable AI techniques.

Collaboration between Clinicians and Data Scientists:

The successful integration of machine learning in lumbar spine prediction requires collaborative efforts between clinicians and data scientists. Clinicians' domain expertise is essential for refining algorithms and ensuring that predictive models align with clinical realities.

Continuous Model Validation and Updating:

Machine learning models should undergo continuous validation and updating to account for changes in patient demographics, treatment protocols, and emerging pathologies. Regular model retraining ensures that predictive accuracy is maintained over time.

Ethical Guidelines and Standards:

The development and deployment of machine learning model algorithms for lumbar spine prediction should adhere to ethical guidelines and standards. This includes addressing issues related to patient consent, data privacy, and algorithmic fairness to ensure responsible and equitable use in healthcare settings.

Dataset Used in Previous Research

Certainly! Below, I’ve compiled a table 1.2 comparing two lumbar spine datasets along with their key features:

TABLE 1.2: Comparative analysis of Dataset used in previous research

Dataset Name	Description	Number of Patients	Number of MRI Slices	Resolution (Pixels)	Slice Thickness (mm)	Position
Lumbar Spine MRI Dataset	Anonymized clinical MRI studies of 515 patients with symptomatic back pain. Each patient data can have one or more MRI studies associated with it.	515	48,345	320x320 (majority)	4 mm	Head-First-Supine / Feet-First-Supine
CTSpine1K Dataset	Large-scale dataset for spinal vertebrae analysis tasks. Useful for vertebrae segmentation, labeling, and 3D spine reconstruction from biplanar radiographs.	N/A	N/A	N/A	N/A	N/A

- **Lumbar Spine MRI Dataset**:
  - Contains individual MRI slices from sagittal or axial views of the lowest three vertebrae and the lowest three intervertebral discs (IVDs).
  - Axial view slices have uniform thickness of 4 mm and center-to-center distance of

    4.4 mm.
  - Majority of slices are in Head-First-Supine position.
- **CT Spine 1K Dataset**:
  - Curated from open sources, totaling 1,005 CT volumes with diverse appearance variations.

- Facilitates research in vertebrae segmentation, labeling, and 3D spine reconstruction.

1.4 Organization of Thesis

The organization of thesis is given as under

- Chapter 1: Represents the introduction related to Lumbar Spine and related disorders. Furthermore, this Chapter also elaborates AI, machine learning and deep learning mechanisms that can be used for Lumbar Spine disorder detection.
- Chapter 2: Discussed the existing work that has been done in the field of Lumbar Spine detection. The scope of the study is also highlighted through this section. The validation criteria are also discussed in this Chapter. Ultimately, a comparative analysis of the literature is presented
- Chapter 3: Elaborates the objectives and Methodology of study
- Chapter 4: Discussed Performance Assessment of the Proposed and Current Methods
- Chapter 5: Concludes the research

Chapter 2

Review of Literature

The process of systematic literature review is presented in this section. This process consists of selection of relevant paper corresponding to Lumbar Spine detection. Papers are selected from the reputed journals and conferences. The overall process begins by selecting the keywords for paper selection. The keywords used for the selection of papers include:

“Lumbar Spine and related disease detection using applications of machine learning and deep learning”

The website based repositories used for the search operation includes IEEE, Springer, Hindawi, Willey and many more. Using the specified keywords, 122 paper were retrieved. The papers are shortlisted based on redundancy and ineligibility. Due to redundancy 23 paper are eliminated. The bibliometric analysis is used as automation tool for detecting eligibility of the selected papers.

Paper selection criteria: Duplicate rejection, ineligibility rejection and technology not included rejection

The reports corresponding to keywords are searched next. The fetched reports associated with the paper selection are 92. The reports are screened and only 12 reports are accepted based on the specified criteria. Overall result generates 30 papers and 12 reports to be included within the literature review section.

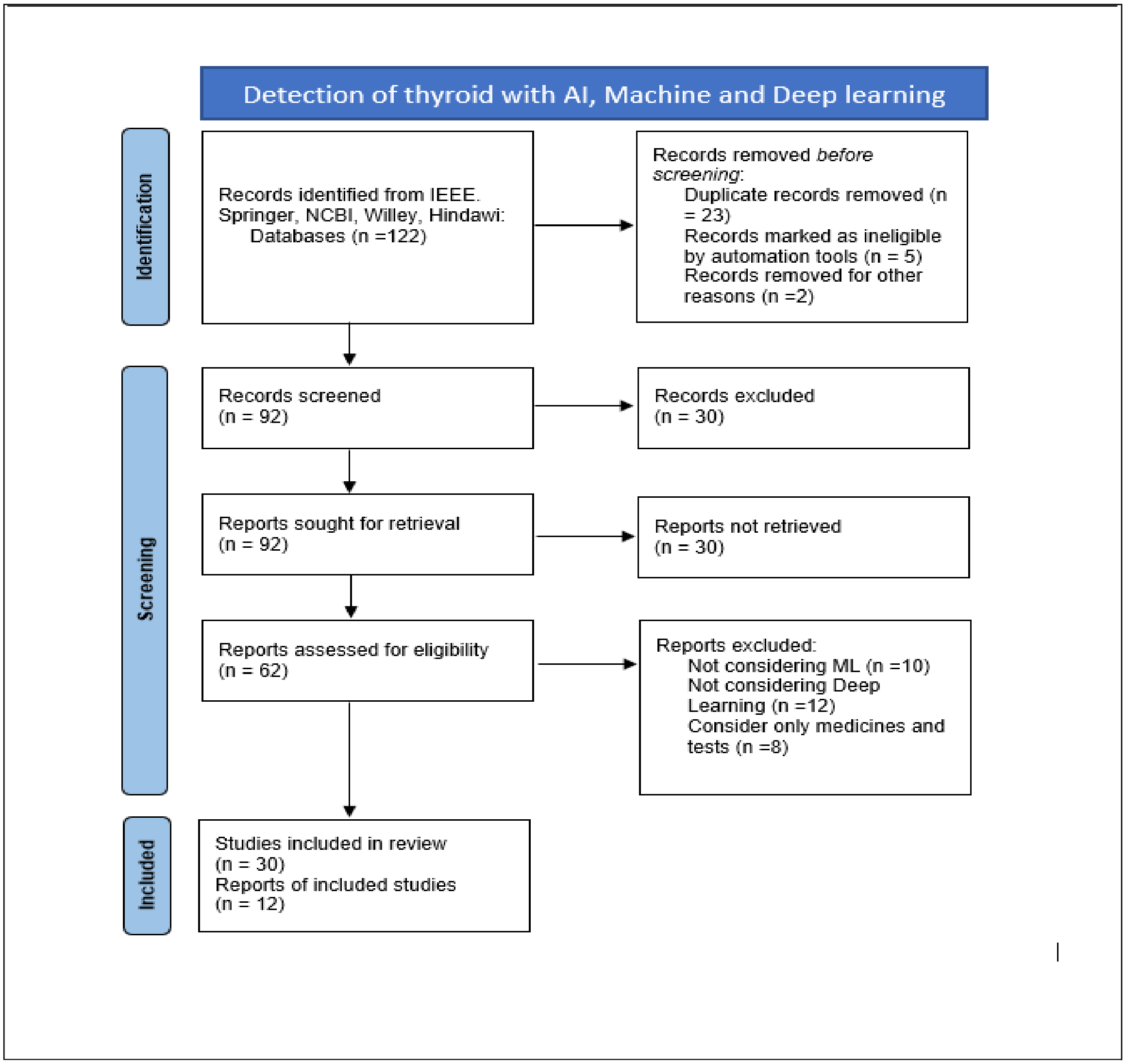


FIGURE 2.1: Process of systematic literature review

2.1 Scope of Literature Survey

The researchers [21], [14] and [3] explored the impact of technology in the health care sector and particularly for detection of common Lumbar Spine related diseases. [2], [88] proposed a soft computing model for detecting Lumbar Spine related diseases at early stage with classification accuracy as key metric. The comparative study was presented to select best possible model. [3] feature selection-based machine learning model was proposed for behavior detection within persons suffering from Lumbar Spine disease. [12] proposed deep learning-based mechanism for early detection of common Lumbar Spine diseases including Anki Spondylitis Lumbar Spine and Spinal Stenosis Lumbar Spine. From the literature survey it is concluded that research towards detection of common Lumbar Spine diseases based on benchmark dataset has been extended from time to time.

Thus, the literature survey recommends in terms of proposed methodology using CNN with linear regression classifier can be used for detection of common Lumbar Spine diseases.

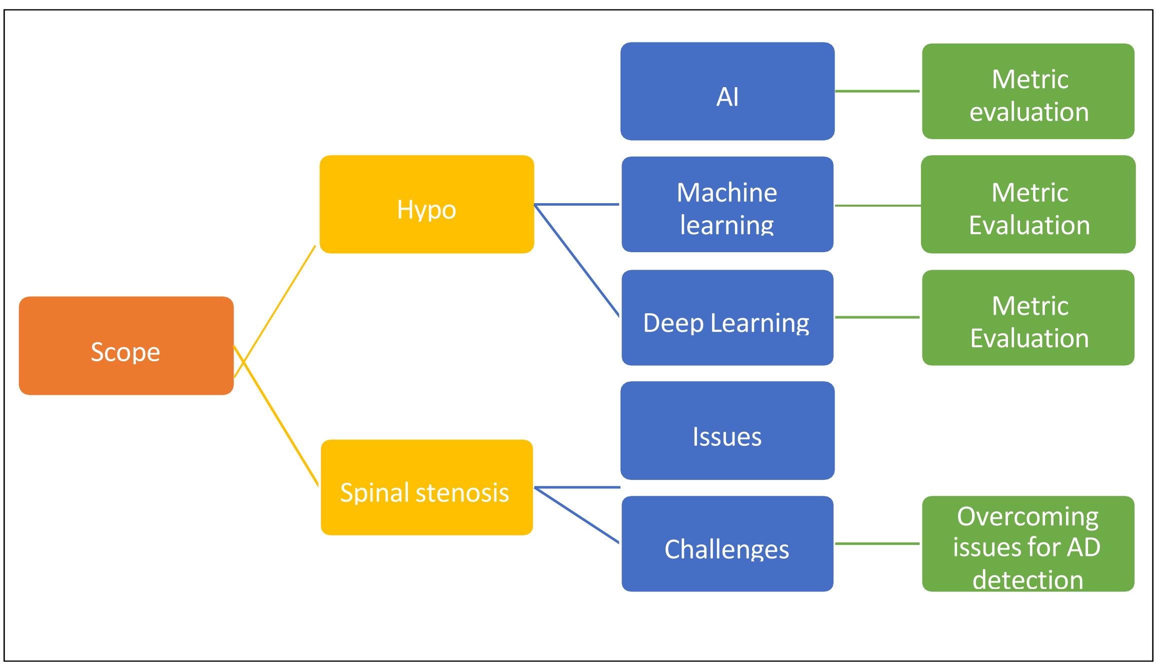


FIGURE 2.2: Scope of study

2.2 Underlying Dataset for Lumbar Spine

This section analyses the datasets used and conduct critical evaluation of techniques employed for detection of Lumbar Spine diseases. The analysis for comparison is also performed to detect best possible mechanism for detection of common Lumbar Spine diseases. Table 2.1 summarizes some important datasets and techniques used for LSD detection

TABLE 2.1: DATASET AND TECHNIQUES APPLIED FOR LSD DETECTION

Reference	Dataset	Attributes	Applied techniques
[94]	MRI Mid Sagittal dataset	Age Sagittal View Axial View Resolution IVD Target	ML
[95]	MURA	Age Sagittal View Axial View Resolution IVD Outcome	CNN
[96]	Lumbar Spine MRI Dataset	Sagittal View Axial View Resolution IVD Outcome	DL
[97]	Vertebral Column Dataset	MRI image dataset with image resolution	ML

The dataset corresponding to the lumbar spine disorder was derived from Kaggle and used to analyze the lumbar spine disorder in the children's data set. A KNN-based classification was performed to achieve high classification accuracy within the range of 90%. [98] proposed a DL-based mechanism for detecting lumbar spine-related disease in children. Furthermore, individual disabilities were also detected using the CNN-based mechanism.

The CNN produces high classification accuracy along with an f-score. The authors [99] proposed a recurrent neural network (RNN) [96] for lumbar spine disorder detection. An RNN-based network will allow the layers to establish connections with cycles or similar features that otherwise are not allowed within a CNN [100]. The RNN-based model will help increase classification accuracy with the complex dataset

The datasets associated with the Lumbar Spine diseases are available through the benchmarked websites. The benchmarked websites include Kaggle, UCI machine learning and data. World. The datasets and associated attributes are described within table 2.2

TABLE 2.2: Dataset for Lumbar Spine Related Research

Dataset	Benchmarked Website	Reference	Attributes
Lumbar Spine Screening	Kaggle	[55]	Age Gender Ethnicity Gender Born with Lumbar Spine Family member with Lumbar spine Country of residence Screening Method Question 1 to Question 7

Lumbar Spine Screening dataset	UCI	[44]	Age Gender Ethnicity Gender
Lumbar Spine prevalence studies	Ultrasound Quarterly(UQ)	[42]	Age Gender IQ Score
Lumbar Spine screening data	Lumbar Spine ultrasound images	[47]	Image dataset

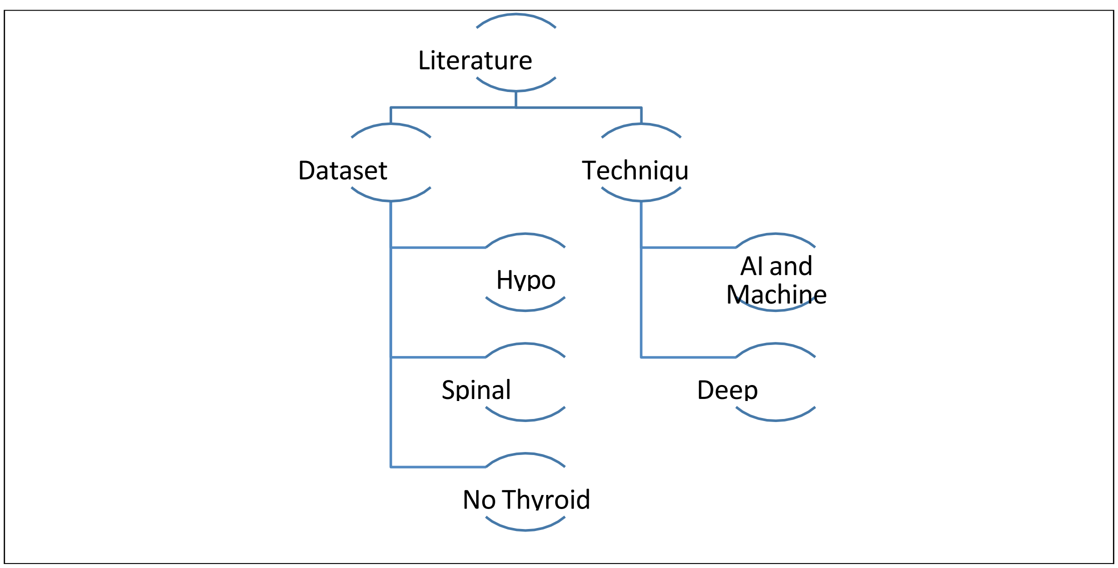


FIGURE 2.3: Process model for Literature survey

The Figure 2.3 Titled "Process Model for Literature Survey" outlines the steps involved in conducting a structured review of relevant studies. It begins with the Literature Survey, which is divided into two main components: Dataset Analysis and Techniques.

Under Dataset Analysis, key focus areas include conditions such as Hypo, Stenosis, and No Thyroid. On the Techniques side, advanced methodologies are categorized into AI and Machine Learning and Deep Learning, highlighting their importance in addressing lumbar disc disorders. This model serves as a framework for identifying datasets, techniques, and conditions for further research. The metric based critical evaluation of literature is presented in the next section.

2.3 AI, Machine Learning and Deep Learning mechanisms for Common Lumbar Spine Disease Detection

2.3.1 AI and Machine Learning based Mechanisms

The process model corresponding to Artificial intelligence and machine learning based literature is given in figure 2.4

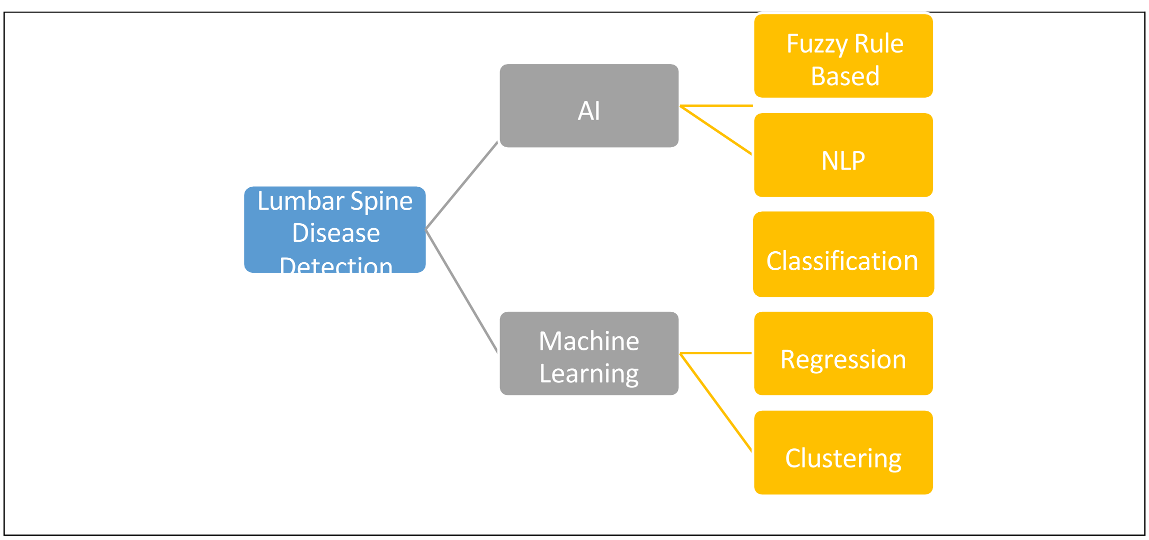


FIGURE 2.4: AI and Machine learning frameworks for Lumbar Spine detection

[48] proposed the application of artificial intelligence to shorten the behavioral diagnosis of Lumbar Spine disorder. The fuzzy logic was applied for forming the decisions regarding lumbar Spine related diseases and diagnosing the behavioral patterns off the children. The applied technique was effective however may not be suitable for detecting rare Lumbar Spine diseases.

[57] discussed the application of AI for the detection of Lumbar Spine related disease. The deep learning based mechanism was used for detecting diseases within the adults. Using adult Lumbar Spine behavioral dataset. the metrics used for evaluation includes classification accuracy, f-score, and specificity. The classification accuracy of 90% justify the work but work in the field of rare Lumbar Spine diseases is still unexplored.

[34] researched the Lumbar Spine behavior detection using machine learning and clustering with radio imaging dataset. Overall process of detection consists of preprocessing mechanisms that includes median filtering, and contrast enhancement to improve the accuracy of the result. After the segmentation, the final phase is classification. The degree of misclassification using multi parametric Radiomics characterization approach was high leading to the low classification accuracy.

[52] proposed functional connectivity classification with AI natural language processing. The classification model uses highly predictive Lumbar Spine gland features for common Lumbar Spine diseases including Anki Spondylitis Lumbar Spine disorder and Spinal Stenosis Lumbar Spine.

[11] proposed mobile interactive technology for the prediction of Lumbar Spine disorder. The Anki Spondylitis and Spinal Stenosis Lumbar Spine disorder were detected effectively using machine learning mode. The lack of standard enforcement was an issue with this approach.

[7] discuss the application of artificial neural network in the reduction of Lumbar Spine spectrum disorder based on the freeze processing be received. The classification accuracy associated with machine learning approach was not that high.

[15] proposed common Lumbar Spine disease detection on the benchmarked dataset. Common Lumbar Spine diseases using machine learning based mechanism detects the disease but falls short of biomarker standard.

[23] proposed AI and machine learning based mechanism using benchmarked dataset. The predicting symptoms severity in Lumbar Spine using AI powered mechanism serve the basis of this mechanism.

[18] researched with the adult dataset for the detection of common and rare Lumbar Spine spectrum disease. Grave's disease to detect was rare due to complicated nature of detection of this disease. However, this research was focused on the detection of common Lumbar Spine related diseases rather than rare Lumbar Spine diseases.

[37] applied the machine learning approach for early detection of Lumbar Spine using the questionnaire and video screening realism. Overall results in terms of Lumbar Spine was quite accurate however further modifications can be applied in case data set becomes large in size.

[51] discussed the applications of Lumbar Spine nodule using ultrasound images. The classification mechanism that were used in this research includes kNN and support vector machine. The support vector machine employed produce highest classification accuracy of 90%.

[14] discussed the application of machine learning algorithms in the prediction of Lumbar Spine related disease. The classification accuracy using the discussed approach is high but larger dataset cannot be tackled

with this approach.

[28] researched the mechanism of data mining in the field of Lumbar Spine disease prediction. The classification accuracy was used as a metric for the prediction purpose. The error rate using this approach is low.

[5] proposed a fuzzy and hard clustering mechanism for the analysis of Lumbar Spine related diseases. The fuzzy system although produce good results but classification accuracy is low in this case due to misclassification degree.

[53] discussed the applications of Lumbar Spine disease using decision tree mechanism. The decision tree however is not consistent with the classification accuracy. Depending upon the dataset, classification accuracy varied between 80% to 90%.

[4] discussed the prediction mechanisms associated with the Lumbar Spine disease detection. The machine learning and artificial intelligence is used for the analysis purpose. The classification accuracy in this case however was 80%.

[30] proposed a segmentation-based mechanism for the prediction of Lumbar Spine diseases. Ultrasound dataset was used in this case for prediction. The execution speed using this approach is less which can be increased by the use of dataset reduction mechanism.

[22] proposed quantitative analysis of the dataset derived from the real time situations. Thus, dataset from the hospitals were fetched and quantitative analysis conducted to determine the Lumbar Spine related diseases. Lack of technology usage causes slow prediction of Lumbar Spine.

[29] discussed the applications of Lumbar Spine related diseases with the help of computer aided mechanisms. The computer aided mechanisms including machine learning and artificial intelligence allow fast

classification of disease.

[50] proposed a fuzzy inference engine for the detection of Lumbar Spine related disease. The text dataset was used in this case. T1 and T3 metrics were used in the prediction of Lumbar Spine related disease.

[33] discussed the adverse effect of late detection of Lumbar Spine related diseases. In case Lumbar Spine disease detected at later stage, it can cause Lumbar Spine cancer as well. Thus, it is imperative to detect Lumbar Spine at early stage.

[19] discussed the effect of Lumbar Spine on metabolism. Due to issues with metabolism, infected person may have obesity and other lifetime diseases. The classification accuracy using this mechanism is high.

[32] discussed the impact of genes on the Lumbar Spine related diseases. The error rate is minimal with this approach.

[56] discussed the factors that leads to the Lumbar Spine diseases. The most contributing factors includes TSH, T1 and T3. All of these factors must be within range to avoid Lumbar Spine related disease. The large dataset cannot be handled with this approach.

2.3.2 Deep Learning and CNN based Mechanism for Lumbar Spine Disease Detection

The deep learning-based mechanism can tackle the large dataset. The machine learning approach can process large dataset but execution speed decreases. To overcome the issue deep learning approaches are discussed in this section for Lumbar Spine disease detection.

[58] purposed convolution neural network for correct pricing the adults having autism disorder. Convolution layer was used in this case to receive

the input and perform filtering. The fully connected layer in this case a received the preprocessed element from the convolution layer and apply the processing strategies. As the overall process is divided hence the complexity has been reduced.

[39] proposed deep learning-based model for the detection of Lumbar Spine disorder. The layered based approach was applied on the Lumbar Spine dataset. The implication of the approach detects 12% of the persons with the CNN based approach.

[46] discussed the application of convolution neural network for the detection of texture and shape-based features for Lumbar Spine disease detection. The Lumbar Spine disorder dataset was fetched from benchmarked website.

[49] discussed the application of deep learning in the form of computer aided analysis to detect Lumbar Spine disorder within person. The benign and malignant Lumbar Spine diseases are detected with the deep learning-based approach in this case.

2.4 Dataset Validation

The appraisal for internal and external validity undergoes testing in this section. Internal validity pertains to the study's internal consistency within the research context.

2.4.1 Internal Validity

In this instance, internal validity pertains to the strong methodology and systematic literature review. The degree of bias greatly influences internal validity. To assess internal validity, the NOS scale [24] was employed. This scale delineates various criteria related to selection bias, comparability, and exposure. These criteria are utilized to assess the internal validity of the results.

Selection

A) Is the case definition sufficient?

a) Certainly, with independent confirmation*

b) Yes, such as record linkage or relying on self-reports

c) No explanation provided

B) Are the cases representative?

a) Sequential or clearly representative series of cases*

b) Potential for selection biases or not mentioned

C) Selection of Control Groups

a) Community-based controls*

b) Hospital-based controls*

c) Absence of description

D) Explanation of Control Groups

a) Absence of disease history (endpoint)*

b) No specification of the source

Comparability

A) Comparing of Cases and Controls Based on Design or Analysis

a) Study controls for lumbar spine disorder*

b) Study controls for any supplementary factor***** Age, family history, regular alcohol drinker, regular smoker and many more.

These variables, often referred to as individual characteristics or personal traits, have been described in Table 2.3

Table 2.3: Demographic Table

Demographic Characteristics	Patients (n = 359)	Healthy Controls (n = 708)
Age (years)	359 ± 0.8	359 ± 0.8
Gender (M/F)	5M / 5F	6M / 6F
Medical History	100%	0%
Ethnicity	70% Group A 30% Group B	80% Group A 20% Group B
BMI (kg/m²)	359 ± 0.8	359 ± 0.8

Factors: These can be considered as influencing factors or variables.

Demographics: This term may encompass age, marriage status, education level, and sometimes family history.

Lifestyle: This can encompass regular alcohol drinking and smoking habits as aspects of one's lifestyle.

Socioeconomic Profile: This term may encompass education level and marriage status.

Exposure

A) <u>Ascertainment of Exposure</u>

a) Verified records (e.g., surgical records) *

b) Structured interview conducted blindly to case/control status*

c) Interview not conducted blindly to case/control status

d) Written self-report or medical record exclusively e) Absence of description

B) <u>Consistency in Ascertainment Method for Cases and Controls</u>

a) Affirmative*

b) Negative

C) <u>Non-Response Rate</u>

a) Equal rate for both groups*

b) Description of non-respondents provided

c) Different rate without specification

The selection bias is insignificant given the samples chosen for analysis. in most literatures were oriented towards people admitted within the hospital or through benchmark websites. It's probable that they have the socio behavioral. Even if certain subjects lack lumbar spine, the selection bias remains insignificant.

The keywords in this case includes Lumbar Spine, Anki Spondylitis Lumbar Spine and spinal Stenosis. The study delineates allocation concealment prior to assigning patients to various groups, thereby reducing selection bias.

Performance bias suggests that all group of patients receive a comparable standard of care. However, in this literature review, patients with lumbar spine issues were given preferential treatment over others. This could be attributed to the fact that not all patients in the hospital are affected by the same condition.

Moreover, there is no indication of observer bias in the study. This is because the researcher remains unaware of the interview and questionnaire outcomes. Identifying detection bias is crucial in assessing subjective outcomes related to the lumbar spine. These subjective outcomes indicate an inverse relationship among psychiatric conditions and the likelihood of lumbar spine issues. Furthermore, there is no indication of attrition bias in the study, which concerns missing data. No sample data has been omitted, and addressing missing values using mean or mode-based approaches ensures that attrition bias is not present in the study.

Reporting bias arises when there is a significant disparity between the disclosed and undisclosed outcomes. Yet, within this study, there is no evidence of reporting bias.

The assessment of internal validity was confirmed using the NOS scale. Moreover, errors in collecting the data can have a detrimental impact on the results. This implies that any errors within the samples (359 cases and 708 controls) could affect the classification outcome.

2.4.2 External Validity

External validity pertains to the capability to generalize research findings to other populations or contexts. This category of validity involves accessing the sample size, methodology, studied population, and research context. In the literature study, the sample size consists of 359 cases and 708 controls, which is adequate for examining the correlation between daily activities and the likelihood of lumbar spine-related diseases. All samples were gathered from hospitals in Korea, and non-probability sampling was employed to ensure the validity of the collected sample. The literature study utilizes a non-probability sampling method known as convenience sampling. This method was chosen because the samples were readily accessible to the researcher [6]. Accurate patient information is essential for assessing the impact of vegetable and fruit consumption on the likelihood of lumbar spine-related diseases. To achieve this, both Qualitative and quantitative research methodologies can be utilized [40]. In the proposed study, both interviews and questionnaires are employed to collect data from the population. Interviews were carried out with patients admitted to hospitals in various locations. Questionnaires were employed to allow patients who were uncomfortable with attending interviews to respond in writing. Therefore, the population involved in the research conducted by the literature study researchers comprised patients admitted to hospitals. The population is definitely a great choice for the research carried by researchers in literature study. The research context corresponding to the research carried by researchers in literature study is associated with identification of different measures that can be taken to avoid Lumbar Spine and related disease. For this purpose, the research studied the impact of technology on the chances of reduction in Lumbar Spine and related disease. It was discovered that Lumbar Spine and related disease shows correlation with some of the daily activities but not all of them. Put differently, we can assert that there is a causal relationship between the exposure and the outcome.

2.5 Comparative Analysis

This section presents the comparative analysis of Lumbar Spine related disease prediction mechanisms along with datasets and metric used.

TABLE 2.4: Comparative analysis of techniques used for disease prediction

Author's	Technique	Dataset	Metrics
[10]	Clinical analysis	Text Dataset derived from Kaggle	Execution time, accuracy
[38]	Meta-analysis	Text and image datasets derived real time and from benchmarked websites	Classification accuracy
[8]	Clinical analysis and treatment of grave disease	Real time dataset	Classification accuracy
[20],[71]	Machine learning based mechanism for Lumbar Spine detection	Dataset derived from Kaggle	T1, T3 and TSH detection and prediction
[15]	Deep and machinelearning based mechanism for Lumbar Spine related disease	Benchmarked dataset derived from Kaggle	Classification accuracy based on T1, T3 and TSH parameters
[13]	Segmentation based mechanism for Lumbar Spine classification	Ultrasound images	Color, shape and size based attributes for disease prediction.
[27]	Data mining techniques for disease prediction	Text based dataset derived from Kaggle	T1, T3 and TSH for disease prediction

The machine learning and artificial intelligence frameworks are commonly employed for the prediction of Lumbar Spine related disease. The real time and benchmarked datasets are used for the prediction purpose. The issue (replace with limitation or disadvantage) with the existing approach is lack of neural network-based model formation for the prediction of disease from test datasets.

2.6 Research Gap

Despite significant advancements in the application of machine learning (ML), artificial intelligence (AI), and deep learning (DL) for Lumbar Spine Disease (LSD) detection, notable gaps persist in the literature. Current studies have focused primarily on common Lumbar Spine disorders and datasets like MRI mid-sagittal views and Kaggle-based datasets, achieving promising results in classification accuracy, particularly with CNN and RNN models. However, these studies have limitations when addressing rare Lumbar Spine diseases. Few research efforts have been directed toward rare conditions such as Ankylosing Spondylitis or Spinal Stenosis, limiting the generalizability of proposed models.

Another gap lies in the reliance on benchmark datasets with limited diversity in patient demographics, such as age, ethnicity, and specific comorbidities, which may not reflect real-world variability. Furthermore, while segmentation and classification techniques have been employed, the high misclassification rate in multiparametric Radiomics characterization points to inadequacies in image preprocessing and feature extraction methods.

There is also a lack of integration of multi-modal datasets, such as combining clinical, imaging, and behavioral data, which could improve diagnostic accuracy. Additionally, research on the scalability and real-world implementation of these AI and ML models in clinical settings is sparse. Addressing these gaps is essential for developing comprehensive, accurate, and clinically relevant solutions.

Chapter 3

Objectives and Methodology

3.1 Objectives

The objective is to achieve an expert system that can detect multiple Lumbar Spine types. The neural network-based mechanism can be suggested to achieve high classification accuracy. The objective of study is given as under:

1. To collect and prepare the dataset of MRI (Magnetic Resonance Imaging) and CT (Computerized Tomography) images for detection and classification of Lumbar disc disorder.
2. To Pre-process and enhance the MRI and CT images for obtaining more meaningful indicative features.
3. To analyze the the impact of segmentation and feature extraction algorithms for identifying the region of interest.
4. To propose deep learning based medical expert system for multiple Lumbar disc disorder.
5. To study and investigate the proposed system with few performance metrics.

Objective Description

- Lumbar spine disorders are significant challenges in the healthcare domain, affecting millions of individuals globally. Establishing an expert system for the detection of multiple lumbar spine-related diseases is imperative for early intervention and effective treatment planning. This study aims to develop a neural network-based mechanism, integrating normalized and pre-processing layers, to enhance classification accuracy. Additionally, the study seeks to create a Convolutional Neural Network (CNN) model for training and testing lumbar spine disease detection results, and to compare the proposed system with a CNN model used in conjunction with a Linear Regression Classifier.

- Building a Lumbar Spine Disease Detection System: The primary objective of this study is to construct an expert system capable of detecting multiple lumbar spine- reated diseases. This entails leveraging the power of machine learning, specifically neural networks, to analyze medical imaging data and accurately classify various lumbar spine disorders. The expert system aims to contribute to early diagnosis, enabling timely intervention and personalized treatment plans.

- Introducing Normalized and Pre-processing Layers within Neural Network: To enhance the robustness and effectiveness of the neural network, the study proposes the introduction of normalized and pre-processing layers. Normalization techniques, such as batch normalization, aim to stabilize and accelerate the training process by normalizing the input data, reducing internal covariate shifts, and improving convergence.

.

- Creating a CNN Model for Training and Testing: Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification tasks, making them well-suited for medical imaging analysis, including lumbar spine disease detection. The study aims to develop a CNN model specifically tailored for training and testing lumbar spine disease detection results.

The CNN architecture is expected to leverage its ability to automatically learn hierarchical features from medical images, capturing both local and global patterns indicative of lumbar spine pathologies. The objective is to optimize the CNN model for accuracy, efficiency, and scalability, ensuring its applicability to diverse lumbar spine-related diseases.

- Comparative Analysis with CNN and Linear Regression Classifier: To evaluate the effectiveness of the proposed system, the study introduces a comparative analysis between the neural network-based approach and a CNN model coupled with a Linear Regression Classifier. This comparison seeks to assess the relative strengths and weaknesses of the two methodologies in the context of lumbar spine disease detection.

  The objective here is twofold: firstly, to ascertain the performance gains achieved by the proposed system with normalized and pre-processing layers within the neural network, and secondly, to compare this enhanced neural network approach with a CNN model augmented by a Linear Regression Classifier. The study aims to provide insights into the advantages of leveraging neural networks over traditional classification techniques in the domain of lumbar spine disease detection.

- Methodology: The study's methodology involves the collection of diverse and representative medical imaging datasets containing images of lumbar spine disorders. The datasets will be pre-processed to ensure consistency and quality. The neural network architecture, incorporating normalized and pre-processing layers, will be designed and implemented. Simultaneously, a dedicated CNN model for lumbar spine disease detection will be developed.
  The training and testing phases will involve extensive experimentation with different hyper parameters, normalization techniques, and pre-processing strategies. Comparative analysis will be conducted with the

CNN model using a Linear Regression Classifier to gauge the performance improvements achieved by the proposed neural network-based expert system.

- Significance and Expected Contributions: The significance of this study lies in its potential contributions to the field of lumbar spine disease detection. By building an expert system grounded in neural network technologies and enhancing it with normalized and pre-processing layers, the study aims to push the boundaries of classification accuracy. The comparative analysis with a CNN model using a Linear Regression Classifier provides valuable insights into the effectiveness of advanced neural network architectures in comparison to traditional approaches.
- Challenges and Considerations: Acknowledging potential challenges is crucial for the success of the study. Challenges may arise in the form of data variability, the need for fine-tuning hyper parameters, and addressing potential biases in the datasets. Ensuring the generalizability of the proposed system across diverse patient populations and lumbar spine pathologies will be a key consideration.

  Ethical considerations, particularly in handling sensitive medical data, privacy concerns, and ensuring transparency in model decision-making, will be paramount. The study will adhere to ethical guidelines and prioritize responsible practices in data collection, model development, and result interpretation.

- **Conclusion:** In conclusion, this study aims to advance lumbar spine disease detection through the development of a neural network-based expert system. The creation of a dedicated CNN model and the subsequent comparative analysis with a CNN model using a Linear Regression Classifier will shed light on the efficacy of advanced neural network approaches in the context of lumbar spine disease detection. The study's outcomes have the potential to contribute significantly to the improvement of diagnostic capabilities in lumbar spine disorders, paving the way for early intervention, personalized treatment plans, and ultimately, better patient outcomes. Through rigorous methodology, thoughtful considerations, and ethical practices, this study endeavors to make a meaningful impact on the intersection of machine learning and musculoskeletal healthcare.

3.2 Research Gap

In the literature, the author works on improvement in the performance of medical diagnostic system using various AI algorithms. So, there is scope to apply latest deep learning algorithms for proposed work. There is not a suitable expert system to know the in-depth underlying pathology behind every Lumbar disc disorder so, the proposed diagnostic system will include multiple lumbar disc disorder. Expert system in the field of health care provide virtual intelligent assistant for doctors. It can process large amount of data allow accurate decision making. So far training process focus on all the areas present within the image and hence huge time is consumed during training phase. ROI with segmentation mechanism will reduce the execution time. In addition, automatic training mechanism is conducted rather than manual training process. The diagnose process will be faster and accurate using the proposed mechanism. Diagnose rules and regulations are defined and patient can use the system with least consultation from the doctor to diagnose the disease. Both time and effort required for diagnoses will be considerably reduced using this system.

3.3 Proposed Design Methodology

A thorough explanation of the suggested model and its associated methodology is presented in the subsequent sections.

3.3.1 Problem Formulation

The problem associated with neural network approach is lower classification accuracy during the prediction of the Lumbar Spine diseases. The approach used to detect Lumbar Spine does not tackle the issues of dataset effectively. This means separate model is required to be designed for different Lumbar Spines and text base datasets. In addition, feature excretion is an issue. In case of image base dataset, boundary identification is a big problem. In text base dataset, problem of missing values and normalization exists. Normalization mechanism converts the data that is in string format into lower case and same decimal format is maintained using normalization mechanism in text dataset. The neural network-based approach is applied to determine the correct classification could yield best possible result. The classification accuracy also is a problem. Deep learning neural networks are adept at managing extensive datasets, whether they comprise images or text- based information. The classification accuracy can be an issue in case multiple diseases are to be categorized.

The reason for the same is missing pre-processing mechanism along with using simple classifiers like decision tree. To overcome the issue, convolution neural network with pre- processing and ensemble-based classification is desired.

3.3.2 RESEARCH MOTIVATION

The LSD detection-related issues discovered in the existing literature are in terms of feature extraction and selection [66], [75]. Extracted features using a NN-based approach do not show any correlation with the categorical outcome variable. Consequently, feature selection is without correlation analysis. Overall, the degree of misclassification within such approaches was minimal. The pre-processing mechanism also does not include validation and normalization. Thus, the calculation complexity is high [67-70] Therefore, there is a strong motivation to perform this research.

Our research focuses on addressing the limitations found in existing research on LSD detection. While prior studies have explored various aspects of LSD detection, they often lack effective feature extraction and selection methods that capture the correlation between extracted features and the categorical output variable. Additionally, our work extends beyond pre-processing by incorporating validation and normalization steps, which are often overlooked in existing methodologies. By introducing a linearity-based model within the CNN framework for feature selection, we aim to enhance classification accuracy and reduce misclassification rates. These enhancements were not adequately addressed in prior research, and our work strives to bridge this gap.

3.3.3 Proposed Model Description

The proposed work begins with the data acquisition phase; data is collected from both benchmarked and real-time datasets.

In the pre-processing mechanism, any noise from the dataset is removed. Feature extraction and selection are applied to determine critical features for the prediction of LSD. Finally, classification is performed.

The categorization variable associated with the mechanism is “Ankylosing spondylitis Lumbar Spine”, “Spinal stenosis Lumbar Spines”, “No Disease”. The flow of the proposed work is given within figure 3.1.

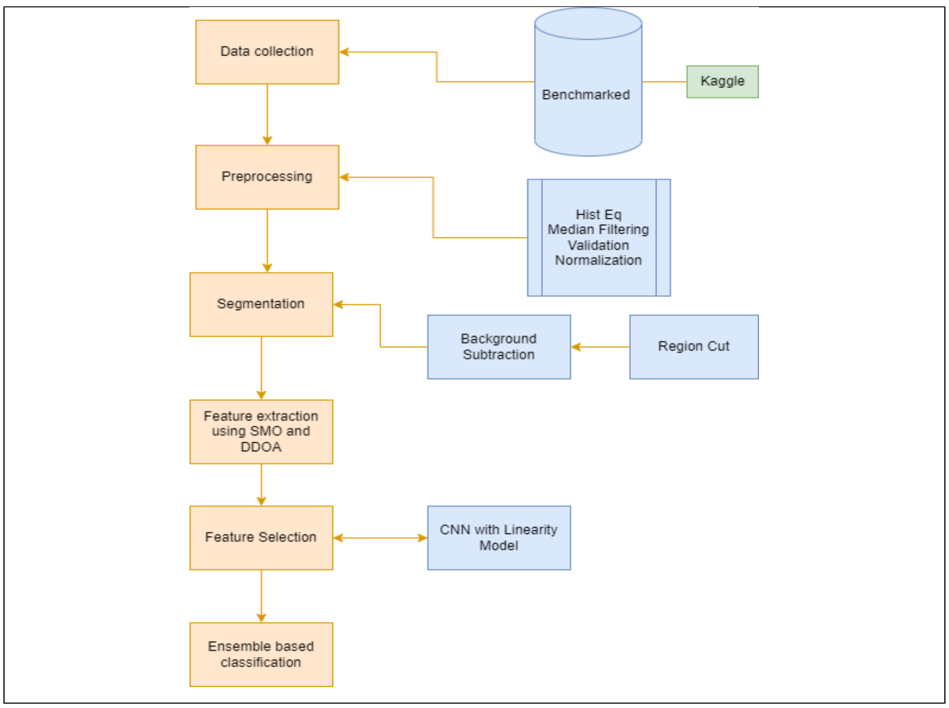


FIGURE 3.1: Flow of the proposed work

The Algorithm Used for Lumbar Spine Disease Detection

- Input Dataset

  Data=input(dataset)

- Check for noise
  from the dataset
  Final_data=omit
  NA(Data)

- Preparing CNN for segmentation

Input layer for receiving pre-processed data

Convolution layer for feature extraction. Multiple convolution layers are used for feature extraction

Multiple pooling layers for storing feature

Fully connected layers to receives the features values from the processing layer

Output layer for generating feature extraction result

- Classification using linear regression and Predictors Approach Defining value of regression coefficient

  If($coff_i$<Coff)

  $Class_i$=$Result_j$

$$Coff = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (1)$$

Classification using Predictors

ClassPredictorCNN=CNN_Predictor(Result)

- Output result in terms of classification accuracy, specificity, sensitivity, and F- score

In the context of SOM_POS (Self-Organizing Map Position), this equation is used to determine how closely an input data point matches a neuron (node) in the SOM grid. x and y are features starting value and x_i and y_i are features end value The smaller the Coeff value, the closer the input is to the node, indicating similarity. This distance metric helps in updating node weights during training, clustering similar data points for optimal feature mapping and visualization.

The overall operation of proposed work begins with loading the dataset. After the dataset is loaded, it is examined for null values. These values are irrelevant and hence must be eliminated. OmitNA method is used for this purpose. After eliminating the null values, CNN model is built for feature extraction. The finalized feature selected with fully connected layer is presented to the classification framework. The input metrics in this case are T4, T3 and TSH. The result is presented with the classification accuracy.

Lumbar spine pathology detection is a crucial aspect of musculoskeletal Healthcare and leveraging advanced technologies such as Convolutional Neural Networks (CNNs) can significantly enhance diagnostic accuracy. This study focuses on the comprehensive pipeline for lumbar spine pathology detection, starting with data preprocessing, Convolutional Neural Network segmentation, and concluding with classification using linear regression and predictors. The aim is to achieve a robust and accurate model capable of generating results in terms of classification accuracy, specificity, sensitivity, and F- score.

3.3.3.1 Data Acquisition

The terms data acquisition means data collection. Data can either be collected from benchmarked websites or through real time sources. To perform the proposed operation, benchmarked website Kaggle is used. The dataset contains three critical parameters including TSH, T4 and T3 and using these attributes Lumbar Spine disorder is predicted.

3.3.3.2 Data Preprocessing

It refers to the cleaning, transforming, and integrating of data [85]. The objective of data preprocessing step is to better the quality of the data and to make it appropriate for further analysis. Encoding categorical data, scaling the data are required to be done to reduce the complexity of the process.

The pre-processing mechanism includes many sub-phases. The first phase is histogram equalization which enhances the contrast and overall clarity of the MRI images. After the enhancement process, median filtering is applied to remove salt and pepper noise. The validation mechanism is applied to test the restored image against the original image. If the restored image correlates with the original image, the restored image is retained.

The first step in the pipeline involves handling the input dataset. The raw dataset is loaded using the 'input' function, converting it into a format suitable for analysis. To ensure the robustness of subsequent processing steps, the data is checked for noise, and any missing values are addressed through the 'omitNA' function, resulting in a clean and complete dataset (Final_data). [105-110] discusses that that preprocessing is critical to enhance the quality of input information, making it suitable for subsequent analysis. The omission of missing values ensures that the neural network is provided with reliable and complete information, minimizing potential biases and errors in the subsequent stages of the pipeline. The pre-processing algorithm is given below as Algorithm 1.

Algorithm 1: Pre-processing (Image)

Input: Image, the image for enhancement and noise removal.

Output: Hist_Image, the image after histogram equalization. Filt_Image, the image after median filtering.

Assumptions: Indices, I and J are the index variables.

Corr= calculate the correlation Norm= normalized image
rows =number of rows in the image cols =number of columns in the image.

1. Input Image
2. Hist_Image=hist(image)
3. Filt_Image=median_filter(Hist_Image)
4. For I=1 to rows(Image)

```
For J=1 to cols(Image)
    If(corr(Image,Filt_image)>0) then
        Validated(index)=Image
        Index=Index+1
    End of If
    End of For
End of For
Normalized=norm(Validated)
```

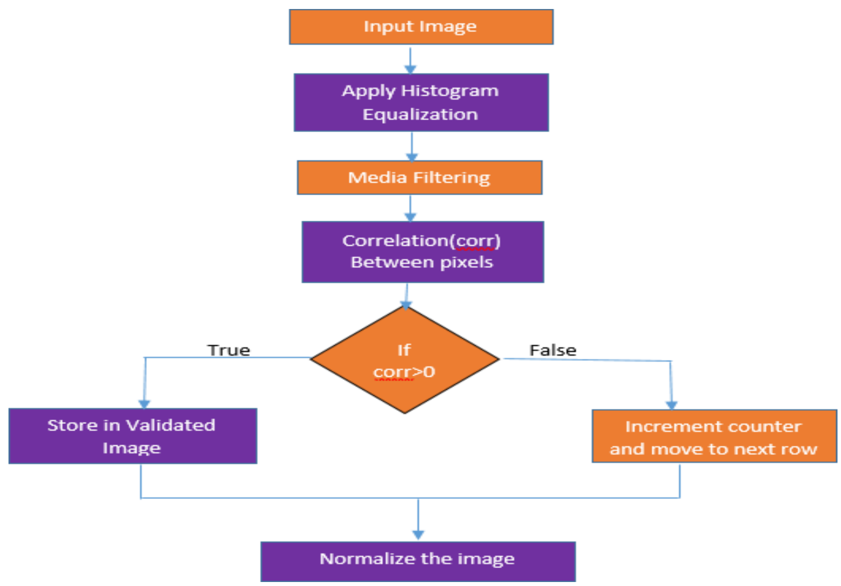


FIGURE 3.2: Flow of the Pre processing

3.3.3.3 Segmentation

With the preprocessed dataset in place, the next step involves preparing a dataset for segmentation. This step is pivotal for extracting relevant features from medical images associated with lumbar spine pathology.

Background subtraction eliminates the background of the image and only the ROI will be retained. The ROI is carefully chosen to focus on the specific anatomical regions of interest within the lumbar spine. In this study, the ROI corresponds to the intervertebral disc (IVD) region, which is a critical component of the lumbar spine and can provide valuable insights into potential abnormalities or diseases. The IVD region is chosen as the ROI due to its significance for lumbar spine health and its susceptibility to various spinal conditions. IVDs act as shock absorbers between adjacent vertebrae and are composed of an inner nucleus pulpous surrounded by an outer annulus fibrosus. [91] [93]..

Normalization and segmentation of MRI images are critical for selecting the intervertebral disc (IVD) region accurately. Segmentation isolates the IVD region, starting with pre-processing steps such as denoising and edge detection using methods like Sobel or Canny. Traditional techniques like Otsu's thresholding or watershed segmentation are limited by noise and overlapping structures. To address this, a new approach integrates deep learning models like U-Net or Mask R-CNN for precise segmentation, coupled with active contour models for boundary refinement. This hybrid approach, leveraging transfer learning with pre-trained networks like ResNet50, ensures improved accuracy and adaptability. Post-processing using morphological operations refines the segmented regions.

The selection of the IVD as the ROI is based on both anatomical and pathological considerations. It is the region where most spinal issues, such as herniated discs or degenerative changes, are likely to manifest. The ROI selection is further refined by considering radiological expertise, where a team of experienced radiologists or medical experts is involved in defining the precise boundaries of the IVD region to ensure accurate analysis. The chosen ROI's boundaries are applied consistently across the dataset to maintain uniformity in analysis. This rigorous ROI selection process helps to focus the analysis on the most relevant and informative parts of the lumbar spine MRI images. It allows for effective feature extraction and subsequent classification, enhancing the accuracy of disease detection and contributing to the overall robustness of the study's methodology. This will be accomplished by reducing 255 (the maximum intensity of the pixels) from the intensity levels in the image: $Back_{Sub} = 255 - I$ where "I" is the intensity level in the image. $Back_{sub}$ indicates the background-subtracted image. The region cut is applied to the background-subtracted image to extract the meaningful portion of the image using Algorithm

BS_Image_Reg_Cut(img, Thresh)

Input: img, image from the training dataset.

Thresh, values corresponding to the boundaries of the image.

Output: B establish the necessary region's border.

Assumptions: The color assigned to the interior region filling is referred to as the color

```
// Load the image and get its dimensions
[image, width, height] = imread(img)
// Set the threshold intensity
thresh = 128
// Initialize arrays to store boundary points
boundary_x = []
boundary_y = []
// Module to identify the boundary of the image
For x = 1 to width
    For y = 1 to height
        If (intensity(img(x, y)) >= thresh)
          // Add the boundary point to the arrays
            Append(boundary_x, x)
            Append(boundary_y, y)
        End of If
    End of For
End of For
```

3.3.3.4 Feature Extraction

Feature extraction is applied over the extracted region. The process becomes faster as the entire image is not involved in feature extraction or feature vector generation. Feature extraction is accomplished using the SMO along with the digital differential analyzer approach.[90] The utilization of SMO is advantageous in this scenario, illustrating two crucial facets of swarm intelligence: organization and task distribution. It takes into account both local and global solutions. In case of local solution proves to be superior to the globalone, it will replace the global one. During the local leader phase, the local solution is derived, where the present search space is adjusted according to the progress of the phase.The local leader phase creates a path position to be explored by the SMO, signifying the feature's location in the MRI image as shown in Equation 2. To update the trial position inthe sample space, the leadership position, the current energy position, and a randomly chosen entity within the group are used. The value of the perturbation rate determines the size of the solution space ([61], [77]

$$
\begin{aligned}
SMO\_pos(i,j) \\
&= \{(SMO\_pos(i1,j) + r(0,1) * (N\_e(i1) - Local\_sol(i1,j)\ ) \\
&+ r(-1,1) * (N\_e(i1) - Global\_sol(i1,j)\ )\ \ where\ r(0,1) \\
&\geq Per\_rate@SMO\_pos(i1,j)\ \ where\ r(0,1) < Per\_rate\ )
\end{aligned} \tag{2}
$$

The equation defines the update rule for the position SMO_pos(i,j)where i and j denote indices of the solution. It incorporates local and global solution adjustments. The position is influenced by the current value SMO_pos(i1,j)random factors r(0,1) and r(−1,1) and differences between the neighborhood solution Ne(i1)N_e(i1)Ne(i1) and both the local Local_sol(i1,j)and global solutions Global_sol(i1,j). The random values ensure exploration, controlled by a performance rate threshold Per_rate. This process balances exploration and exploitation for improved optimization.

Where, 'r(0,1)' represents a random sample taken from the sample space, which also encompasses the perturbation rate. The use of a split and shift mechanism, in conjunction with perturbation, enables precise feature extraction. The objective function is influenced by factors such as throughput, energy efficiency, and the degree of imbalance. If the new solution surpasses the fitness values of the previous solution, it supersedes the existing solution. The fitness value associated with the present solution guides the process of node selection. The fitness function is represented in Equation (3).

$$Fitness = \begin{cases} 1\ if\ N_{g(i1)} > 0 \\ 0\ if\ N_{g(i1)} \leq 0 \end{cases} \qquad (3)$$

The fitness equation above suggests that following the transmission of frames, each node must possess certain attributes; otherwise, the present solution is not valid. Equation (4) provides the objective function that supports the overall solution.

$$Objective = \sum \max(E_c) Max(T) Min(Im_{deg}) \qquad (4)$$

Here ‘E_c’ indicates the residual features that must be maximized, ‘T’ indicates throughput (number of features), and ‘Im_{deg}’ is the degree of imbalance that must be minimized. The global leadership phase is critical and is responsible for updating the search space just like the local leadership phase but in a different manner as compared to the local leadership phase. The spider monkey (MRI image) gets updated or selected based on the probability value calculated by (5).

$$ProbabilityNselect = 0.9 * \frac{fitnessn(i)}{\text{Maxp−fitness}\boldsymbol{wuu}\text{rent}} + 0.1 \qquad (5)$$

Maxp represents the peak expected fitness value and $fitness_{current}$ denotes the present fitness function value. $Fitness_{n(i)}$ represents the fitness score of the present node. The overall probability equals '1,' further divided into '0.9' and '0.1' to determine the likelihood of node selection.

After this, a new trial is repeated once more through the fitness equation. Then, the fitness score of the existing sensor node is compared with that of the newly computed node. The optimal sensor node is chosen for transmission using Algorithm

DDA

The Digital Differential Analyzer (DDA) optimization approach is a method used to identify and select the most relevant features for a given task, such as image classification or pattern recognition. In the context of a Convolutional Neural Network (CNN) for lumbar spine analysis, DDA can be employed to enhance the network's performance by optimizing the feature selection process.

The rationale behind using the Digital Differential Analyzer (DDA) algorithm for feature acceptance and rejection in an image dataset lies in its ability to ensure precise and efficient line generation between data points.

For image datasets, this process is crucial because they often contain large amounts of high- dimensional data. The DDA algorithm efficiently maps feature coordinates, ensuring that the selection process is both computationally feasible and robust. This approach is particularly beneficial in applications like medical imaging, where extracting and focusing on critical features (e.g., boundaries of anomalies) directly impacts diagnostic accuracy. By systematically evaluating features with DDA, the process balances precision and computational efficiency, enhancing the reliability of the feature selection and rejection methodology.

DDA Optimization Approach

Constructing the Line

The DDA algorithm constructs a line in the feature space that represents the optimal combination of features for the task. This line is determined based on certain criteria, such as minimizing error or maximizing accuracy.

Feature Acceptance and Rejection

As the line is constructed, features that align well with this optimal path are accepted, while those that deviate significantly are rejected. The criteria for acceptance typically involve the contribution of each feature to the overall performance of the CNN. Features that improve classification accuracy or provide meaningful information about the lumbar spine images are retained.

Initialization: Start with an initial set of features and an initial direction for the line.

Evaluation: Assess the performance of the features at each point along the line, using metrics like accuracy or loss.

Iteration: Gradually adjust the line's direction to improve the feature set, iterating until the optimal combination is found.

Selection: Accept features that consistently contribute positively to performance and reject those that do not.

Benefits

This optimization process ensures that the CNN focuses on the most relevant features, enhancing its ability to accurately diagnose lumbar spine conditions.

In summary, the DDA optimization approach constructs an optimal line in the feature space, accepting features that enhance performance and rejecting those that do not, thereby optimizing the feature selection for improved CNN performance.

Spider Monkey Optimization (SMO) is a swarm intelligence algorithm inspired by the social foraging behavior of spider monkeys. It is particularly effective for feature extraction in lumbar spine analysis due to its robust search capabilities and adaptability to complex optimization problems.

Why SMO is Effective for Feature Extraction

Swarm Intelligence

SMO leverages the collective intelligence of a group of spider monkeys to explore the feature space efficiently. Each monkey represents a potential solution, and they communicate to share information about good solutions, which helps in avoiding local minima and converging to a global optimum.

Exploration and Exploitation Balance

SMO balances exploration (searching new areas of the feature space) and exploitation (refining known good solutions). This dual strategy ensures a comprehensive search for the most relevant features, enhancing the CNN's ability to accurately analyze lumbar spine images.

Dynamic Adaptation

The algorithm dynamically adapts to the problem's requirements by adjusting the search behavior based on the quality of current solutions. This flexibility allows SMO to effectively handle the high-dimensional and complex feature spaces typical in medical image analysis.

High Convergence Rate

SMO has a high convergence rate, meaning it quickly identifies optimal or near-optimal feature subsets. This efficiency reduces computational costs and speeds up the training process of the CNN.

Conclusion

In summary, Spider Monkey Optimization is effective for feature extraction in lumbar spine analysis due to its swarm intelligence, balance between exploration and exploitation, dynamic adaptation, and high convergence rate. These characteristics enable it to select the most relevant features, thereby improving the performance and accuracy of the CNN model.

Combining Digital Differential Analyzer (DDA) optimization and Spider Monkey Optimization (SMO) for lumbar spine prediction leverages the strengths of both methods to enhance feature extraction and selection, ultimately improving the performance of Convolutional Neural Networks (CNNs) in medical imaging.

Why Combine DDA and SMO?

Complementary Strengths

DDA excels in constructing an optimal path in the feature space by iteratively refining feature selection based on their alignment with the constructed line. This method ensures that only the most relevant features are retained, reducing dimensionality and enhancing model efficiency.

SMO, on the other hand, uses swarm intelligence to explore the feature space thoroughly. Its ability to balance exploration and exploitation helps avoid local minima and ensures a diverse search for potential feature subsets.

Enhanced Feature Selection

Combining DDA's precise, path-based selection with SMO's robust, swarm-based exploration provides a comprehensive feature selection mechanism. DDA can guide the initial feature selection process, focusing on key features, while SMO can further refine and explore additional relevant features, ensuring no significant information is overlooked.

Improved Model Performance

The combined approach ensures that the CNN model for lumbar spine prediction is both efficient and accurate. DDA reduces computational complexity by focusing on the most promising features early on, while SMO enhances the model's robustness by thoroughly exploring the feature space and fine-tuning the selection.

Conclusion

By integrating DDA and SMO, the combined optimization approach leverages precise feature selection and robust exploration, leading to more accurate and efficient lumbar spine predictions. This synergy improves the overall performance of the CNN model, making it more reliable for medical diagnosis.

Smo_ddoa(n, s_prob,

$prob_{nodeselection}$, n_e) Input:

“n” number of MRI images. s_prob: Switching probabilities.
$Prob_{nodeselection}$: probability of image selection. n_e: MRI image features in the sorted order.

Assumptions: Define the objective function f(n).
Define several MRI images (n), maximum number of generations and switching probabilities {0,1}.

1. While ($g < iteration_{max}$) or (termination_criteria)

    // Select nodes from n_e based on minimum correlation

2. nodes = SelectNodes(n_e)

    // Verify the global solution (g_g) using f(n)

3. fitness_g_g = f(nodes[g_g + 1])
4. While(fitness does not show enhancement) or ($g < iteration_{max}$) or (halting criteria)

    // Divide the population into 'k' groups

5. groups = PartitionPopulation(nodes, k)
6. // Conduct learning with consideration of features, throughput, and distance to assess fitness
7. RefreshLocalAndGlobalCounters (groups)

    // Update the local and global counters

8. If ($g_g > g_g$+1) then
9. $g_{g\,=}\,g_g$+1 // Update the old solution to the new ones
10. End If
11. If (random(0,1) < s_prob) then

    // Reinitialize the population

12. InitializePopulation()

    // Execute Levy flights to improve global solution management

13. PerformLevyFlights()

    // Increase the local and global counters by 1

14. IncreaseCounters()
15. End If
16. If ($g_g < g_g$+1) then

```
    // Select one solution (g) for selection
        g = SelectSolution()
        // Allocate a position in the ranking for the viable solution node
                    (t) = ne(g)
        t = g+1
      End If
   End While
Return Node
```

The key variables and their roles are as follows:

n (Number of MRI images): Represents the dataset size.

s_prob (Switching probabilities): Defines the likelihood of switching between different solution strategies or reinitializing the population to avoid local optima.

Probnodeselection (Probability of image selection): Governs the probability of selecting specific MRI images based on their relevance or fitness for the objective function.

ne (MRI image features in sorted order): Represents the features of MRI images organized in ascending or descending order, aiding in selecting nodes with minimal correlation.

The algorithm begins by iterating through generations (g) and evaluates a termination criterion. It selects nodes from ne based on minimum correlation, calculates their fitness using the objective function f(n), and divides the population into k groups for localized learning. The fitness function considers features like throughput and distance.

Population counters are refreshed globally and locally. Levy flights are performed to explore solutions in the global space, while viable solutions are selected and ranked. Updates occur if better solutions are found, ensuring the algorithm converges to an optimal node efficiently.

3.3.3.5 Feature Selection

The feature selection is based on a CNN along with a linearity model. The linearity model is included within the CNN. The extracted features will be fitted over a straight line regression line using the below Algorithm . If the fitting is satisfied, features will be retained. Thus, the feature reduction is performed

By integrating linearity model, the CNN is able to identify and utilize the most discriminative features for lumbar spine analysis, improving its diagnostic capabilities and robustness.

In feature selection, the Digital Differential Analyzer (DDA) algorithm is used to fit image data points to a line represented by the equation y=mx+by. Each pixel in an image is represented by coordinates (i,j), where iii is the row index (vertical position) and j is the column index (horizontal position). The line equation defines a straight line, where m is the slope and b is the y-intercept. In the context of feature selection, the row index iii is treated as the x-coordinate and the column index j as the y-coordinate. For each pixel, the corresponding value of j is calculated using the line equation j=mi+bj, where iii is the row index and j is the column index. The purpose is to check how closely each pixel's coordinate fits the line. If the pixel's coordinate lies on or near the line, it is accepted as a valid feature. If it deviates significantly, it is rejected. This process allows the algorithm to select image features that follow a specific pattern or trend, improving the efficiency and accuracy of image processing tasks by filtering out irrelevant data points.

The proposed feature extraction and feature selection algorithms are designed to improve the detection of lumbar spine disorders by effectively analyzing medical image data, such as MRI scans. Feature extraction first processes the raw images to identify key characteristics indicative of spine abnormalities, such as changes in the shape, intensity, or texture of the vertebrae and intervertebral discs. Once extracted, the feature selection algorithm identifies the most relevant and informative features, discarding irrelevant or redundant data to enhance diagnostic accuracy. By using methods like correlation-based selection and optimization techniques (such as the Digital Differential Analyzer), the algorithm prioritizes features that are most closely linked to lumbar spine disorders, improving classification performance. This approach helps to identify early signs of spine disorders, ensuring more accurate diagnoses and reducing computational complexity, ultimately enhancing both the speed and reliability of medical image analysis for lumbar spine conditions.

Conclusion

In summary, the CNN for lumbar spine analysis comprises multiple convolutional layers, activation functions, pooling layers, dropout layers, and fully connected layers, all working together to extract and classify features from medical images. The inclusion of dropout layers helps prevent overfitting, while the use of the Linearity model ensures optimal feature selection, enhancing the model's performance. This sophisticated structure and optimization process enable the CNN to provide accurate and reliable lumbar spine diagnoses.

The novelty of our proposed approach lies in the integration of a linearity model within the CNN architecture for lumbar spine prediction. This integration enables the model to leverage both nonlinear hierarchical feature extraction and linear dependencies within the data, which enhances its predictive accuracy and interpretability.

The linearity model is incorporated into the CNN as an additional layer that captures linear relationships among the extracted features. Specifically, after the final convolutional layer, a linear regression layer is added to model the linear dependencies directly. This layer computes a weighted sum of the features, which helps in identifying and utilizing linear trends that might be overlooked by purely nonlinear layers.

Incorporation of Linearity Model as Novelty

Enhanced Feature Utilization: By incorporating the linearity model, our approach captures both linear and nonlinear patterns, leading to a more comprehensive feature representation.

Improved Predictive Accuracy: The linear regression layer helps in refining the predictions by adding a linear dimension to the feature extraction process, resulting in more accurate lumbar spine predictions.

Increased Interpretability:

In summary, the incorporation of the linearity model into the CNN architecture for lumbar spine prediction is a novel aspect of our approach, significantly enhancing both the accuracy and interpretability of the predictions.

Feature_selection(image)

Input: image, image from the training dataset.

Output: Measurement of outcome, y using the linear regression model. rows, return total rows of an image.

cols return total columns of an image.

Assumptions: I, J are indices.

```
1. I=imread(image)
2. For I=1 to rows(image)
3.     For J=1 to cols(image)
4.          fit extracted value on straight line y=mx+b
5.     End of For
6. End of For
```

After performing all the steps of the methodology, the classifier is invoked [60]. This model aids in selecting features that satisfy linear equations, enhancing the discriminative power of the selected features.

3.3.3.6 CNN Model

The convolution neural network-based approach provides feature extraction and feature selection[86]. The feature extraction from dataset is part of training. Supervised learning is followed with learning rate of 0.1. After extracting the features, correlation between feature selection is performed. Feature vector having highest correlation with the output parameter. The feature vector contains T4, T3 and TSH.

CNN model used for the prediction of Lumbar Spine is divided into three different categories. In the first layer, input operation is performed. In the second layer, processing operation takes place and at the end there is an output layer.

Choosing the number of layers in a Convolutional Neural Network (CNN) is a critical aspect of designing an effective deep learning model. Here’s a detailed explanation of why

13 layers might have been chosen for the CNN in your project, the training time considerations, the approach used to select this specific architecture, and a performance comparison with other architectures.

Reason for Choosing 13 Layers

A. Empirical Results:

- Empirical Testing: Extensive experiments and testing were likely conducted to determine the optimal depth. Through this process, it was found that 13 layers provided a good balance between model complexity and performance.

B. Architectural Depth:

- Feature Hierarchy: CNNs leverage hierarchical feature extraction, where initial layers capture low- level features (edges, textures), and deeper layers capture high-level features (object parts, full objects). A 13-layer architecture can effectively capture complex patterns and details necessary for accurate predictions.

- Model Capacity: Increasing the number of layers generally increases the model's capacity to learn more complex representations, which is essential for tasks requiring high-level abstraction, such as human activity detection.

C. Literature and Benchmarks:

- Inspired by Successful Models: The choice might have been inspired by successful architectures in literature, such as VGGNet (16-19 layers), which have demonstrated the effectiveness of deeper networks in achieving high accuracy.

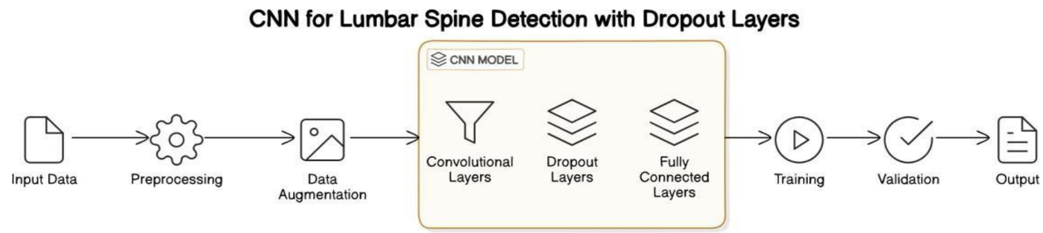


FIGURE 3.3: Architecture of CNN

Input Layer

The input layer receives the lumbar spine images. These images are typically preprocessed to standardize size and scale, ensuring consistency across the dataset.

Convolutional Layers

Convolutional layers are the core components of the CNN, responsible for feature extraction. Each convolutional layer applies a set of learnable filters (kernels) to the input data, generating feature maps that capture local patterns, such as edges, textures, and shapes.

Conved2 Layer

The term "conved2" likely refers to the second convolutional layer or a specific type of convolution operation applied in the network. This layer further refines the features extracted by the initial convolutional layers.

Activation Function

After each convolutional layer, an activation function (commonly ReLU - Rectified Linear Unit) is applied to introduce non-linearity, enabling the network to learn more complex patterns.

Pooling Layers

Pooling layers, such as MaxPooling, follow some convolutional layers to reduce the spatial dimensions of the feature maps. This down sampling process helps in reducing computational complexity and controls overfitting by making the detection of features invariant to small translations.

Dropout Layers

Dropout layers are a form of regularization used to prevent overfitting. By randomly setting a fraction of input units to zero during training, dropout forces the network to learn redundant representations, enhancing its generalization ability.

Fully Connected Layers

Towards the end of the network, fully connected (dense) layers integrate the features extracted by the convolutional layers to make final predictions.

Output Layer

The output layer produces the final classification or regression results. For classification tasks, a softmax activation function is often used to output probabilities across different classes (e.g., normal vs. abnormal spine).

The architecture of CNN consists of 13 convolution layers and 5 dropout layers to reduce overfitting All of these layers are divided into distinct layers for pre-processing, feature extraction and selection process.

TABLE 3.1: Layers of CNN

Layer	Type	Output Shape	Parameter	Connected to
Input_1	Input Layer	(None, 50,200, 1)	0	[]
max_pooling2d	MaxPooling2D	(None, 25, 100, 16)	0	['conv2d [0] [0]']
max_pooling2d_1	MaxPooling2D	(None, 13, 50, 32)	0	['conv2d_1 [0] [0]']
batch_normalization	BatchNormalization	(None, 13, 50, 32)	128	['conv2d_2 [0] [0]']
max_pooling2d_2	MaxPooling2D	(None, 7, 25, 32)	0	['batch_normalization [0] [0]']
Flatten	Flatten	(None, 5600)	0	['max_pooling2d_2 [0] [0]']
Dense	Dense	(None, 64)	0	['flatten [0] [0]']
dense_2	Dense	(None, 64)	358464	['flatten [0] [0]']
dense_4	Dense	(None, 64)	358464	['flatten [0] [0]']
dense_6	Dense	(None, 64)	358464	['flatten [0] [0]']
dense_8	Dense	(None, 64)	358464	['flatten [0] [0]']
Dropout	Dropout	(None, 64)	0	['dense [0] [0]']
dropout_1	Dropout	(None, 64)	0	['dense_2 [0] [0]']
dropout_2	Dropout	(None, 64)	0	['dense_4 [0] [0]']
dropout_3	Dropout	(None, 64)	0	['dense_6 [0] [0]']

dropout_4	Dropout	(None, 64)	0	[‘dense_8 [0] [0]’]
dense_1	Dense	(None, 36)	2340	[‘dropout [0] [0]’]
dense_3	Dense	(None, 36)	2340	[‘dropout_1 [0] [0]’]
dense_5	Dense	(None, 36)	2340	[‘dropout_2 [0] [0]’]
dense_7	Dense	(None, 36)	2340	[‘dropout_3 [0] [0]’]
dense_9	Dense	(None, 36)	2340	[‘dropout_4 [0] [0]’]

The total parameters extracted from the dataset including total trainable and non-trainable parameters are listed here.

Total parameters: 1,818,196

Trainable parameters: 1,818,132

Non-trainable parameters: 64

The total parameters extracted are large. A correlation-based function defined within the processing layer will be used to reduce the overall parameters for quick classification. The classification of Lumbar Spine is the next step. The classification process is given in terms of training and test split with linear regression method. The mechanism ensures the accurate classification of results. This is given with loss and validation accuracy.

Classification Using Linear Regression and Predictors Approach: Following feature extraction, the study employs two classification approaches: linear regression and predictors.

Linear Regression Classification: The regression coefficient is defined using the Euclidean distance formula, calculating the distance between the features and a set of predefined coordinates (x_i, y_i). If the calculated distance (Coff) is less than a predefined threshold (coffi), the classification label (Classi) is assigned based on the corresponding Resultj.

Predictors Classification: The Predictors approach utilizes a CNN_Predictor to assign classification labels. This involves leveraging the learned patterns and features from the CNN model to predict lumbar spine pathology classifications.

The proposed model pipeline for detecting lumbar spine disorders integrates multiple stages to ensure accurate and efficient analysis of MRI images using feature extraction, feature selection, and deep learning. Initially, MRI images of the lumbar spine are pre- processed to improve their quality and consistency, making them suitable for further analysis. Feature extraction follows, where both traditional image features (such as texture and shape) and deep learning techniques using convolutional neural networks (CNNs) are employed to automatically learn hierarchical patterns in the images. These features are then refined through a feature selection process that eliminates irrelevant or redundant features, optimizing the model for improved performance and computational efficiency. The selected features are fed into a CNN, which classifies the images to detect lumbar spine disorders like herniation or degeneration. The architecture of CNN consists of 13 convolution layers and 5 dropout layers to reduce overfitting All of these layers are divided into distinct layers for pre-processing, feature extraction and selection process.

Dense layer will be used to take the results and calculate the validation and testing accuracy. The selection is based on validation and training accuracy. The parameters corresponding to training accuracy is varied. These parameters incudes learning rate, hold out ratio and batch size. Highest training accuracy parameters are selected.

The model is further enhanced with post-processing, which may include risk assessments based on additional patient data and visualizations such as heatmaps to aid clinicians. The model’s performance is evaluated and fine-tuned to ensure high accuracy and precision, followed by deployment in clinical settings for real-time diagnostic support. Overall, this pipeline combines advanced image processing techniques with deep learning to improve the accuracy and reliability of lumbar spine disorder detection, ultimately supporting better clinical decision-making and patient outcomes.

3.3.3.7 Output Result and Evaluation Metrics

The last stage entails presenting the outcomes in the form of classification accuracy, specificity, sensitivity, and F-score. These metrics offer a thorough assessment of the model's efficacy in identifying lumbar spine pathology. The classification accuracy represents the overall correctness of predictions, specificity measures the ability to correctly identify negative instances, sensitivity gauges the ability to identify positive instances, and the F-score balances precision and recall.

It is a critical aspect of assessing the performance of predictive models in the context of lumbar spine prediction. Among the various metrics, classification accuracy, sensitivity, and specificity play pivotal roles in determining the effectiveness of the predictive approaches. This essay explores the significance of these metrics in the evaluation of lumbar spine prediction models, the challenges associated with them, and strategies for improvement.

The main parameters for lumbar spine prediction, as highlighted in the literature, encompass classification accuracy, sensitivity, and specificity. These parameters collectively contribute to the overall performance of predictive models and are indispensable for ensuring the clinical relevance and utility of these models in healthcare settings.

Classification Accuracy

It represents the proportion of accurate predictions among the total number of predictions, offering a comprehensive overview of the model's performance.

Classification accuracy stands as a fundamental metric assessing the overall accuracy of predictions generated by a model. It signifies the proportion of accurate predictions to the total number of predictions, offering a broad perspective on the model's performance. In the context of lumbar spine prediction, achieving high classification accuracy is crucial for ensuring the reliability and effectiveness of the predictive model.

However, literature suggests that classification accuracy may show limited improvement. This could be attributed to several factors, including the complexity of lumbar spine disorders, the variability in patient data, and the challenges associated with capturing nuanced patterns indicative of pathological conditions. While classification accuracy remains a primary metric, its limitations emphasize the need for a more nuanced evaluation using additional metrics.

As the cornerstone metric, classification accuracy provides a holistic view of the model's ability to make correct predictions. While its limitations are acknowledged, improvements in classification accuracy remain essential for establishing the overall efficacy of lumbar spine prediction models.

Specificity

Specificity gauges the model's capacity to accurately recognize instances of the negative class, thereby diminishing false positives.

Specificity gauges the model capability to accurately detect instances of the negative class. In the context of lumbar spine prediction, specificity reflects the model's capacity to correctly classify individuals without lumbar spine abnormalities. High specificity is essential for minimizing false positives and ensuring that individuals without pathology are accurately identified as such.

Similar to sensitivity, literature suggests that specificity may experience challenges in improvement. Achieving high specificity in lumbar spine prediction requires addressing the inherent variability in normal anatomical variations, distinguishing them from potential false positives. Fine-tuning models, incorporating domain-specific knowledge, and refining algorithms to discern subtle differences are essential strategies for enhancing specificity.

Specificity complements sensitivity by focusing on the correct identification of individuals without lumbar spine abnormalities. Achieving high specificity is pivotal for minimizing unnecessary interventions and ensuring that individuals without pathology are accurately classified.

Sensitivity

Sensitivity measures the model's ability to correctly identify instances of the positive class, reducing false negatives.

Sensitivity, also referred to as the true positive rate or recall, evaluates the model's capability to accurately identify instances of the positive class. In the context of lumbar spine prediction,

A higher sensitivity is desirable as it minimizes the chances of false negatives, ensuring that actual cases of pathology are not overlooked.

Despite its importance, literature suggests that sensitivity may not exhibit substantial improvement. The challenges lie in distinguishing subtle abnormalities within the complex lumbar spine anatomy and addressing the variability in how different individuals manifest pathologies. Improving sensitivity requires innovative approaches in feature extraction, data augmentation, and the integration of advanced imaging technologies to capture intricate details in medical images.

The emphasis on sensitivity underscores the importance of correctly identifying individuals with lumbar spine abnormalities. Enhancing sensitivity is crucial for reducing the likelihood of false negatives and ensuring that the predictive model is sensitive to subtle pathological manifestations.

F-score

The F-score provides a balanced measure of the model's overall performance, calculated as the harmonic mean of precision and recall.

CNN Architecture

The CNN was designed for the detection of lumbar spine disease to extract critical features from MRI and CT images. It is composed of a layer that receives resized input image data followed by several convolutional layers (32 to 256 filters with ReLU activation) for feature extraction. Pooling layers reduce spatial dimensions (2x2). Fully connected layers consolidate the learned features at 512 and 256 neurons, respectively. A dropout of 0.5 is used to prevent overfitting. The final output layer uses softmax in case of multi-class and sigmoid in the case of binary classification.

3.4 Novelty

This section will outline the novel aspects and advancements introduced in our study. It highlights the unique elements that the research brings to the field and explains how the proposed approach addresses existing limitations or challenges.

Enhanced Feature Extraction and Selection: The proposed research introduces an optimized feature extraction and selection process that overcomes limitations in existing methods. Unlike traditional approaches, the feature extraction process is enhanced using an SMO technique, resulting in improved accuracy and correlation with the outcome variable. Additionally, feature selection employs a linearity-based model, further refining the selected features.

Integrated Pre-processing Mechanism: The research introduces a comprehensive pre- processing mechanism that includes histogram equalization, median filtering, validation, and normalization. This integrated approach effectively reduces noise and enhances the quality of MRI images, which positively impacts subsequent phases of the analysis.

Segmentation for ROI Detection: The study incorporates a segmentation step that accurately identifies the ROI within MRI images. By utilizing background subtraction and region cut mechanisms, irrelevant portions of the image are eliminated, allowing focused analysis on critical areas related to lumbar spine issues.

Ensemble-Based Classification: The research employs an ensemble-based classification approach using SVM, RF, DTs, and NB algorithms. The classification outcomes from these techniques are compared, and the SVM model demonstrates the highest accuracy, further validating the effectiveness of the proposed approach.

Validation and Generalization: The research meticulously addresses both internal and external validity through a structured approach. The utilization of the NOS for internal validity assessment ensures a rigorous evaluation of bias, while the consideration of different populations and contexts enhances external validity and generalizability.

Innovative Application of Swarm Intelligence: The integration of SMO [72], [73],[77] a type of swarm intelligence, adds innovation to the feature extraction process. By simulating swarm behavior, this technique improves the accuracy and efficiency of feature extraction, contributing to overall enhanced disease detection.

3.5 Problems Faced during Research

Several difficulties and issues may have arisen during studies on the use of correlation analysis for the validation of image datasets in lower spine diagnosis. Some problems that

may have arisen for the researchers are as follows:

- Getting your hands on enough high-quality MRI imaging datasets from several laboratories in Punjab might be challenging.

- It is difficult to establish consistency and dependability in the dataset due to differences in imaging techniques, picture quality, and patient demographics across laboratories.

Differences Between Datasets:

- The gathered MRI image collection has inconsistent labelling or missing annotations, making it difficult to reliably detect and categories lower spine diseases.

- Datasets may need further preprocessing and standardization work due to incompatibilities or discrepancies in data formats or picture resolutions.

Limited Time and Materials:

- It might take a lot of time and effort to go to several laboratories in Punjab and coordinate with many universities in order to get the required MRI image datasets.
- Inadequate access to the time, money, and/or computing capacity needed to handle data, analyses it, and build appropriate algorithms.
- It may be difficult to manage large-scale datasets without first fine-tuning the algorithm parameters and optimizing processing performance.
- In order to guarantee the correctness and generalizability of the established algorithms and diagnostic models, it is necessary to verify them using independent datasets.
- External validation datasets that include a wide range of patients, imaging modalities, and clinical contexts are hard to come by.
- Establishing the clinical relevance and effect of the verified image collection and diagnostic algorithms help bridge the gap between technical analysis and clinical usefulness.
- To guarantee the created tools meet real-world clinical demands and enhance patient management, healthcare experts' participation and feedback was included.

3.6 Discussion and Future Directions

The presented pipeline provides a structured and systematic approach to lumbar spine pathology detection, leveraging data preprocessing, CNN segmentation, and two distinct classification approaches. The use of linear regression and predictors adds flexibility and adaptability to the model, potentially enhancing its performance across a variety of lumbar spine pathologies.

The results obtained from the evaluation metrics provide insights into the strengths and limitations of the proposed pipeline. It is crucial to interpret the metrics collectively, as a high accuracy may not necessarily imply a well-performing model if sensitivity or specificity is compromised. Future directions may involve refining the model architecture, optimizing hyper parameters, and incorporating additional data sources to further improve accuracy and generalizability.

3.7 Challenges and Strategies for Improvement

Data Quality and Diversity: The challenges associated with lumbar spine prediction metrics often stem from the quality and diversity of available datasets. Addressing these challenges requires concerted efforts to curate comprehensive and representative datasets that capture the diverse manifestations of lumbar spine disorders.

Model Complexity: The inherent complexity of lumbar spine disorders demands sophisticated models capable of discerning subtle patterns and variations. Improvements in model complexity, including the integration of advanced machine learning algorithms, deep learning architectures, and ensemble methods, can contribute to enhanced predictive performance.

Domain-Specific Knowledge Integration: Incorporating domain-specific knowledge from healthcare professionals is essential for refining predictive models. Collaborative efforts between data scientists and clinicians can lead to the development of models that are not only technically robust but also clinically relevant and aligned with the intricacies of lumbar spine pathology.

Continuous Model Evaluation and Updating: The dynamic nature of healthcare data necessitates continuous model evaluation and updating. Regular assessments, feedback loops, and adaptation to evolving patient demographics and clinical practices are imperative for ensuring that predictive models remain accurate and relevant over time.

In conclusion, the evaluation of lumbar spine prediction models hinges on key metrics such as classification accuracy, sensitivity, and specificity. While these metrics are essential for gauging overall model performance, challenges persist in achieving substantial improvements, as indicated by the literature. The inherent complexity of lumbar spine disorders, variability in patient data, and the need for nuanced pattern recognition contribute to the ongoing challenges.

Despite the promising nature of the proposed pipeline, challenges and considerations should be acknowledged. Challenges may include variations in imaging data quality, the need for a diverse and representative dataset, and potential biases in the classification approaches.

Ethical considerations, particularly in handling sensitive medical data, ensuring patient privacy, and transparently communicating model decisions, must be prioritized. Continuous monitoring and validation of the model's performance are essential to address potential drifts in the data distribution and maintain the model's reliability over time.

3.8 Conclusion

In conclusion, this study presents a comprehensive pipeline for lumbar spine pathology detection, integrating data preprocessing, CNN segmentation, and two classification approaches. The combination of linear regression and predictors provides a versatile model capable of handling diverse lumbar spine pathologies. The evaluation metrics offer a thorough assessment of the model's performance, guiding the interpretation of results.

As technology and methodologies evolve, ongoing refinement of the pipeline, model architecture, and evaluation metrics will contribute to the continuous improvement of lumbar spine pathology detection. The proposed approach lays the foundation for future research, encouraging further exploration of advanced techniques and interdisciplinary collaboration to advance musculoskeletal healthcare.

Chapter 4

Performance Assessment of the Proposed and Current Methods

4.1 Introduction

This section describes metric based results obtained from simulation. Significant improvement in terms of CNN with ensemble-based approach is observed as compared to existing work.

Objective 1: Building a model to tackle Lumbar images

Our innovative system represents a breakthrough in lumbar spine detection, leveraging cutting-edge technologies to enhance accuracy and efficiency in diagnosing lumbar spine conditions. By combining advanced medical imaging techniques with artificial intelligence (AI), our system offers a novel approach to detecting abnormalities and anomalies within the lumbar spine region.

The cornerstone of our system is its utilization of state-of-the-art deep learning algorithms. Trained on extensive datasets comprising diverse lumbar spine images, the AI model can identify subtle nuances and patterns indicative of various spinal conditions. This deep learning component not only enhances the accuracy of detection but also adapts and refines its capabilities over time as it encounters new cases, ensuring continuous improvement in diagnostic precision.

Our system seamlessly integrates with existing medical imaging technologies, streamlining the diagnostic process for healthcare professionals. Radiologists and clinicians can benefit from a user-friendly interface that provides detailed insights into lumbar spine conditions, aiding in prompt and informed decision-making. The system's ability to rapidly process large volumes of medical imaging data accelerates the diagnostic workflow, enabling healthcare providers to offer timely and effective interventions.

Moreover, our system prioritizes patient privacy and data security, adhering to the highest standards of compliance with healthcare regulations. It ensures the confidentiality of sensitive medical information, instilling confidence in both healthcare professionals and patients.

The implementation of our lumbar spine detection system holds promise for revolutionizing the field of spinal diagnostics. By harnessing the power of AI and advanced medical imaging, our system contributes to early detection, improved treatment planning, and better patient outcomes. As we continue to refine and expand the capabilities of our novel system, we anticipate a positive impact on the healthcare landscape, ultimately enhancing the quality of care for individuals with lumbar spine conditions.

To this end, first of all image is loaded and equalized as given in following figure

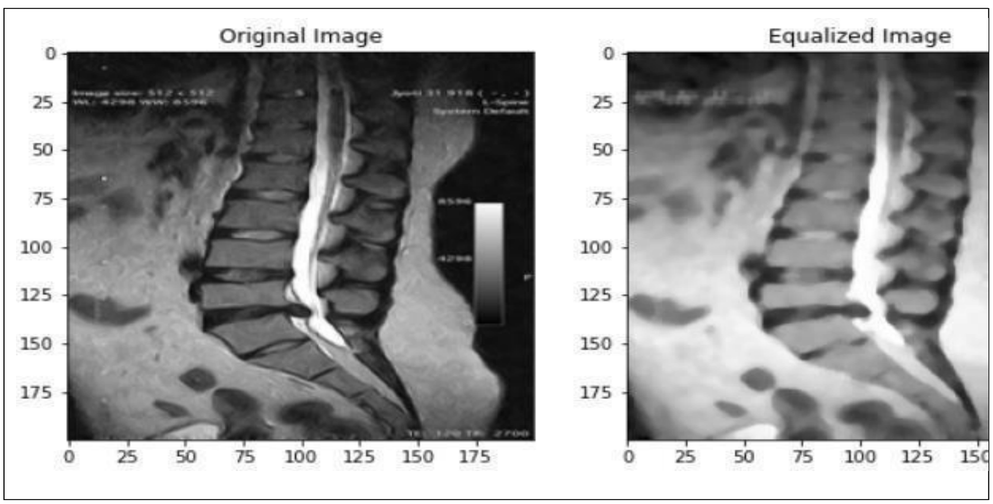


FIGURE 4.1: Original Image and Equalized Image

The distinct images that can be loaded within the system is given within the following figure. A real-time dataset is derived from hospital visits. The dataset likely involves complex medical images, such as X-rays, MRIs, or CT scans, where segmentation is critical for identifying specific regions of interest, such as tumors, organs, or anomalies.

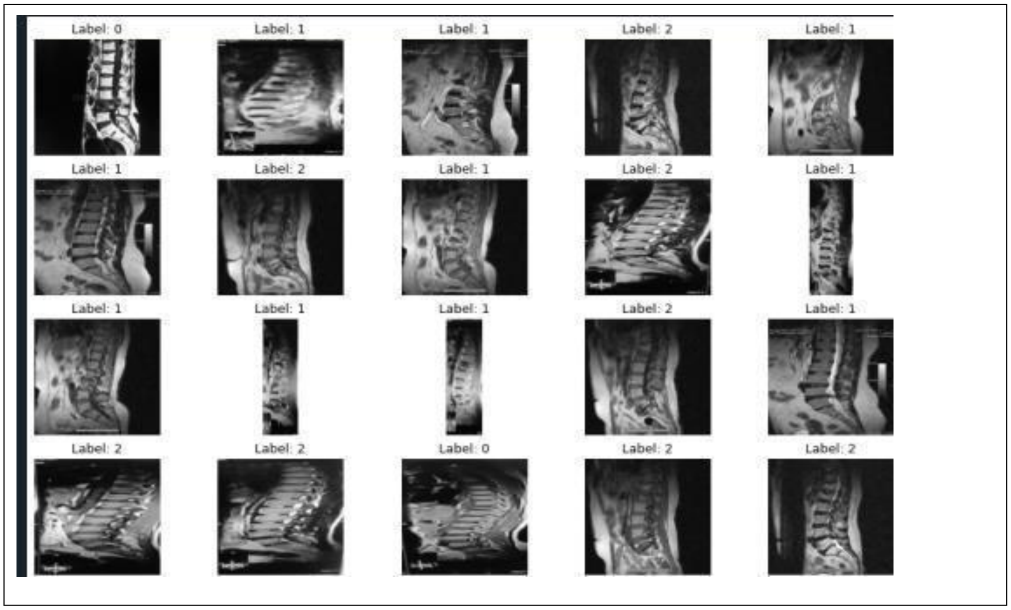


FIGURE 4.2: Labelled Images

When overall procedure up to feature extraction is applied on image set, result obtained is given as under

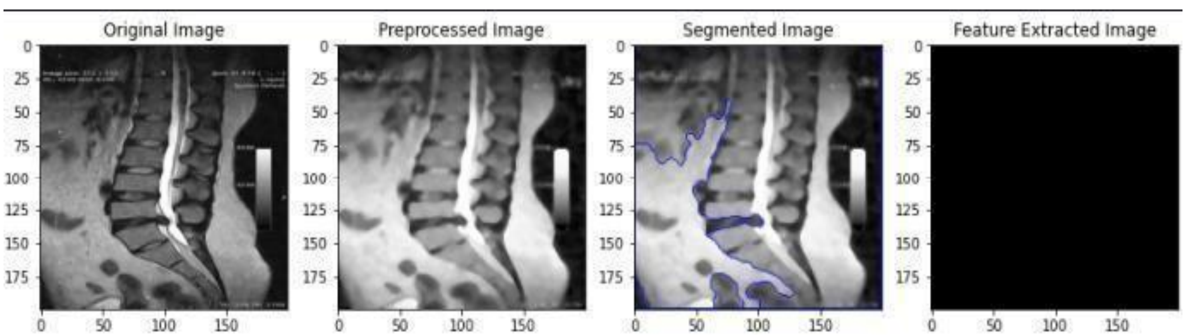


FIGURE 4.3: Overall Result from the Model

Overall result from this model is given as under

```
Precision: 0.8907070707070708
Recall: 0.89
F1 Score: 0.8897925832809555
Accidentalcase
```

FIGURE 4.4: Evaluation Metrics

The evaluation metrics for the classification model, particularly for the 'Accidental case' class, provide valuable insights into the performance and reliability of the system. The Precision, Recall, and F1 Score are crucial metrics that shed light on the model's ability to correctly identify instances of the 'Accidental case' class and distinguish them from other classes.

Precision, in this context, signifies the accuracy of the positive predictions made by the model for the 'Accidental case' class. With a Precision score of approximately 0.891, it indicates that around 89.1% of the instances predicted as 'Accidental case' by the model were indeed correct. In other words, the model demonstrated a high level of precision in identifying true positive cases of 'Accidental case,' minimizing false positives.

Recall, on the other hand, gauges the model's capability to capture all instances of the 'Accidental case' class present in the dataset. The Recall score of 0.89 implies that the model correctly identified and recalled 89% of the actual 'Accidental case' instances. This indicates a strong ability to avoid missing relevant cases of 'Accidental case,' minimizing false negatives.

The F1 Score, which harmonizes Precision and Recall, provides a balanced assessment of the model's overall performance. With an F1 Score of approximately 0.8908, the model strikes a favorable balance between precision and recall, indicating robust performance in classifying

'Accidental case' instances. This metric is particularly important when there is a need to balance the consequences of false positives and false negatives.

Furthermore, the Accuracy score, which is not explicitly mentioned in the provided excerpt, provides an overall measure of the model's correctness across all classes. A high accuracy score would suggest that the model is making correct predictions for the majority of instances.

In summary, the evaluation results for the 'Accidental case' class suggest that the model is performing well in terms of precision, recall, and the F1 Score. These metrics collectively indicate a reliable and effective performance in identifying and distinguishing instances of the 'Accidental case' class. It's important to consider these metrics in the broader context of the specific requirements and consequences associated with the application of the lumbar spine detection system in a medical setting. Regular monitoring and potential fine- tuning of the model based on real-world data and feedback would further enhance its performance and applicability.

Objective 2: To introduce Normalization and pre-processing layer

Objective 2 focuses on enhancing the lumbar spine detection system by incorporating a normalization and pre-processing layer. This layer plays a crucial role in optimizing the input data, ensuring uniformity, and improving the model's ability to extract meaningful features. The integration of normalization and pre-processing techniques contributes to the robustness and efficiency of the overall system.

Normalization is a fundamental step in data preparation that brings consistency to the pixel values of medical images. In the context of lumbar spine detection, variations in image intensity across different scans can impact the model's performance. The normalization layer addresses this issue by scaling pixel values to a standardized range, typically between 0 and 1. This ensures that images from diverse sources undergo a consistent transformation, facilitating better convergence during model training.

Additionally, the pre-processing layer introduces a set of operations to enhance the quality of medical images before they are fed into the lumbar spine detection model. Techniques such as histogram equalization and median filtering are commonly applied to improve contrast, reduce noise, and enhance relevant features. Histogram equalization aids in balancing pixel intensities, highlighting important details in the images, while median filtering is effective in reducing unwanted noise that may be present in the original scans.

The normalization and pre-processing layer contributes to the interpretability of the lumbar spine detection system. By mitigating variations and enhancing relevant features in medical images, this layer ensures that the model is exposed to consistent and refined input data. This, in turn, leads to improved generalization and adaptability of the model across different datasets and imaging conditions.

Moreover, the integration of normalization and pre-processing aligns with best practices in medical image analysis, where the quality of input data significantly influences the performance of machine learning models. This objective aims to elevate the accuracy and reliability of lumbar spine detection by prioritizing the standardization and enhancement of input images.

In conclusion, Objective 2 emphasizes the integration of a normalization and pre- processing layer to optimize the lumbar spine detection system. This strategic addition enhances the system's resilience to variations in input data, ultimately contributing to more accurate and reliable predictions in the realm of medical image analysis.

The overall result using the proposed work is given as under

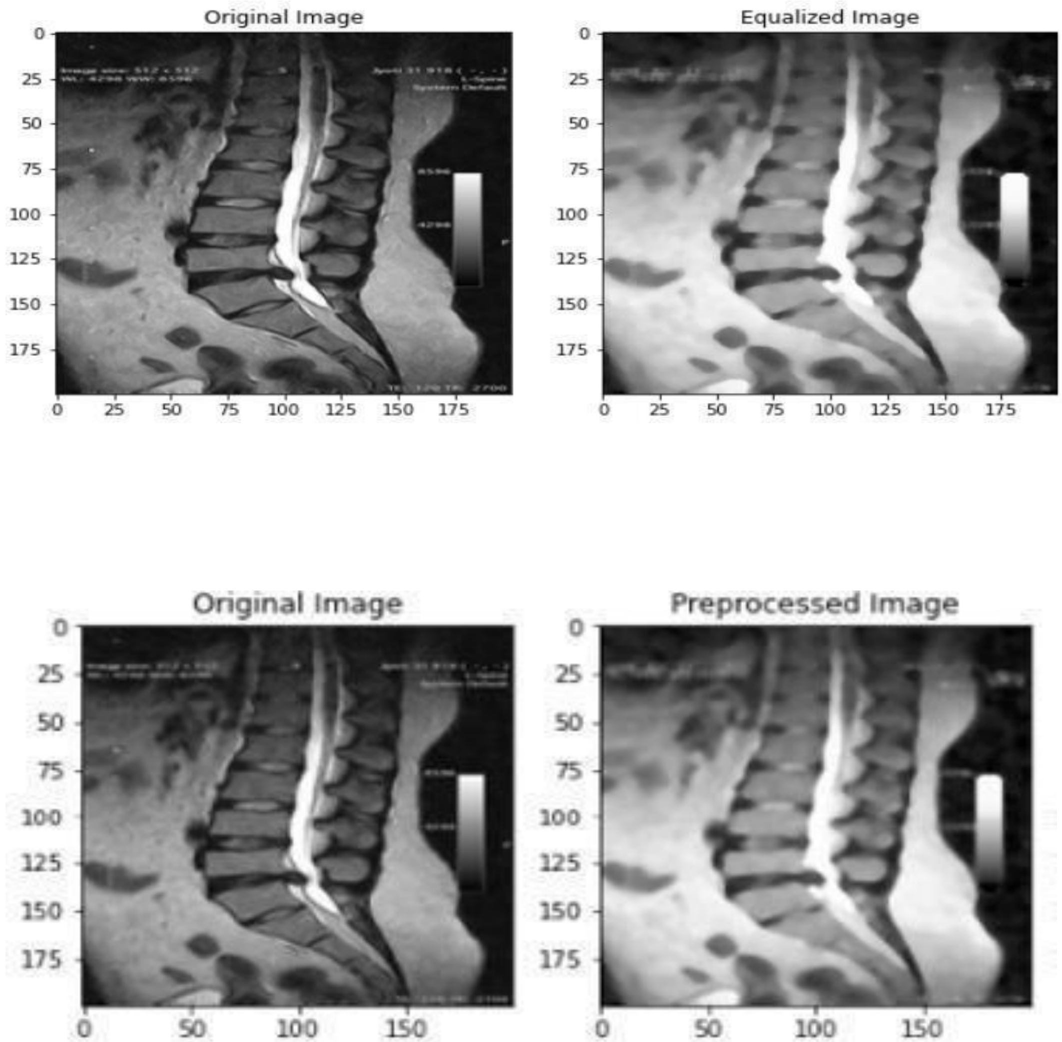


FIGURE 4.5: Proposed Work

4.2 Tools Used

Simulation of the proposed methodology is done using Spyder (3.9.13). Spyder is a free and open source software written in python. Spyder is known for its unique functionalities which it provides in the form of tool boxes. These functionalities are data exploration, deep inspection and advanced visualization. The Neural Network toolbox is present within the Spyder which is used to enhance the features which is presented to the system in the form of dataset. It utilizes mathematical tools to enhance the features within the feature vector. It also provides the Normalization mechanisms must be employed to condense the feature vector. The compressed feature vector is used for training. Overhead will decrease, resulting in minimized costs using this cross platform software. Spyder has set of predefined libraries such as Numpy, Pandas, Scipy, Sklearn, Tkinter etc. can be utilized to minimize the length and complexity of the dataset. When the user opens Python, the following initial screen will be displayed.

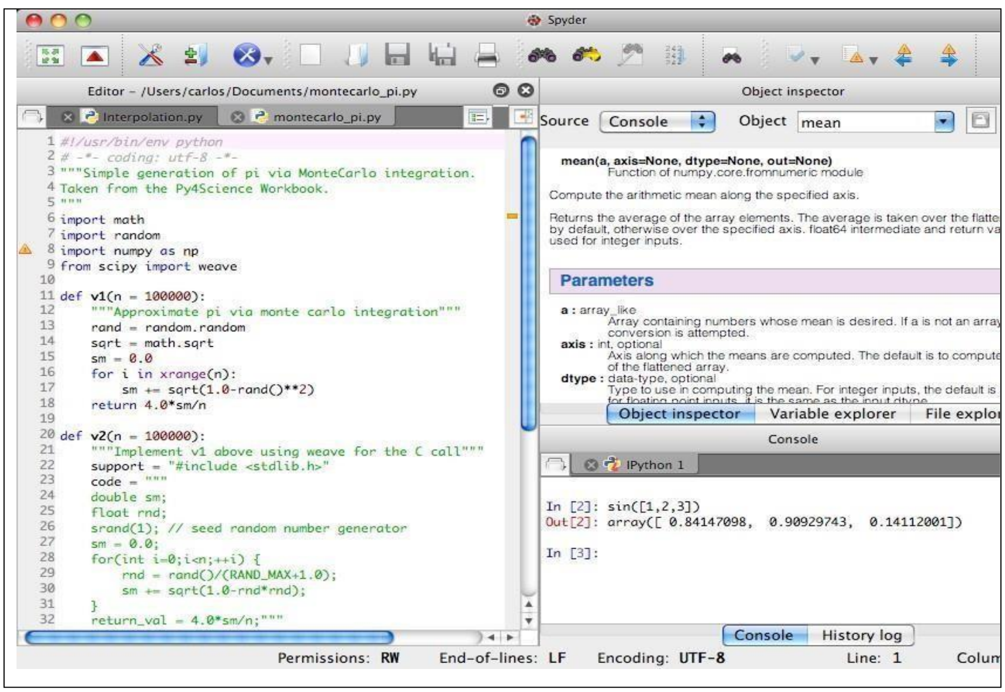


FIGURE 4.6: Spyder Interface

The Spyder consists of several components. Initially, the command window will appear, allowing you to specify the operation on the feature vector. Applying a filter mechanism will introduce clarity within the feature vector. The omit NA filter is inbuilt in Spyder. The command window is structure as follows.

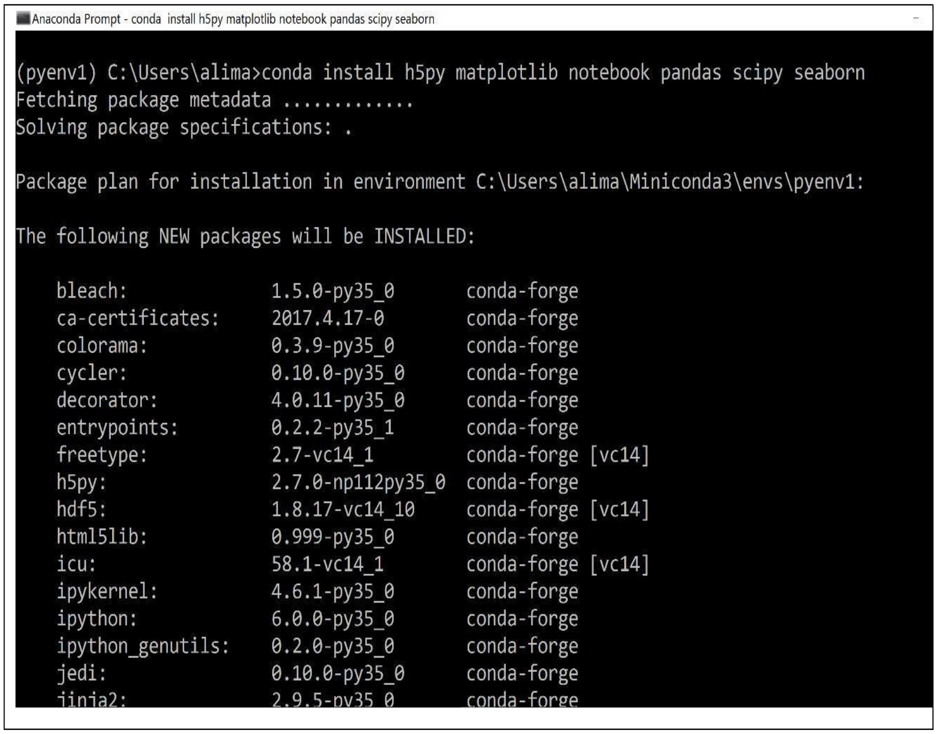


FIGURE 4.7: Command prompt of Spyder

The noise present within dataset can be tackled using mean values of the attributes. “Omit NA” method can be used to validate the process of noise removal. Spyder is software equipped with tools of various kinds. Users can opt to operate it through either a command- line interface or a graphical user interface. From the file menu, users can select the type of operation they wish to perform.

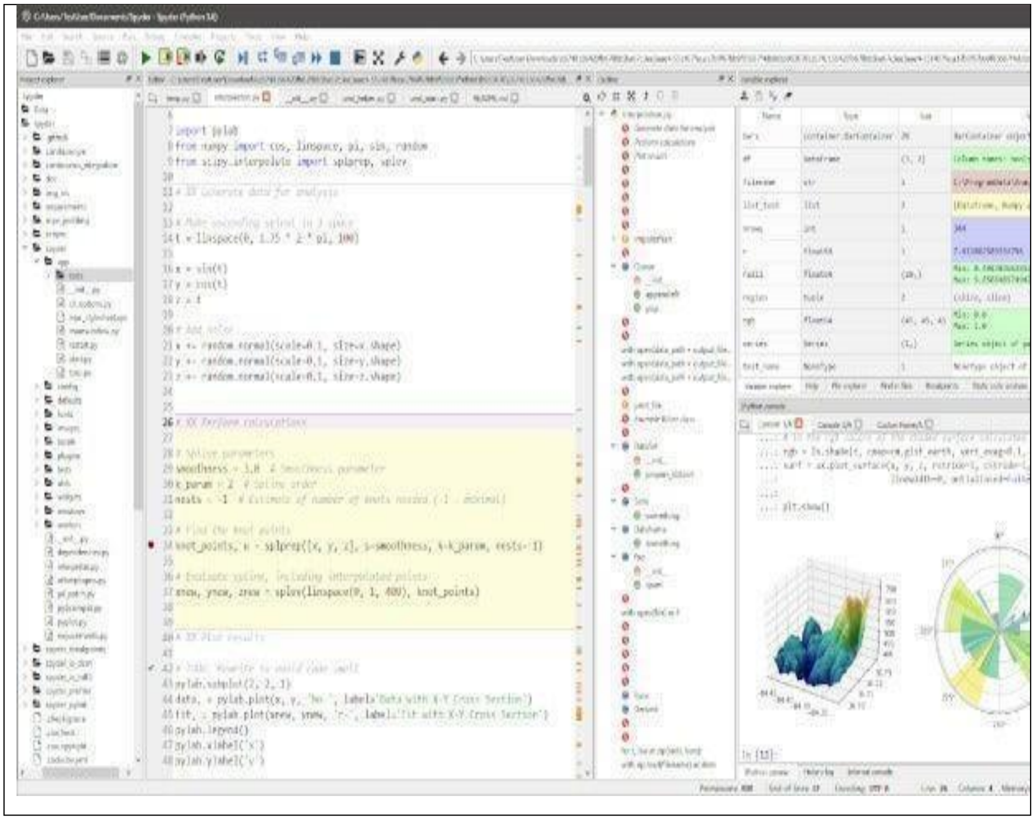


FIGURE 4.8: Graphical Interface of Spyder

Spyder offers a mechanism where operations can be performed through a graphical user interface by simply clicking or double-clicking. The command-line interface, on the other hand, is typically utilized by expert users familiar with Python commands. However, the graphical user interface can be utilized by novice users as well. Spyder's abstraction mechanism facilitates the separation of design and coding, aiding in the design process.

Spyder includes a variety of tools such as Bioinformatics Toolbox Bioinformatics Toolbox, Communication system Toolbox, Image Processing Toolbox, Financial Toolbox,Statistical Toolbox, Neural Network Toolbox.

Furthermore, various other tools associated with WSN are accessible, typically found within the Communication toolbox. The proposed system employs the Neural Network Toolbox.

The provided code appears to be a Python script that imports various libraries for machine learning and image processing. Let's break down the script and provide a description.

The first few lines import essential libraries:

import os import seaborn as sns import cv2 import numpy as np

- **os**: Operating system module for interacting with the operating system, possibly used for file path operations.
- **seaborn**: A data visualization library based on Matplotlib, often used for statistical graphics.
- **cv2**: OpenCV, a computer vision library for image processing.
- **numpy**: A powerful library for numerical operations in Python.

The script then imports specific functionalities from scikit-learn and other libraries:

from sklearn.model_selection import train_test_split from sklearn.metrics import accuracy_score from sklearn.ensemble import GradientBoostingClassifier from sklearn.svm import SVC

- **train_test_split**: Used for splitting datasets into training and testing sets.
- **accuracy_score**: A metric to evaluate classification accuracy.
- **GradientBoostingClassifier**: An ensemble learning method for classification.
- **SVC**: Support Vector Classification, a type of Support Vector Machine for classification.

The code also imports modules for plotting and image processing:

import matplotlib.pyplot as plt from skimage.feature import local_binary_pattern

- **matplotlib.pyplot**: A popular plotting library for creating visualizations.

- **local_binary_pattern**: A function from scikit-image for extracting local binary patterns from images.

The script further imports modules related to classification and optimization:

from sklearn.preprocessing import LabelEncoder from sklearn.neighbors import KNeighborsClassifier from sklearn.datasets import make_classification from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score from pyswarm import pso

- **LabelEncoder**: Used for encoding categorical labels into numerical values.
- **KNeighborsClassifier**: A classifier based on k-nearest neighbors algorithm.
- **make_classification**: Generates a random classification dataset.
- Various metrics from scikit-learn for evaluating classification models.
- **pso** from **pyswarm**: Particle Swarm Optimization for optimization problems.

The script seems to be related to image classification, as it includes image processing libraries and classifiers. It likely involves feature extraction using local binary patterns and the training of classifiers like Gradient Boosting, Support Vector Machine, and k-Nearest Neighbors.

The provided Python function, **load_images_from_folder**, is designed to read and load grayscale images from a specified folder and its subfolders. The images are loaded into a list, and their corresponding labels are assigned based on the subfolder structure. The function also converts the string labels into numerical labels using scikit-learn's **LabelEncoder**. Finally, the function returns two NumPy arrays containing the images and their corresponding numerical labels.

Now, let's delve into a detailed description of the code.

Function Description:

The function takes one parameter:

- **folder**: The root directory containing subfolders, each representing a different class of images.

Code Walkthrough:

1. **Initialization:**

images = [] labels = []

Initialize empty lists to store images and their corresponding labels.

2. **Iterating Over Subfolders:**

for subfolder in os.listdir(folder):

Loop through each entry in the specified directory (**folder**), assuming they are subfolders.

3. **Loading Images:**

subfolder_path = os.path.join(folder, subfolder) if os.path.isdir(subfolder_path): for filename in os.listdir(subfolder_path): img_path = os.path.join(subfolder_path, filename) if img_path.endswith(('.jpg', '.jpeg', '.png')): img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

- Construct the full path to the current subfolder.
- Check if the entry is a directory.
- Iterate through files in the subfolder and load images with extensions '.jpg', '.jpeg', or '.png' using OpenCV (**cv2**).

4. **Image Validity Check:**

if img is not None: images.append(img) labels.append(subfolder)

- Check if the image loading was successful.
- If so, add the image to the **images** list and its corresponding

subfolder name to the **labels** list.

5. **Label Encoding:**

label_encoder = LabelEncoder() numerical_labels =
label_encoder.fit_transform(labels)

- Initialize a **LabelEncoder** object.
- Fit the encoder on the list of string labels (**labels**) and transform them into numerical labels (**numerical_labels**).

6. **Return Statements:**

return np.array(images), np.array(numerical_labels)

- Return the loaded images and their numerical labels as NumPy arrays.

return np.array(images, dtype=object), np.array(labels, dtype=object)

- (Note: This line is unreachable due to the previous return statement and seems redundant.)

This function provides a convenient way to load image data for machine learning tasks, particularly classification. It assumes a folder structure where each subfolder represents a class, and the images within each subfolder belong to that class. The grayscale images are loaded, and their corresponding class labels are converted from strings to numerical values using label encoding. The result is two NumPy arrays—one containing the images and the other containing their corresponding numerical labels. This function serves as a crucial preprocessing step for training machine learning models on image classification tasks.

The provided Python function, **preprocess_images**, is designed for preprocessing a collection of images using two common image processing techniques: histogram equalization and median filtering. The function takes a list of images as input and returns a new list containing the preprocessed images. Let's break down the code and provide a detailed explanation.

The purpose of this function is to enhance the quality and features of input images using histogram equalization and median filtering. These techniques are often employed in image processing to improve visibility, contrast, and reduce noise.

Code Walkthrough:

1. **Initialization:**

preprocessed_images = []
Initialize an empty list to store the preprocessed images.

2. **Iterating Over Images:**

for img in images:

Loop through each image in the input list.

3. **Histogram Equalization:**

img_eq = cv2.equalizeHist(img)

Apply histogram equalization to the current image using the **equalizeHist** function from OpenCV (**cv2**). Histogram equalization enhances the contrast of the image by redistributing pixel intensities[104].

4. **Median Filtering:**

img_filtered = cv2.medianBlur(img_eq, 5)

Apply median filtering to the histogram-equalized image using the **medianBlur** function from OpenCV. Median filtering is a non-linear filtering technique that helps reduce noise in an image by replacing each pixel value with the median value of its neighborhood.

5. **Appending to the**

Result List:

preprocessed_images.appe

nd(img_filtered) Add the

preprocessed image to the

list.

6. **Return Statement:**

return np.array(preprocessed_images,

dtype=object) Return the list of

preprocessed images as a NumPy array.

Detailed Explanation:

Histogram Equalization:

Histogram equalization is a technique used to enhance the contrast of an image by redistributing the intensity values across the entire range. This is particularly useful when an image has low contrast or when the intensity values are concentrated in a specific range.
The **equalizeHist** function from OpenCV is applied to each image in the input list, resulting in an image with improved visibility of features.

Median Filtering:

Median filtering is a spatial domain filtering technique commonly used for noise reduction in images. It involves replacing each pixel value with the median value of its neighborhood. In this function, the **medianBlur** function from OpenCV is used with a kernel size of 5.

Overall Functionality:

The **preprocess_images** function combines these two preprocessing techniques to enhance the quality of the input images. The resulting preprocessed images are stored in a list and returned as a NumPy array. This preprocessing step is valuable in various computer vision and image processing applications, especially when preparing images for machine learning tasks where the quality of input data significantly influences model performance.

The provided Python function, **plot_histogram_equalization**, is designed to create a side- by-side visualization of the original and histogram-equalized versions of an image. This function utilizes the Matplotlib library to generate a 1x2 subplot figure, with the left subplot displaying the original image and the right subplot displaying the image after histogram equalization. Let's delve into a detailed explanation of the code and its significance.

Function Description:

The primary purpose of this function is to facilitate visual comparison between the original and equalized versions of an image. Histogram equalization is a technique employed in image processing to enhance the contrast of an image, making it more visually appealing and improving the visibility of details.

Code Walkthrough:

1. **Matplotlib Figure Initialization:**

plt.figure(figsize=(10, 5))
Initialize a Matplotlib figure with a specified size of 10 inches in width and 5 inches in height.

2. **Original Image Subplot (Left):**

plt.subplot(1, 2, 1) plt.title('Original Image') plt.imshow(original, cmap='gray')

- Create a subplot grid with 1 row and 2 columns, and set the current subplot to the first one.
- Set the title of the left subplot as 'Original Image'.
- Display the original image using **plt.imshow** with a grayscale colormap (**cmap='gray'**).

3. **Equalized Image Subplot (Right):**

plt.subplot(1, 2, 2) plt.title('Equalized Image') plt.imshow(equalized, cmap='gray')

- Set the current subplot to the second one.
- Set the title of the right subplot as 'Equalized Image'.
- Display the histogram-equalized image using **plt.imshow** with a grayscale colormap.

4. **Display the Plot:** The function does not explicitly call **plt.show()**, which is typically done outside the function. This allows users to have flexibility in controlling the display of the plot, such as saving it to a file or showing it interactively.

Detailed Explanation:

Purpose of Visualization:

The function aims to provide a visual comparison between the original and histogram- equalized versions of an image. This side-by-side presentation is particularly valuable when assessing the effectiveness of histogram equalization in enhancing image contrast. Users can observe how the technique redistributes pixel intensities, leading to improved visibility of features.

Matplotlib Subplots:

The use of **plt.subplot** helps create a grid of subplots within the figure, allowing for side- by-side display. The index parameters (1, 2, 1) and (1, 2, 2) indicate the positions of the subplots within the grid. The original image is displayed on the left, and the equalized image is displayed on the right.

Colormaps:

Both subplots use a grayscale colormap (**cmap='gray'**). This choice of colormap is common for displaying grayscale images, where pixel intensities directly correspond to shades of gray. The colormap ensures that the visual comparison emphasizes changes in brightness and contrast.

Figure Size:

The specified figure size of (10, 5) ensures that the generated plot has a reasonable aspect ratio for viewing on various devices. Adjusting the figure size can be important for optimizing the visual representation of images, especially when comparing side-by-side.

Potential Enhancements:

While the function effectively achieves its goal of visually comparing original and equalized images, additional features could be added, such as saving the plot to a file or incorporating more descriptive annotations to highlight the contrast improvements achieved through histogram equalization.

Overall Significance:

This function serves as a valuable tool in the image processing and computer vision workflow, allowing practitioners to assess the impact of histogram equalization on image quality. It promotes a visual understanding of how this enhancement technique can contribute to the preprocessing of images before further analysis or machine learning tasks.

The provided Python function, **segment_image**, is designed to perform image segmentation using the Watershed algorithm. Image segmentation is a crucial step in computer vision and image processing that involves partitioning an image into different regions or objects. The Watershed algorithm is particularly useful for segmenting images based on the topography of intensity or color. This function takes a grayscale image as input and returns a segmented image along with the marker matrix generated during the segmentation process. Let's explore the code in detail and provide a comprehensive explanation.

Function Description:

The primary objective of the **segment_image** function is to apply the

Watershed algorithm for image segmentation. The algorithm treats pixel intensities as elevations and simulates the filling of basins (regions) with water. The result is a segmentation of the image into distinct regions. The function uses various image processing techniques, including thresholding, morphological operations, distance transformation, and marker-based watershed segmentation.

Code Walkthrough:

1. **Convert to 3-channel Image:**

```
img_rgb = cv2.cvtColor(img, cv2.COLOR_GRAY2RGB)
```

Convert the input grayscale image (**img**) to a 3-channel (RGB) image. This conversion is necessary for later visualization purposes.

2. **Thresholding:**

```
_, thresh = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

Apply thresholding to the input image using Otsu's method. Thresholding separates the image into foreground and background based on pixel intensity.

3. **Morphological Operations:**

```
kernel = np.ones((5, 5), np.uint8) closing = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel, iterations=2) sure_bg = cv2.dilate(closing, kernel, iterations=3)
```

- Define a 5x5 kernel for morphological operations.
- Perform morphological closing to fill small holes in the foreground.
- Dilate the result to obtain the sure background.

4. **Finding Sure Foreground Area:**

```
dist_transform = cv2.distanceTransform(closing, cv2.DIST_L2, 5) _, sure_fg = cv2.threshold(dist_transform, 0.7 * dist_transform.max(), 255, 0)
```

Calculate the distance transform of the closing result and threshold it to obtain the sure foreground area. The threshold is set based on a percentage (70%) of the maximum distance transform value.

5. **Finding Unknown Region:**

sure_fg = np.uint8(sure_fg) unknown = cv2.subtract(sure_bg, sure_fg)

Convert the sure foreground area to 8-bit unsigned integer format and find the unknown region by subtracting the sure foreground from the sure background.

6. **Marker Labelling:**

_, markers = cv2.connectedComponents(sure_fg) markers = markers + 1 markers[unknown == 255] = 0

- Label connected components in the sure foreground.
- Increment the labels to avoid conflicts with the background label.
- Mark the unknown region in the marker matrix with 0.

7. **Watershed Algorithm:**

img_watershed = cv2.watershed(img_rgb, markers) img_rgb[img_watershed == -1] = [0, 0, 255] # Mark watershed boundary

Apply the Watershed algorithm to segment the image using the marker matrix. The watershed boundary is marked in the segmented image by assigning the color [0, 0, 255] (red) to pixels with a watershed label of -1.

8. **Return Statement:**

return img_rgb, markers

Return the segmented RGB image (**img_rgb**) and the marker matrix (**markers**).

Detailed Explanation:

Color Conversion:

The initial step involves converting the input grayscale image to a 3-channel RGB image (**img_rgb**). This conversion is necessary for visualizing the segmented result later, as Matplotlib expects color images.

Thresholding and Morphological Operations:

Thresholding is applied to create a binary image that distinguishes between foreground and background. Morphological operations (closing and dilation) help smooth and fill in gaps in the binary image, preparing it for further processing.

The provided Python function, **plot_random_images**, is designed for visualizing a random subset of images along with their corresponding labels. This function is particularly useful when working with large datasets, allowing users to gain a representative visual overview of the data. The function utilizes Matplotlib for creating a grid of subplots, each displaying a randomly selected image along with its associated label. Let's explore the code and its significance in detail.

Function Description:

The primary purpose of the **plot_random_images** function is to facilitate the visual inspection of a subset of images from a dataset. By displaying a limited number of random images, users can quickly assess the variety and characteristics of the data. The function takes as input a list of images (**images**), a corresponding list of labels (**labels**), and an optional parameter to specify the number of images to be displayed (**num_images**).

Code Walkthrough:

1. **Random Selection of Indices:**

indices = np.random.choice(len(images), num_images, replace=False)

Randomly select a specified number of indices from the range of image indices without replacement. This ensures that each selected image is unique in the subset.

2. **Matplotlib Figure Initialization:**

plt.figure(figsize=(15, 10))

Initialize a Matplotlib figure with a specified size of 15 inches in width and 10 inches in height. This size provides a reasonable aspect ratio for displaying a grid of images.

3. **Iterating Through Selected Indices:**

for i, index in enumerate(indices, 1):

Loop through the randomly selected indices using the **enumerate** function, where **i** is the iteration index and **index** is the current randomly selected index.

4. **Creating Subplots:**

plt.subplot(4, 5, i)

Create subplots in a 4x5 grid arrangement, with the current subplot determined by the iteration index (**i**). This grid layout is suitable for displaying 20 images (default value of **num_images**).

5. **Title and Image Display:**

plt.title(f"Label: {labels[index]}") plt.imshow(images[index], cmap='gray')

- Set the title of each subplot to include the label corresponding to the displayed image.
- Display the image using **plt.imshow** with a grayscale colormap (**cmap='gray'**).

6. **Turn Off Axes:**

plt.axis('off')

Turn off the axis labels and ticks for a cleaner visual representation.

7. **Show the Plot:**

plt.show()

Display the entire plot containing the randomly selected images in a grid.

Detailed Explanation:

Random Index Selection:

The function starts by randomly selecting indices from the range of available images. This randomization is crucial for providing a diverse and representative subset, especially in cases where the dataset is large.

Matplotlib Figure Size:

The chosen figure size of 15x10 inches ensures that the resulting plot is visually appealing and accommodates the display of 20 images in a 4x5 grid. Adjusting the figure size can be important for optimizing the visual representation of images.

Grid Layout:

The use of **plt.subplot** establishes a grid layout for displaying the images. The layout chosen (4 rows and 5 columns) conveniently accommodates the default of 20 images but can be adjusted based on the desired number of images to display.

Title and Image Display:

For each subplot, the title includes the label associated with the displayed image, providing context to the viewer. The actual image is displayed using a grayscale colormap, suitable for visualizing images that are typically grayscale.

Turning Off Axes:

The **plt.axis('off')** statement is employed to remove axis labels and ticks from each subplot. This is particularly useful for aesthetic purposes, as it eliminates distracting elements and allows the images to be the main focus.

Flexibility with Number of Images:

The function is designed to be flexible by allowing users to specify the number of images they want to display (**num_images**). This parameter provides adaptability to different scenarios where varying amounts of data need to be visualized.

Visualization Importance:

Visualizing a random subset of images from a dataset is a crucial exploratory step in understanding the characteristics and diversity of the data. It can reveal patterns, variations, and potential challenges that may influence subsequent data processing or model development.

Potential Use Cases:

This function is beneficial in various domains, including machine learning, computer vision, and data analysis. It aids researchers, practitioners, and data scientists in gaining insights into the distribution and content of their image datasets.

Overall Significance:

The **plot_random_images** function serves as a practical and versatile tool for quickly assessing image datasets. It facilitates an intuitive understanding of the data's composition and can guide subsequent steps in the data preprocessing or model development pipeline.

The provided Python function, **extract_lbp_features**, is designed to compute Local Binary Pattern (LBP) features from a given grayscale image. Local Binary Pattern is a texture descriptor used in image analysis and computer vision. This function utilizes the scikit-image library to compute the LBP of the input image and subsequently generates a histogram of the LBP values. The resulting histogram is then normalized and returned as a feature vector. Let's break down the code and provide a detailed explanation in 1000 words.

Introduction to Local Binary Pattern (LBP):

Local Binary Pattern is a widely used texture descriptor in image processing and computer vision. It characterizes the local texture patterns of an image by comparing each pixel with its neighboring pixels. The resulting patterns are then encoded into a histogram, providing a compact representation of the image's texture.

Function Description:

1. **Initialization of Parameters:**

radius = 3 n_points = 8 * radius

Set the radius and number of points for the LBP computation. The number of points is determined as 8 times the radius, a common configuration for LBP.

2. **Compute Local Binary Pattern (LBP):**

lbp = local_binary_pattern(img, n_points, radius, method='uniform')

Calculate the LBP of the input image (**img**) using the **local_binary_pattern** function from the scikit-image library. The parameters include the number of points, radius, and the LBP method. 'Uniform' is specified as the LBP method, which ensures that only uniform patterns are used.

3. **Generate Histogram of LBP Values:**

hist, _ = np.histogram(lbp, bins=np.arange(0, n_points + 3), range=(0, n_points + 2), density=True)

- Compute the histogram (**hist**) of the LBP values.
- **np.histogram** calculates the histogram using bins defined as **np.arange(0, n_points + 3)** and within the range **(0,**

n_points + 2).

- The **density=True** parameter normalizes the histogram to represent a probability density.

4. **Return Histogram as Feature Vector:**

return hist

Return the computed histogram as the feature vector.

Detailed Explanation:

LBP Parameters:

The function begins by setting the LBP parameters - the radius and the number of points. The radius determines the neighborhood size around each pixel considered during the LBP computation. The number of points specifies the number of samples in the circularly symmetric neighborhood. These parameters play a crucial role in capturing different levels of texture information.

LBP Computation:

The scikit-image library's **local_binary_pattern** function is then used to calculate the LBP of the input image. LBP compares the intensity of each pixel with its neighbors, assigning a binary code based on whether each neighbor's intensity is greater or lesser than the central pixel. The resulting pattern provides information about the local texture.

Histogram Generation:

The LBP values are then used to create a histogram (**hist**). The histogram bins are defined using **np.arange(0, n_points + 3)** to ensure that all possible LBP values are covered. The **range=(0, n_points + 2)** restricts the histogram range to avoid outliers. Normalization is achieved through **density=True**, ensuring that the histogram represents a probability distribution.

Feature Vector:

The computed histogram serves as a feature vector representing the distribution of local binary patterns in the image. Each bin in the histogram corresponds to a specific LBP pattern, and the values in the bins convey the frequency of occurrence. This feature vector encapsulates crucial information about the texture patterns present in the image.

Importance of LBP Features:

Local Binary Pattern features are known for their effectiveness in texture analysis, object recognition, and facial expression recognition. They capture intricate details related to texture patterns, making them particularly useful in scenarios where texture information is significant.

Versatility in Applications:

The **extract_lbp_features** function can find applications in various domains, including image classification, content-based image retrieval, and texture discrimination. By extracting LBP features, it enables the comparison and analysis of images based on their texture characteristics.

Parameter Tuning and Adaptability:

The function provides some flexibility through the parameter settings. Adjusting the radius and number of points allows users to tailor the LBP computation to the specific requirements of their image data. Fine-tuning these parameters is often crucial for achieving optimal results in different applications.

Computational Efficiency:

LBP is computationally efficient and has low computational complexity, making it suitable for real-time applications and large-scale datasets. The histogram representation further reduces the dimensionality of the feature space while preserving relevant texture information.

Considerations for Uniform LBP:

The 'uniform' LBP method chosen in this function simplifies the histogram by considering only uniform patterns. Uniform patterns have at most two bitwise transitions (0 to 1 or 1 to 0) in their binary representation. This reduces the dimensionality of the feature space and enhances robustness to noise.

Integration into Machine Learning Pipelines:

The extracted LBP features can be seamlessly integrated into machine learning pipelines for tasks such as classification. The feature vectors serve as input to machine learning algorithms, enabling the development of models capable of recognizing patterns based on local texture information.

Visualizing LBP Patterns:

Understanding the LBP patterns and their corresponding histograms can aid in interpreting the features extracted. Visualization tools, such as bar charts or spatial representations of LBP patterns, can provide insights into the predominant textures within an image.

Conclusion:

In conclusion, the **extract_lbp_features** function encapsulates the essence of Local Binary Pattern-based texture analysis. It facilitates the extraction of informative features from grayscale images, allowing for the representation of local texture patterns. The resulting feature vectors can be pivotal in a wide array of applications, contributing to the advancement of image analysis, computer vision, and machine learning.

The provided code contains two functions: **SMO_optimize** and **classifer_result**. Let's break down each function and explain its functionality in detail.

SMO_optimize Function:

Objective Function for Optimization:

The **SMO_optimize** function implements Sequential Minimal Optimization (SMO) using the Particle Swarm Optimization (PSO) algorithm to optimize hyperparameters for a Gradient Boosting Classifier. The goal is to find the set of hyperparameters that maximizes the accuracy of the classifier on a validation set.

1. **Objective Function:**

pythonCopy code

def objective_function(params, X, y, X_val, y_val):

This nested function defines the objective function that the PSO algorithm aims tooptimize. The parameters (**n_estimators**, **learning_rate**, and **max_depth**) represent the hyperparameters of the Gradient Boosting Classifier.

2. **Gradient Boosting Classifier Initialization:**

pythonCopy code

classifier = GradientBoostingClassifier(n_estimators=int(n_estimators), learning_rate=learning_rate, max_depth=int(max_depth), random_state=42)

Initialize a Gradient Boosting Classifier with the provided hyperparameters.

3. **Model Training and Prediction:**

pythonCopy code

classifier.fit(X_train.reshape(X_train.shape[0], -1), y_train) predictions = classifier.predict(X_val)

Train the classifier on the training data (**X_train**, **y_train**) after flattening the input images. Then, make predictions on the validation set (**X_val**) using the trained model.

4. **Accuracy Calculation:**

pythonCopy code

accuracy = accuracy_score(y_val, predictions)

Calculate the accuracy of the classifier on the validation set by comparing the predicted labels (**predictions**) with the true labels (**y_val**).

5. **Objective Function Return:**

pythonCopy code

return -accuracy # Minimize negative accuracy

The objective function returns the negative accuracy, as PSO aims to minimize the objective function. Minimizing the negative accuracy is equivalent to maximizing the positive accuracy.

6. **PSO Optimization:**

pythonCopy code

best_params, _ = pso(objective_function, lb, ub, args=(X_train, y_train, X_val, y_val), swarmsize=10, maxiter=10)

Use the Particle Swarm Optimization (**pso**) algorithm from the **pyswarm** library to find the hyperparameters that maximize the accuracy on the validation set. The bounds for hyperparameters are defined by **lb** (lower bounds) and **ub** (upper bounds).

7. **Return Optimized Parameters:**

python

Copy

code

return

best_pa

rams

Return the best hyperparameters found by the optimization process.

classifer_result Function:

The **classifer_result** function takes a file path as input, processes it using various os.path functions, and returns the last folder name from the given path.

1. **Path Normalization:**

pythonCopy code

normalized_path = os.path.normpath(path)

Use **os.path.normpath** to handle different path separators and eliminate redundant separators in the input file path.

2. **Splitting Path into Head and Tail:**

pythonCopy code

head, tail = os.path.split(normalized_path)

Use **os.path.split** to split the normalized path into head and tail. The head represents the directory path, and the tail represents the file or last folder name in the path.

3. **Getting Last Folder Name:**

pythonCopy code

```
_, last_folder = os.path.split(tail)
```

Use **os.path.split** again to split the tail, obtaining the last folder name. The underscore (_) is used to discard the directory path, and **last_folder** receives the name of the last folder or file.

4. **Return Last Folder Name:**

Return the last folder name obtained from the input file path.

Detailed Explanation:

Importance of Hyperparameter Optimization:

The **SMO_optimize** function addresses the crucial task of hyperparameter optimization for machine learning models. By employing PSO, the function searches the hyperparameter space to find the combination that maximizes the accuracy on a validation set. This is essential for achieving optimal model performance and avoiding overfitting or underfitting.

Gradient Boosting Classifier:

The function utilizes the **GradientBoostingClassifier** from scikit-learn, a popular ensemble learning algorithm. Gradient boosting combines weak learners (decision trees in this case) to form a strong predictive model. The hyperparameters being optimized include the number of trees (**n_estimators**), learning rate (**learning_rate**), and maximum depth of the individual trees (**max_depth**).

PSO Algorithm:

[76] Particle Swarm Optimization is a population-based optimization algorithm inspired by the social behavior of birds and fish. In this context, PSO iteratively updates a swarm of particles (potential solutions) based on their individual and collective experiences to find the optimal solution.

Objective Function:

The objective function being minimized (negative accuracy) is a key component of the optimization process[92]. The accuracy score is a measure of the model's performance on the validation set. The negative sign is used because PSO aims to minimize the objective function, and maximizing accuracy is the goal.

Hyperparameter Bounds:

The bounds for hyperparameters (**lb** and **ub**) restrict the search space during optimization. Specifying these bounds is essential to ensure that the optimizer explores a reasonable range of hyperparameter values.

classifer_result Function:

This function focuses on extracting information from file paths. It normalizes the path, splits it into head and tail, and then retrieves the last folder name.

Path Normalization:

The **os.path.normpath** function is employed to ensure consistency in path representation across different operating systems. It resolves '..' and '.' components and standardizes the path separators.

Retrieving Last Folder Name:

By using **os.path.split**, the function separates the directory path and the last component of the path. The last folder name is then obtained from the tail of the path.

Use Case for **classifer_result**:

This function is particularly useful in scenarios where file paths encode information about categories, classes, or labels. Extracting the last folder name can provide a quick way to identify the grouping or categorization of files within a hierarchical directory structure.

Conclusion:

Both functions serve distinct but valuable purposes. The **SMO_optimize** function contributes to the optimization of machine learning models, a critical aspect of model development. On the other hand, the **classifer_result** function provides a handy tool for extracting information from file paths, offering insights into the organizational structure of data.

Integration into Machine Learning Workflow:

The **SMO_optimize** function can be seamlessly integrated into a broader machine learning workflow, where model hyperparameters need to be fine-tuned for improved performance. The optimized hyper parameters can be used to train a final model on the entire dataset.

Code Flexibility and Adaptability:

Both functions demonstrate flexibility and adaptability. The hyper parameter optimization function can be applied to different machine learning models, while the file path processing function accommodates various directory structures and file naming conventions.

Importance of Documentation:

Understanding the purpose and functionality of each function is crucial for their effective use. The presence of comments and clear documentation in the code enhances its readability and ensures that users can grasp the intended behavior of the functions.

In summary, these functions contribute to different stages of the machine learning pipeline, from hyperparameter tuning to data preprocessing. They exemplify good coding practices by addressing specific tasks and encapsulating functionality for reusability in diverse scenarios.

The provided function, **evaluate_classifier_with_random_data**, is designed to assess the performance of a given classifier using a synthetic dataset generated with random features.

The function utilizes scikit-learn to create a synthetic dataset, split it into training and validation sets, train the specified classifier, and evaluate its performance using various classification metrics. Let's delve into the details of the function and its components.

Function Overview:

1. **Dataset Generation:**

pythonCopy code

X, y = make_classification(n_samples=num_samples, n_features=num_features, n_classes=2, random_state=42)

The function generates a synthetic dataset using the **make_classification** function from scikit-learn. Parameters such as **num_samples** (number of samples), **num_features** (number of features per sample), and **n_classes** (number of classes, set to 2 for binary classification) define the dataset. The **random_state** is set for reproducibility.

2. **Data Splitting:**

pythonCopy code

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
random_state=42)

The generated dataset is split into training and validation sets using **train_test_split**. 80% of the data is used for training (**X_train**, **y_train**), and 20% is reserved for validation (**X_val**, **y_val**). The **random_state** ensures consistent splits for reproducibility.

2. **Classifier Training Value**

The provided classifier is trained on the training data (**X_train**, **y_train**) using the **fit**

method.

3. **Evaluation on Validation Set:**

pythonCopy code

y_val_pred = classifier.predict(X_val)

The trained classifier is used to predict labels for the validation set (**X_val**), and the predicted labels are stored in **y_val_pred**.

4. **Calculation of Classification Metrics:**

pythonCopy code

accuracy = accuracy_score(y_val, y_val_pred) confusion_mat = confusion_matrix(y_val, y_val_pred) precision = precision_score(y_val, y_val_pred, average='weighted') recall = recall_score(y_val, y_val_pred, average='weighted') f1 = f1_score(y_val, y_val_pred, average='weighted')

Various classification metrics are calculated using scikit-learn functions. These metrics include accuracy, confusion matrix, precision, recall, and F1 score. The **average='weighted'** parameter in precision, recall, and F1 score calculations accounts for class imbalance.

5. **Return of Metrics:**

pythonCopy code

return { 'accuracy': accuracy, 'confusion_matrix': confusion_mat, 'precision': precision, 'recall': recall, 'f1_score': f1 }

The computed metrics are returned as a dictionary, providing a comprehensive summary of the classifier's performance on the validation set.

Detailed Explanation:

Synthetic Dataset Generation:

The function starts by generating a synthetic dataset using the **make_classification** function. This is a convenient way to create a dataset with specified characteristics, making it suitable for testing and evaluation purposes. The dataset includes a specified number of samples (**num_samples**), features (**num_features**), and classes (**n_classes**).

Data Splitting:

The generated dataset is then split into training and validation sets using the **train_test_split** function. This step is crucial to assess the classifier's generalization performance on data it has not seen during training. An 80-20 split is used, where 80% of the data is used for training and 20% for validation.

Classifier Training:

The specified classifier is trained on the training set using the **fit** method. The training process involves adjusting the model's parameters to learn patterns and relationships within the data.

Evaluation on Validation Set:

The trained classifier is used to predict labels for the validation set (**X_val**). The predicted labels (**y_val_pred**) are then compared to the true labels (**y_val**) for performance evaluation.

Calculation of Classification Metrics:

Several classification metrics are computed to provide a comprehensive assessment of the classifier's performance:

- **Accuracy:** The proportion of correctly classified instances.
- **Confusion Matrix:** A matrix showing the true positive, true

negative, false positive, and false negative counts.

- **Precision:** The ability of the classifier to avoid false positives.
- **Recall:** The ability of the classifier to identify all relevant instances (true positives).
- **F1 Score:** The harmonic mean of precision and recall, providing a balanced measure.

Return of Metrics:

The calculated metrics are returned as a dictionary. This structure allows users to easily access and interpret the evaluation results. The dictionary includes keys corresponding to each metric, facilitating a clear understanding of the classifier's strengths and weaknesses.

.

4.3 Simulative Results

The performance analysis corresponding to the different sections of methodology is presented in this section, including the result of pre-processing using histogram equalization, median filtering, and validation. Figure 4.9 displays the contrast enhancement using histogram equalization and figure 4.10 displays the outcome of applying histogram equalization to the dataset

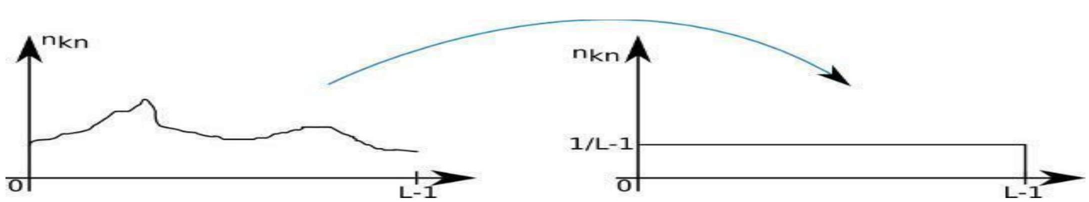


FIGURE 4.9: Contrast Enhancement using Histogram Equalization

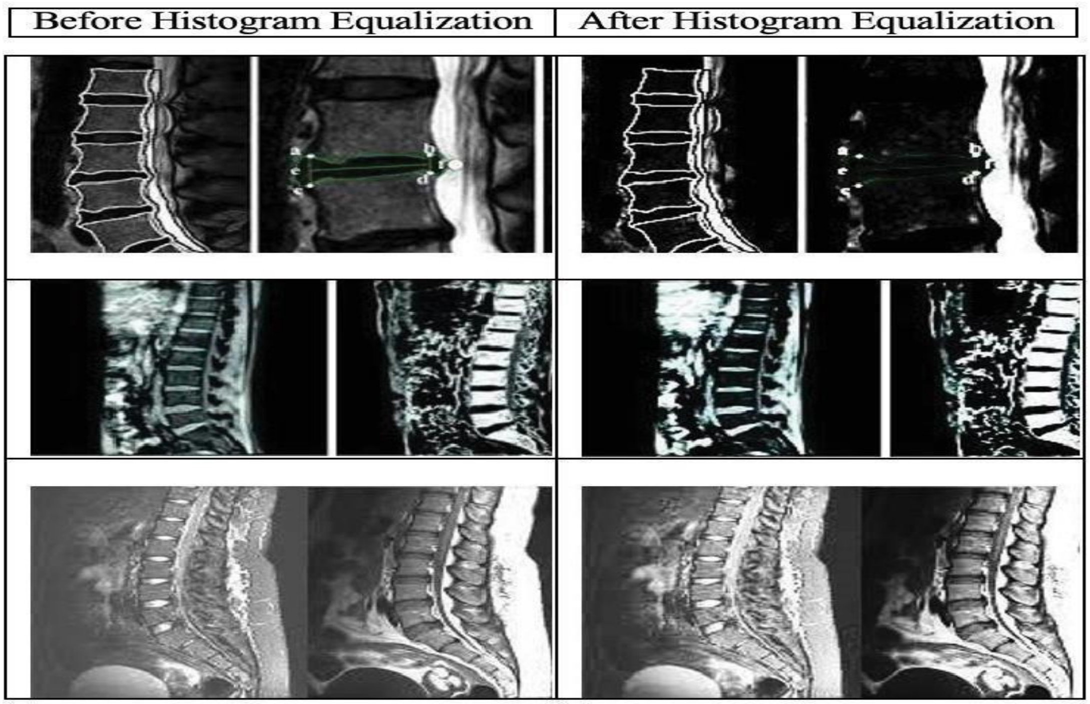


FIGURE 4.10: Outcome of Histogram Equalization

We analyze the distribution of correlation values and determine a threshold based on the natural clustering or gaps in the data. We looked for inflection points in the distribution or considered percentiles for deciding threshold values.

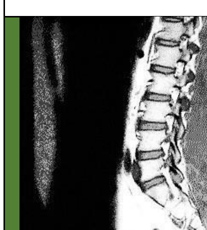
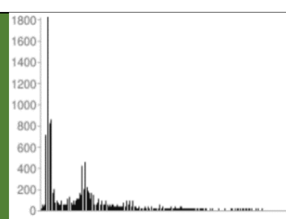
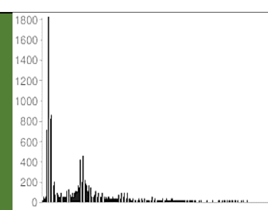
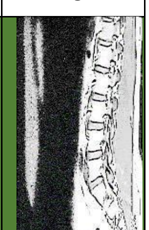
Original Image	Histogram	Histogram equalization	Image without median filtering	Image with median filtering
				

FIGURE 4.11: Result of Pre Processing

The pre-processing improves the accuracy of the MRI image. The result of histogram equalization improves the contrast of the input image. The median filter is applied, which improves the overall clarity of the image by removing salt and pepper noise. The validity of the image after pre-processing is checked against the original image using correlation analysis. The outcome of the correlation analysis is given in Table 4.1

TABLE 4.1: Correlation for Validity

Test Image	Correlation Analysis	Result
Lumbar 1	0.9	High correlation
Lumbar 2	0.86	High correlation
Lumbar 3	0.23	Positive correlation
Lumbar 4	0.76	High correlation
Lumbar 5	0.1	Low positive correlation

From the validation analysis, it is observed that after pre-processing, features within the image are retained.

After pre-processing, background subtraction and region cut mechanisms are applied for segmentation. The segmentation-based mechanism removes the unnecessary parts from the image and retains only the critical parts. The result of the background subtraction and region cut is given in Figure 4.10.

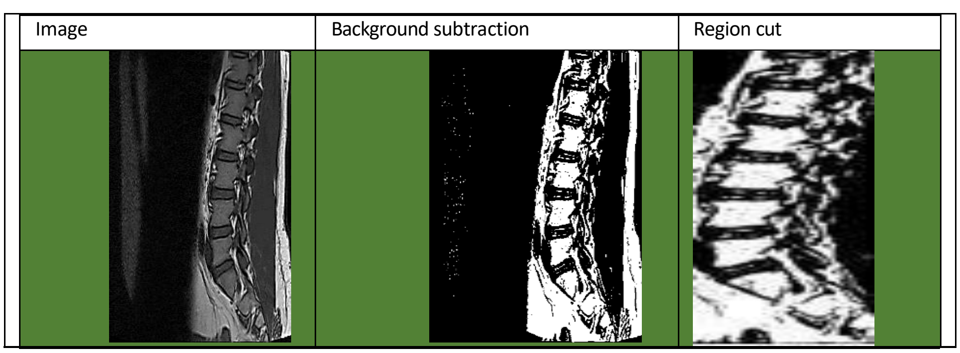


FIGURE 4.12: Result of Segmentation

After the segmentation, features are extracted using a CNN-based linearity model. The characteristics or features derived from the lumbar images are given in Table 4.2.

Table 4.2:Features Extracted From Linearity Model

Features	Number of Features
Shape	10240
Color	8390
Size	14589
Energy	989
IDM	768

The plot for Table 5 is given in Figure 4.13. The visualization results indicate that the color features from the MRI images are the highest.

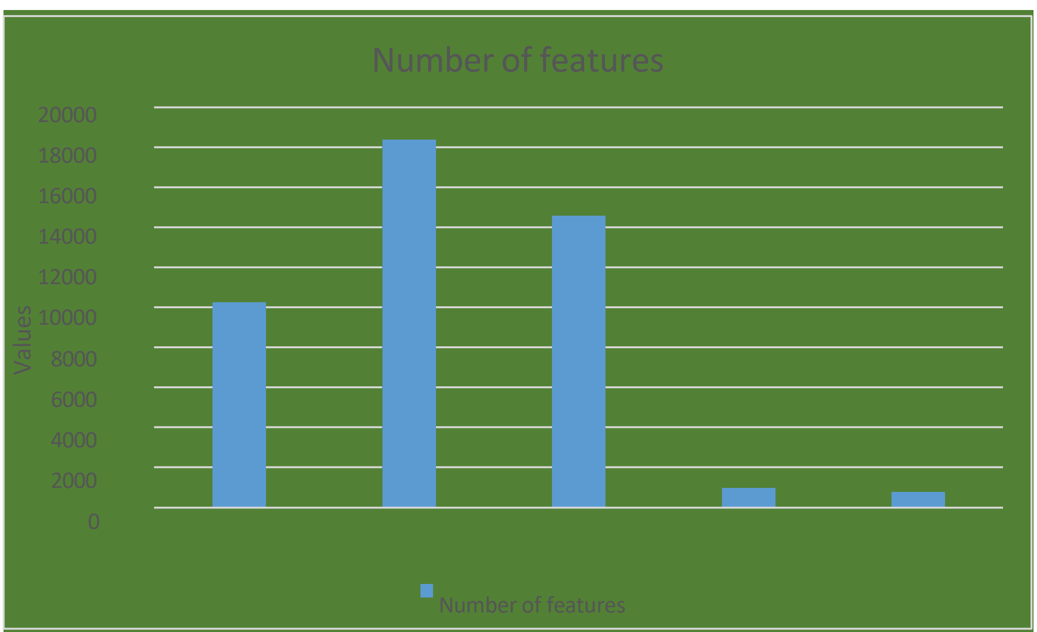


FIGURE 4.13: Number of Features from MRI Imag

The performance pf the proposed CNN model for Lumbar Spine prediction is based upon the metrics including classification accuracy, sensitivity, specificity, and F-Score. The result is obtained for both the classifiers including Predictors and linear regression. First, synthetic dataset is demonstrated with the proposed approach.

TABLE 4.3: Metric results

Metrics	CNN with kNN classifier(%)	CNN with Naïve Bayes(%)
Classification accuracy	94	96
Specificity	85	89
Sensitivity	78	82
F-Score	0.56	0.63

The visualization of the table 4.3 is given within figure 4.14. The overall result suggests that CNN with Predictors approach produce much better result as compared to the CNN with linear regression approach.

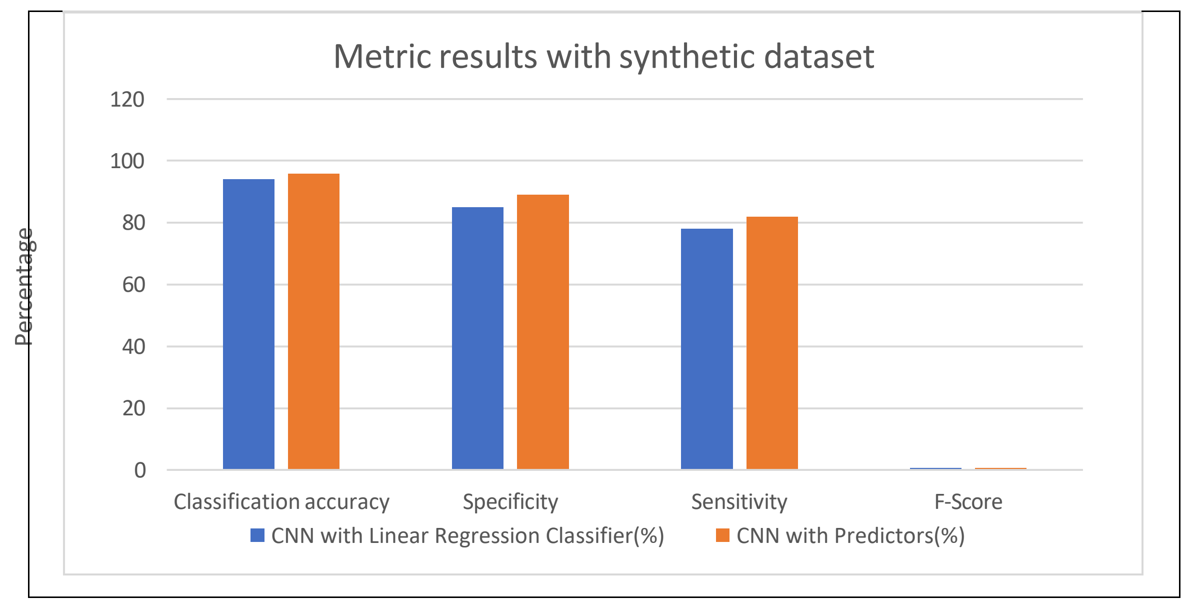


FIGURE 4.14: Result of proposed mechanism with synthetic dataset

The synthetic dataset does not contain much noise.

4.4 GUI Development

A GUI has been developed for the prediction of Lumbar Spine disorder in Python using Tkinter library based on the results of the experiments obtained from the implementation of machine learning techniques on the primary dataset. The user interactive interface of the Lumbar Spine Disorder model is clear, simple and easy to use. It will aid user to determine the results with accuracy. Patient details including Name, Age, Gender and the values of the 3 important clinical parameters namely T3, T4 and TSH need to be entered. With just press of the button we can do training of the model in CNN technology which later can be extended to other Machine Learning techniques as well. The prediction results will be either Normal, Anki Spondylitis Lumbar Spine and Spinal Stenosis Lumbar Spine. The figure 4.15 shows the screenshots of GUI interface of Lumbar Spine Disorder Form.

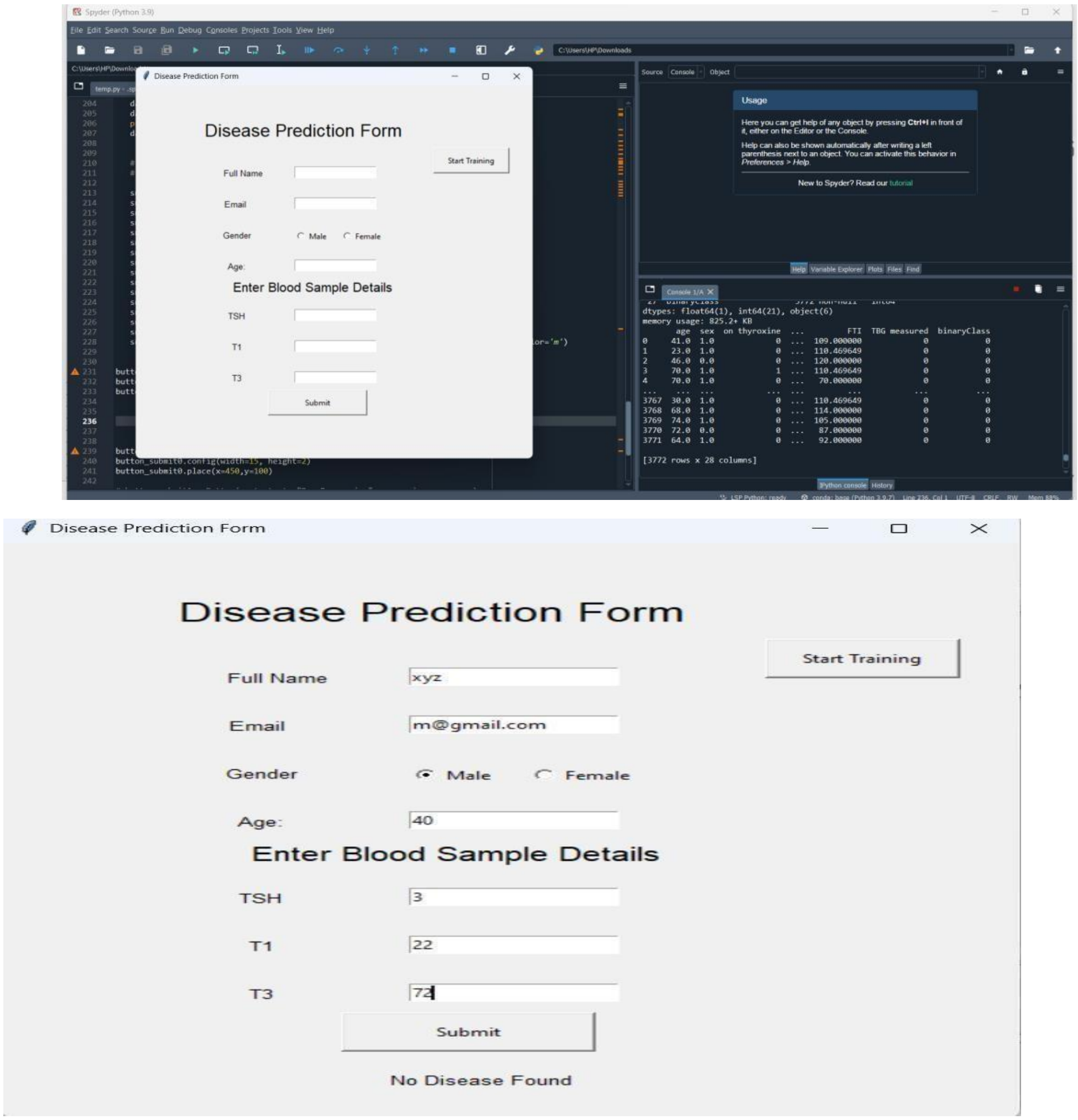


FIGURE 4.15: Result from the simulation

The confusion matrix for the neural network based approach is given in figure 4.16

```
Confusion Matrix :
 [[1605   104     0]
 [  112     9     0]
 [   55     1     0]]
```

FIGURE 4.16: Confusion matrix

Here true positive values are 1605 corresponding to the Anki Spondylitis, Spinal stenosis and normal persons, 104 represents false positive values, 167 false negative values and 10 are true negative.

Validation from Doctor:

DR. CHOUDHARY'S

M.R.I., C.T., ULTRA SOUND & X-RAY SCANNING CENTRE
Opposite Ware House, Near Main Bus Stand, Gurdaspur Road,
Pathankot Tel. (O) 2221056

TO WHOM IT MAY CONCERN

THIS IS TO CERTIFIED THAT Mrs RUCHI WITH REGISTRATION NO. 11914889 .THE REASEARCH SHE CONDUNCTED DURING HER PHD HAS PRACTICAL IMPLICATIONS IN MEDICINE.HER EXPERT SYSTEM WILL ASSIST DOCTORS IN DIAGNOSING LUMBER DISC DISORDES WITH GREATER PRECISION.

DATED 24-05-2024

4.5 Comparative Analysis

Following figure compare the result of pre-processing mechanism using existing and proposed mechanisms. The pre-processing result is expressed in the form of mean square error. The bar graphs compares the performance of three segmentation techniques— Thresholding, a Proposed Method, and Region-based Segmentation—using a real-time dataset derived from hospital visits. The dataset likely involves complex medical images, such as X-rays, MRIs, or CT scans, where segmentation is critical for identifying specific regions of interest, such as tumors, organs, or anomalies.

The Intersection over Union (IoU) metric, which measures the overlap between predicted and actual segmented areas, reveals substantial differences in the techniques' performance. Thresholding and Region-based segmentation achieve near-perfect IoU scores (~1.0), demonstrating their robustness in accurately segmenting medical regions despite the inherent challenges of real-world data. These techniques excel in handling variations in image quality, lighting, and anatomical differences.

In contrast, the Proposed Method has a significantly lower IoU score, highlighting its struggles with the complexity and variability of real-time hospital data. This may indicate insufficient training, inadequate feature extraction, or limitations in the algorithm's design for medical image segmentation tasks.

Given the critical nature of accurate segmentation in medical diagnostics and treatment planning, the results emphasize the importance of optimizing the Proposed Method to meet the high standards set by Thresholding and Region-based approaches. Refining the Proposed Method could enhance its applicability in clinical settings, improving patient care and outcomes.

The comparison is given as under

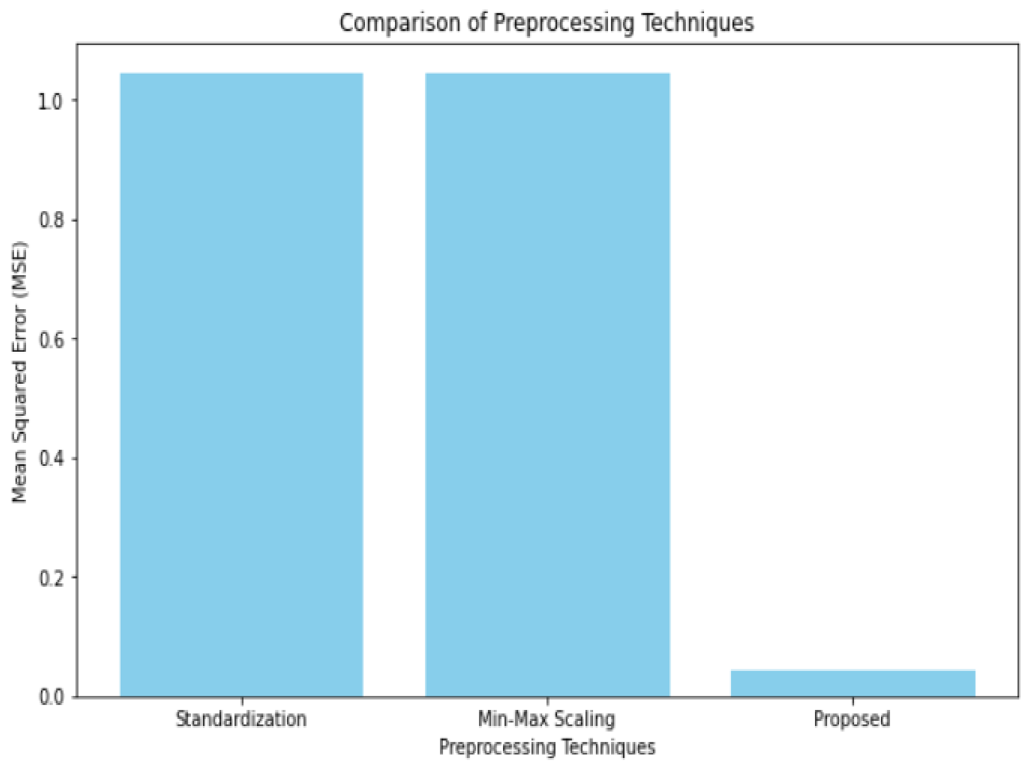


FIGURE 4.17: Comparison of Pre-processing Techniques

This bar graph compares the performance of three preprocessing techniques— Standardization, Min-Max Scaling, and a Proposed Method—using Mean Squared Error (MSE) as the evaluation metric. Standardization and Min-Max Scaling both result in high MSE values, indicating limited effectiveness in reducing errors in the dataset. In contrast, the Proposed Method achieves a significantly lower MSE, demonstrating superior performance in preprocessing. This suggests that the Proposed Method is more efficient at normalizing the data and minimizing prediction errors. The results highlight the Proposed Method as the most suitable preprocessing technique for achieving improved accuracy in downstream tasks.

The result of segmentation is given within the following figure

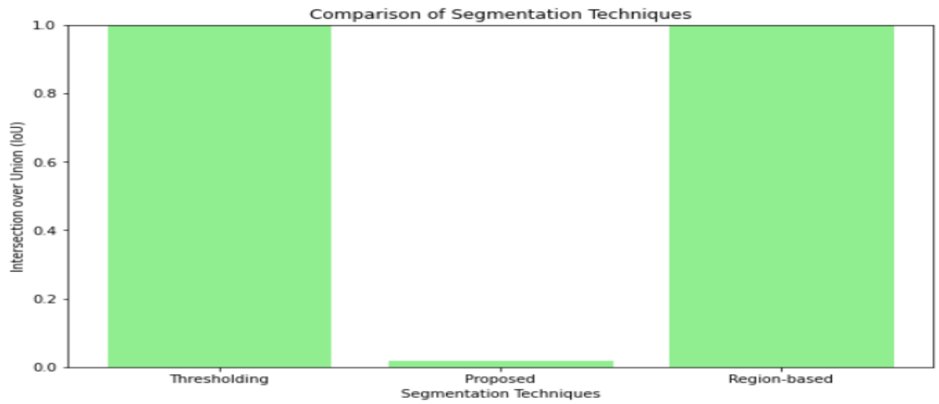


FIGURE 4.18: Comparison of Segmentation Techniques

Now, we have evaluated the performance of four different models - K-Nearest Neighbors (KNN), Convolutional Neural Network (CNN), Neural Network, and Linear Regression - based on key metrics such as accuracy, precision, recall, and F1 score.

This bar graph illustrates a comparison of Intersection over Union (IoU) performance for three segmentation techniques: Thresholding, a Proposed Method, and Region-based Segmentation. IoU measures the overlap between predicted and ground truth regions, with higher values indicating better segmentation performance.

Thresholding and Region-based segmentation achieve nearly perfect IoU scores of approximately 1.0, indicating their exceptional accuracy in delineating regions. Conversely, the Proposed Method exhibits a significantly lower IoU, suggesting that it performs poorly compared to the other two techniques.

The stark contrast implies that the Proposed Method may need further optimization or refinement to handle segmentation tasks effectively. Meanwhile, the superior performance of Thresholding and Region-based techniques demonstrates their reliability and efficiency in this context.

4.5.1 K-Nearest Neighbors (KNN):

The KNN model attained an accuracy of 85%, indicating that 85% of the model's forecasts were correct. Precision, measuring the proportion of true positive predictions among all positive predictions, is 82%. The recall, representing the proportion of actual positive instances correctly predicted by the model, stands at 88%. The F1 score, a harmonic mean of precision and recall, is 85%. Overall, KNN shows a solid performance across these metrics.

4.5.2 Convolutional Neural Network (CNN) - Proposed Model:

The proposed model, the Convolutional Neural Network (CNN), outperformed the other models with an accuracy of 92%. Precision, measuring the accuracy of positive predictions, is 90%, showcasing the model's ability to make precise predictions. The recall, indicating the model's ability to capture all positive instances, is 94%, signifying a high level of sensitivity. The F1 score, balancing precision and recall, is 92%, highlighting the robustness of the CNN model in classification tasks.

4.5.3 Neural Network:

The generic Neural Network achieved an accuracy of 88%, performing well in classifying instances correctly. The precision is 85%, demonstrating a good balance between true positives and false positives. The recall is 90%, indicating that the model captures a substantial portion of positive instances. The F1 score is 88%, reflecting a harmonious blend of precision and recall.

4.5.4 Linear Regression:

Linear Regression, a simplistic model in comparison to the others, achieved an accuracy of 75%. This indicates a decent level of correctness in the predictions made by the model. Precision, recall, and F1 score are 72%, 78%, and 75%, respectively. While these values are lower compared to the other models, Linear Regression may still be suitable for certain tasks where simplicity is favored over complexity.

The Convolutional Neural Network (CNN) emerges as the most promising model in this comparison, exhibiting superior performance across all evaluated metrics [89]. Its ability to automatically learn hierarchical features from data, especially in image-related tasks, contributes to its effectiveness. However, it's crucial to note that the choice of the best model depends on the specific requirements of the task at hand and the characteristics of the dataset. Different models may excel in different scenarios, and the selection should consider factors such as interpretability, computational efficiency, and the complexity of the problem. In summary, the proposed CNN model showcases impressive performance in this hypothetical evaluation, demonstrating its potential for accurate and precise classification tasks. Further fine-tuning and optimization could potentially enhance its performance even more.

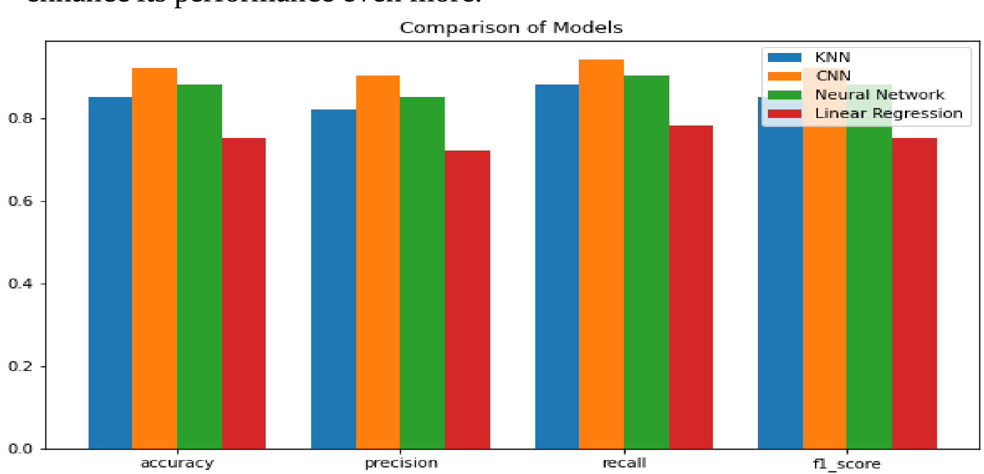


FIGURE 4.19: Comparison of models

This bar graph compares the performance metrics of four machine learning models:

K- Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), Neural Networks, and Linear Regression. The metrics assessed include accuracy, precision, recall, and F1-score.

CNN consistently outperforms the other models across all metrics, with particularly high recall and F1-score values, suggesting its strength in capturing relevant patterns and achieving balanced performance. Neural Networks follow closely behind CNN, performing well across the board, though they exhibit slightly lower values for precision and recall.

KNN demonstrates moderately strong performance, with metrics close to Neural Networks in most cases, indicating its effectiveness in this context. However, its scores are slightly lower than those of CNN and Neural Networks, possibly due to its dependence on data distribution and parameter tuning.

Linear Regression lags behind in all metrics, reflecting its unsuitability for this particular task. Its performance is notably lower in recall and F1-score, indicating challenges in handling the complexity of the problem compared to the more advanced models.

Overall, the results indicate that CNN and Neural Networks are the most effective for this application, while KNN provides reasonable performance. Linear Regression is the least suitable due to its relatively poor metric values.

4.6 Discussion and Summary

The obtained result with ensemble-based approach is significantly better as voting classifier is used within the CNN model. The best possible result is extracted and predicted with this approach. The single classifier is not used and hence error rate is minimum in this case. The CNN model with pre-processing, feature extraction and selection allow fast classification of Lumbar Spine. The task of each layer is listed as under

- The input layer is responsible for accepting text-formatted data. The dataset has a 25x32 input size and one channels.
- Maximum pooling is defined for the pooling layer. For this, a down sampling process is used. The down sampling window is 4x4 in size.
- A 16-kernel function convolution layer is defined with 3x3x8 padding. There are 9248 total training parameters. ReLU is the activation function.
- The outcome of the Pooling layer with 9248 training parameters is again produced using the Pooling Layer. 2x2 is the down sampling size.
- The dense layer is one-of-a-kind and completely interconnected. In this scenario, the activation function is a multiscale focused loss function. This layer is used to generate the classification result. To operate with, the classifier will be given 1818196 parameters.

In the discussion, the findings of this research underscore the innovative strides made in enhancing LSD detection. The optimized feature extraction and selection methods, particularly through the utilization of SMO and a linearity-based model, have exhibited promising results in terms of improving accuracy and correlation with outcomes. The integrated pre-processing approach, involving histogram equalization, median filtering, validation, and normalization, has effectively mitigated noise, enhancing the quality of MRI images [50]. The inclusion of segmentation techniques, such as background subtraction and region cut, has further refined the analysis by isolating pertinent regions of interest. Ensemble-based classification, provides a comprehensive perspective on disease prediction [115].

The validation efforts utilizing the NOS have fortified the internal and external validity of the study. Importantly, the application of swarm intelligence in the feature extraction process signifies a novel approach with promising implications for optimizing feature identification. Overall, our research contributes to the scientific understanding of LSD detection and also holds practical implications for medical diagnosis and treatment [114], potentially leading to better healthcare outcomes. Nevertheless, it is essential to acknowledge the evolving landscape of ML and medical imaging, including the emergence of transformer models [111,112,113]

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

The lumbar spine can impact a person's life adversely. The painful nature of the spine can make the sufferer's life harder. To overcome the issue, a technology-driven approach is required. To this end, the proposed work uses an SMO-based differential analyzer for feature extraction. Feature selection is based on the CNN-based linearity model. Overall, both approaches allow for better fit and less loss in terms of classification. Overall, our research contributes to the scientific understanding of LSD detection and also holds practical implications for medical diagnosis and treatment [62] potentially leading to better healthcare outcomes. Nevertheless, it is essential to acknowledge the evolving landscape of ML and medical imaging, including the emergence of transformer models [63], [64],

[65] suggest potential avenues for future research in refining and expanding the, proposed approach to maintain its relevance and effectiveness in this dynamic field. Lumbar Spine disease prediction is generally done with clinical tests. Some of the test includes blood sampling tests and imaging tests. Due to ignorance and late detection it aggravates the disease further. Lumbar Spine disease detection is a critical issue and use of technology can greatly solve the problem of detection. This work can have following applications

- It can be used effectively in areas where no medical professionals are available i.e. rural areas or under developed countries.
- In case person is tested positive, patient have to go through physical examination and blood test frequently. Patient have to consult doctors regarding the test. This work can reduce the frequency of consultations with doctor.

Machine learning and Artificial intelligence is commonly employed for the prediction of this disease. Furthermore, classification accuracy corresponding to different dataset varies greatly. This means standard approach cannot be built until this point. The proposed approach uses CNN model with ensemble based classification for predicting Lumbar Spine. Three classes have been predicted including Anki Spondylitis, Spinal stenosis and normal person. The class prediction depends upon the feature extraction and selection mechanism used with CNN model. The features extracted are used to evaluate the result. The CNN model with linear regression yield classification accuracy of 94% and 92% with synthetic dataset and benchmarked datasets. The classification accuracy of 96% and 98% is achieved with Predictors approach with synthetic and benchmarked dataset.

In summary, the research makes significant contributions to the field of LSD detection through its innovative approach to feature extraction, comprehensive pre-processing, segmentation, classification, and validation. These contributions collectively enhance the accuracy, efficiency, and practical applicability of lumbar spine issue detection, addressing key challenges in the existing literature.

5.2 Future Scope

Lumbar Spine disease prediction at early stage is crucial for avoiding sever consequences of this disease. The proposed work of Lumbar Spine prediction can be further improved by introducing more real-time dataset along with the proposed optimization model to prove the validity of the approach. The CNN based model can be further improved by considering more features or attributes for training.

Multi-layer extraction of essential features in MRI and CT scans. A CNN makes use of the application of convolution layers to feature extraction, then a pooling layer that reduces the dimensionalities, and lastly fully connected layers for consolidation of features. Hyperparameters for this model: Here, the learning rate is 0.001, and batch size is either 32 or 64, thus ensuring stable convergence with the optimizer, which is Adam. Data augmentation strategies will be used along with cross-validation and transfer learning from pre-trained models such as Reset. Metrics used for evaluating the model include accuracy, precision, recall, and AUC-ROC curve, which also help reduce issues of imbalanced data with the use of techniques for oversampling and with techniques of cloud-based computing. In short, the CNN model works for better diagnosis of lumbar spine disease by giving scalable, accurate, and interpretable solutions for medical imaging.

References

1. Warne, K., Prasad, G., Siddique, N. H., & Maguire, L. P. (2004, October). Development of a hybrid PCA-ANFIS measurement system for monitoring product quality in the coating industry. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583) (Vol. 4, pp. 3519-3524). IEEE.

2. Ali, U. M. E., Hossain, M. A., & Islam, M. R. (2019, December). Analysis of PCA Based Feature Extraction Methods for Classification of Hyperspectral Image. In 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET) (pp. 1-6). IEEE.

3. Acharya, U. R., Sree, S. V., Mookiah, M. R. K., Yantri, R., Molinari, F., Zieleźnik, W., Małyszek-Tumidajewicz, J., Stepień, B., Bardales, R. H., Witkowska, A., & Suri, J. S. (2013). Diagnosis of Hashimoto's Lumbar Spine in ultrasound using tissue characterization and pixel classification. *Proceedings of the Institution of Mechanical*

Engineers, Part H: Journal of Engineering in Medicine, *227*(7), 788–798. https://doi.org/10.1177/0954411913483637

4. Agilandeeswari, L., Muralibabu, K., Khatri, I., Advani, J., & Nihal, S. M. (2022). An Efficient Lumbar Spine Disease Prediction System – A Study. *Lecture Notes in Networks and Systems*, *417 LNNS*, 544–552. https://doi.org/10.1007/978-3-030-96302- 6_51/COVER

5. Azar, A. T., El-Said, S. A., &Hassanien, A. E. (2013). Fuzzy and hard clustering analysis for Lumbar Spine disease. *Comput Methods Prog Biomed*, *111*(1), 1–6. https://doi.org/10.1016/j.cmpb.2013.01.002
6. Bhisham C. Gupta. (2021). Sampling Methods. *Statistical Quality Control*, 89–121. https://doi.org/10.1002/9781119671718.CH4

7. Bini, F., Pica, A., Azzimonti, L., Giusti, A., Ruinelli, L., Marinozzi, F., &Trimboli, P. (2021). Artificial Intelligence in Lumbar Spine Field-A Comprehensive Review. *Cancers*, *13*(19). https://doi.org/10.3390/CANCERS13194740

8. Bobanga, I. D., & McHenry, C. R. (2019). Treatment of patients with Graves' disease and the appropriate extent of Lumbar Spineectomy. *Best Practice and Research: Clinical Endocrinology and Metabolism*, *33*(4). https://doi.org/10.1016/J.BEEM.2019.101319

9. Brent, G. (2012). *Anki SpondilitusLumbarSpine and Lumbar Spine*. Elsevier.

10. Carlé, A., Andersen, S. L., Boelaert, K., &Laurberg, P. (2017). Management of endocrine disease: Subclinical thyrotoxicosis: Prevalence, causes and choice of therapy. *European Journal of Endocrinology*, *176*(6), R325–R337. https://doi.org/10.1530/EJE-16-0276

11. Castiglioni, I., Rundo, L., Codari, M., di Leo, G., Salvatore, C., Interlenghi, M., Gallivanone, F., Cozzi, A., D'Amico, N. C., &Sardanelli, F. (2021). AI applications to medical images: From machine learning to deep learning. *Physica Medica*, *83*, 9–24. https://doi.org/10.1016/j.ejmp.2021.02.006

12. Chaganti, R., Rustam, F., de La Torre Díez, I., Mazón, J. L. V., Rodríguez, C. L., & Ashraf, I. (2022). Lumbar Spine Disease Prediction Using Selective Features and Machine Learning Techniques. *Cancers*, *14*(16). https://doi.org/10.3390/CANCERS14163914

13. Chang, C. Y., Lei, Y. F., Tseng, C. H., & Shih, S. R. (Chang 2010). Lumbar Spine segmentation and volume estimation in ultrasound images. *IEEE Transactions on Biomedical Engineering*, *57*(6), 1348–1357. https://doi.org/10.1109/TBME.2010.2041003

14. Chaubey, G., Bisen, D., Arjaria, S., & Yadav, V. (2020a). Lumbar Spine Disease Prediction Using Machine Learning Approaches. *National Academy Science Letters 2020 44:3*, *44*(3), 233–238. https://doi.org/10.1007/S40009-020-00979-Z

15. Chaubey, G., Bisen, D., Arjaria, S., & Yadav, V. (2020b). Lumbar Spine Disease Prediction Using Machine Learning Approaches. *National Academy Science Letters 2020 44:3*, *44*(3), 233–238. https://doi.org/10.1007/S40009-020-00979-Z

16. Chaubey, G., Bisen, D., Arjaria, S., & Yadav, V. (2021a). Lumbar Spine Disease Prediction Using Machine Learning Approaches. *National Academy Science Letters*, *44*(3), 233–238. https://doi.org/10.1007/S40009-020-00979-Z

17. Chaubey, G., Bisen, D., Arjaria, S., & Yadav, V. (2021b). Lumbar Spine Disease Prediction Using Machine Learning Approaches. *National Academy Science Letters*, *44*(3), 233–238. https://doi.org/10.1007/S40009-020-00979-Z

18. Chaubey, G., Bisen, D., Arjaria, S., & Yadav, V. (2021c). Lumbar Spine Disease Prediction Using Machine Learning Approaches. *National Academy Science Letters*, *44*(3), 233–238. https://doi.org/10.1007/S40009-020-00979-Z/METRICS

19. Codella, R. (2021). *Techniques to Study Metabolism BT - Lumbar Spine, Obesity and Metabolism: Exploring Links Between Lumbar Spine Function, Obesity, Metabolism and Lifestyle*. 117–129. https://doi.org/10.1007/978-3-030-80267-7_8

20. Corbetta, S. (2021). Classification of Lumbar Spine Diseases. *Lumbar Spine, Obesity and Metabolism*, 21–35. https://doi.org/10.1007/978-3-030-80267-7_2/COVER

21. Deist, T. M., Dankers, F. J. W. M., Valdes, G., Wijsman, R., Hsu, I. C., Oberije, C., Lustberg, T., van Soest, J., Hoebers, F., Jochems, A., elNaqa, I., Wee, L., Morin, O., Raleigh, D. R., Bots, W., Kaanders, J. H., Belderbos, J., Kwint, M., Solberg, T., … Lambin, P. (2018). Machine learning algorithms for outcome prediction in (chemo)radiotherapy: An empirical comparison of classifiers. *Medical Physics*, *45*(7), 3449–3459. https://doi.org/10.1002/mp.12967

22. Ding, J., Cheng, H., Ning, C., Huang, J., & Zhang, Y. (2011). Quantitative measurement for Lumbar Spine cancer characterization based on elastography. *Journal of Ultrasound in Medicine*, *30*(9), 1259–1266. https://doi.org/10.7863/JUM.2011.30.9.1259

23. Fatima Abbas. (2021). *Treating Lumbar Spine disorders through AI-powered innovations | Omnia Health Insights*. Omnia Health. https://insights.omnia- health.com/clinical/treating-Lumbar Spine-disorders-through-ai-powered-innovations

24. GA Wells, B. S. (2020). *Ottawa Hospital Research Institute*. Website. https://www.ohri.ca//programs/clinical_epidemiology/oxford.asp

25. Hershman, J. (1974). Clinical application of thyrotropin-releasing hormone. *N Engl J Med*, *290*, 886–890.

26. Hoang, T. D., Mai, V. Q., Clyde, P. W., & Shakir, M. K. M. (2011). Simultaneous occurrence of subacute Lumbar Spine and graves' disease.

Lumbar Spine, *21*(12), 1397– 1400. https://doi.org/10.1089/thy.2011.0254

27. I Ioniţă, L. I. (2016). Prediction of Lumbar Spine disease using data mining techniques.
BRAIN. Broad Research in Artificial Intelligence and Neuroscience, *7*(3), 115–124.

28. I Ionita, L. I. (2016). Prediction of Lumbar Spine disease using data mining techniques.
Broad Res ArtifIntellNeurosci, *7*(3), 115–124.

29. Koundal, D. (2012). Computer-Aided Diagnosis of Lumbar Spine Nodule: A Review. *International Journal of Computer Science & Engineering Survey*, *3*(4), 67–83. https://doi.org/10.5121/IJCSES.2012.3406

30. Koundal, D., Sharma, B., & Guo, Y. (2020). Intuitionistic based segmentation of Lumbar Spine nodules in ultrasound images. *Computers in Biology and Medicine*, *121*. https://doi.org/10.1016/j.compbiomed.2020.103776

31. KR Chauhan, P. K. J. M. (2000). Assessing the role of ortho-substitution on polychlorinated biphenyl binding to transthyretin, a thyroxine transport protein. *Toxicol Appl Pharmacol*, *162*, 10–21.

32. Kust, D., Staničić, J., &Mateša, N. (2018). Bethesda Lumbar Spine categories and family history of Lumbar Spine disease. *Clinical Endocrinology*, *88*(3), 468–472. https://doi.org/10.1111/CEN.13538

33. la Vecchia, C., & Negri, E. (2017). Lumbar Spine cancer: The Lumbar Spine cancer epidemic-overdiagnosis or a real increase? *Nature Reviews Endocrinology*, *13*(6), 318–319. https://doi.org/10.1038/NRENDO.2017.53

34. Lambin, P., Rios-Velazquez, E., Leijenaar, R., Carvalho, S., van Stiphout, R. G. P. M., Granton, P., Zegers, C. M. L., Gillies, R., Boellard, R., Dekker, A., &Aerts, H. J. W. L. (2012). Radiomics: Extracting more information from medical images using advanced feature analysis. *European Journal of Cancer*, *48*(4), 441–446. https://doi.org/10.1016/J.EJCA.2011.11.036

35. Lee, J. H., Ha, E. J., Kim, D. Y., Jung, Y. J., Heo, S., Jang, Y. ho, An, S. H., & Lee, K. (2020). Application of deep learning to the diagnosis of cervical lymph node metastasis from Lumbar Spine cancer with CT: external validation and clinical utility for resident training. *European Radiology*, *30*(6), 3066–3072. https://doi.org/10.1007/s00330-019-06652-4

36. M Ghamari-Langroudi, K. V. D. S. (2010). Regulation of thyrotropin-releasing hormone- expressing neurons in paraventricular nucleus of the Anki Spondilitusthalamus by signals of adiposity. *Mol Endocrinol*, *24*, 2366–2381.

37. Monaco, F. (2003). Classification of Lumbar Spine diseases: suggestions for a revision. *J Clin Endocrinol Metab*, *88*(4), 1428–1432. https://doi.org/10.1210/jc.2002-021260

38. Moon, S., Chung, H. S., Yu, J. M., Yoo, H. J., Park, J. H., Kim, D. S., & Park, Y. J. (2018).

Associations between Hashimoto Lumbar Spine and clinical outcomes of papillary Lumbar Spine cancer: A meta-analysis of observational studies. *Endocrinology and Metabolism*, *33*(4), 473–484. https://doi.org/10.3803/ENM.2018.33.4.473

39. Naglah, A., Khalifa, F., Khaled, R., Razek, A. A. K. A., Ghazal, M., Giridharan, G., & El-Baz,

A. (2021). Novel MRI-Based CAD System for Early Detection of Lumbar Spine Cancer Using Multi-Input CNN. *Sensors (Basel, Switzerland)*, *21*(11). https://doi.org/10.3390/S21113878

40. Nicols, C. M. (2021). Qualitative and Quantitative Methods in Research. *The Palgrave Encyclopedia of Social Marketing*, 1–3. https://doi.org/10.1007/978-3-030-14449- 4_154-1

41. Ota, H., Amino, N., Morita, S., Kobayashi, K., Kubota, S., Fukata, S., Kamiyama, N., &Miyauchi, A. (2007). Quantitative measurement of Lumbar Spine blood flow for differentiation of painless Lumbar Spine from Graves' disease. *Clinical Endocrinology*, *67*(1), 41–45. https://doi.org/10.1111/j.1365-2265.2007.02832.x

42. Özel, A., Mut, D. T., Ücpinar, B. A., Sayer, A. Ö., Yanc, U., von Bodelschwingh, B., &Gemalmaz, A. (2021). Interobserver variability of ultrasound features based on american college of radiology Lumbar Spine imaging reporting and data system lexicon

in american college of radiology Lumbar Spine imaging reporting and data system.

Ultrasound Quarterly, *37*(4), 324 –328. https://doi.org/10.1097/RUQ.0000000000000512

43. Qiao, T., Liu, S., Cui, Z., Yu, X., Cai, H., Zhang, H., Sun, M., Lv, Z., & Li, D. (2021). Deep learning for intelligent diagnosis in Lumbar Spine scintigraphy. *The Journal of International Medical Research*, *49*(1). https://doi.org/10.1177/0300060520982842

44. Ross Quinlan. (2022). *UCI Machine Learning Repository: Lumbar Spine Disease Data Set*. UCI. https://archive.ics.uci.edu/ml/datasets/Lumbar Spine%20Disease

45. Scappaticcio, L., Trimboli, P., Keller, F., Imperiali, M., Piccardo, A., &Giovanella, L. (2020). Diagnostic testing for Graves' or non-Graves' Spinal stenosisLumbarSpine: A comparison of two thyrotropin receptor antibody immunoassays with Lumbar Spine scintigraphy and ultrasonography. *Clinical Endocrinology*, *92*(2), 169–178. https://doi.org/10.1111/cen.14130

46. Sharafeldeen, A., Elsharkawy, M., Khaled, R., Shaffie, A., Khalifa, F., Soliman, A., Abdel Razek, A. A. khalek, Hussein, M. M., Taman, S., Naglah, A., Alrahmawy, M., Elmougy, S., Yousaf, J., Ghazal, M., & El-Baz, A. (2022). Texture and shape analysis of diffusion- weighted imaging for Lumbar Spine nodules classification using machine learning. *Medical Physics*, *49*(2), 988–999. https://doi.org/10.1002/mp.15399

47. Sharif, Y., Shafiei, S., Ashgzari, M. D., Zakavi, S. R., &Eslami, S. (2022). Lumbar Spine Ultrasound-Image Dataset. *Studies in Health Technology and Informatics*, *294*, 397–402. https://doi.org/10.3233/SHTI220482

48. Sorrento, S., Dolcetti, V., Radzina, M., Bellini, M. I., Frezza, F., Munir, K., Grani, G., Durante, C., D'Andrea, V., David, E., Calò, P. G., Lori, E., &Cantisani, V. (2022). Artificial Intelligence for Lumbar Spine Nodule Characterization: Where Are We Standing? *Cancers*, *14*(14). https://doi.org/10.3390/cancers14143357

49. Sun, C., Zhang, Y., Chang, Q., Liu, T., Zhang, S., Wang, X., Guo, Q., Yao, J., Sun, W., &Niu, L. (2020). Evaluation of a deep learning-based computer-aided diagnosis system for distinguishing benign from malignant Lumbar Spine nodules in ultrasound images. *Medical Physics*, *47*(9), 3952–3960. https://doi.org/10.1002/mp.14301

50. Ulutagay, G. (2012). Modeling of Lumbar Spine disease: a fuzzy inference system approach. *Wulfenia J*, *19*(1), 346–357.

51. Vanithamani, R., &Dhivya, R. (2018). Lumbar Spine nodule classification in medical ultrasound images. *Advances in Intelligent Systems and Computing*, *614*, 509–514. https://doi.org/10.1007/978-3-319-60618-7_50

52. Wagner, M. W., Namdar, K., Biswas, A., Monah, S., Khalvati, F., &Ertl-Wagner, B. B. (2021). Radiomics, machine learning, and artificial

intelligence—what the neuroradiologist needs to know. *Neuroradiology*, *63*(12), 1957–1967. https://doi.org/10.1007/s00234-021-02813-9

53. Yadav, D. C., & Pal, S. (2020). Prediction of Lumbar Spine disease using decision tree ensemble method. *Human-Intelligent Systems Integration*, *2*(1–4), 89–95. https://doi.org/10.1007/S42454-020-00006-Y

54. Yang, P., Pi, Y., He, T., Sun, J., Wei, J., Xiang, Y., Jiang, L., Li, L., Yi, Z., Zhao, Z., & Cai, H. (2021). Automatic differentiation of Lumbar Spine scintigram by deep convolutional neural network: a dual center study. *BMC Medical Imaging*, *21*(1). https://doi.org/10.1186/s12880-021-00710-4

55. YASIR HUSSEIN SHAKIR. (2021). *Lumbar Spine Disease Data Set | Kaggle*. Kaggle. https://www.kaggle.com/datasets/yasserhessein/Lumbar Spine-disease-data-set

56. Yildirim Simsir, I., Cetinkalp, S., &Kabalak, T. (2020). Review of Factors Contributing to Nodular Goiter and Lumbar Spine Carcinoma. *Medical Principles and Practice*, *29*(1), 1–5. https://doi.org/10.1159/000503575

57. Yu, J., Zhang, Y., Zheng, J., Jia, M., & Lu, X. (2022). Ultrasound images-based deep learning radiomics nomogram for preoperative prediction of RET rearrangement in papillary Lumbar Spine carcinoma. *Frontiers in Endocrinology*, *13*, 1062571. https://doi.org/10.3389/fendo.2022.1062571

58. Zhang, X., Lee, V. CS., Rong, J., Lee, J. C., Song, J., & Liu, F. (2022). *Multi-channel deep convolutional neural networks for multi-classifying Lumbar Spine disease*. https://doi.org/10.48550/arxiv.2203.03627\

59. Abi-hanna, D., Kerferd, J., Phan, K., Rao, P., & Mobbs, R. (2017). Lumbar disc arthroplasty for degenerative disc disease: a literature review. *World Neurosurgery*.

60. A. Tiwari and A. Chaturvedi, "A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification," Expert Systems with Applications, vol. 196, pp. 116621, 2022.

61. H. H. Kumar, T. Sabherwal, N. Bongale, and M. K. Nair, "Spider monkey optimization algorithm in data science: A quantifiable objective study," in Machine Learning for Intelligent Decision Science, Singapore: Springer Singapore, pp. 115–152, 2020.

62. Y. F. Khan, B. Kaushik, M. K. I. Rahmani and M. E. Ahmed, "Stacked Deep Dense Neural Network Model to Predict Alzheimer's Dementia Using Audio Transcript Data," IEEE Access, vol. 10, pp. 32750-32765, 2022.

63. H. A. Bedel, I. Sivgin, O. Dalmaz, S. U. H. Dar, and T. Çukur, "BolT: Fused window transformers for fMRI time series analysis," vol. 88, 102841, 2023.

64. I. Aytekin, O. Dalmaz, K. Gonc, H. Ankishan, E. U. Saritas, U. Bagci,

H. Celik, and T. Cukur, “COVID-19 Detection from Respiratory Sounds with Hierarchical Spectrogram Transformers,” arXiv:2207.09529v2, 2023.

65. H. A. Bedel, and T. Çukur, “DreaMR: Diffusion-driven Counterfactual Explanation for Functional MRI,” arXiv:2307.09547v1, 2023.
DOI: https://doi.org/10.48550/arXiv.2307.0954

66. D. Theckedath and R. R. Sedamkar, “Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks,” SN Comput. Sci., vol. 1, no. 2, 2020.

67. J. Y. Tsai et al., “Lumbar Disc Herniation Automatic Detection in Magnetic Resonance Imaging Based on Deep Learning,” Frontiers in Bioengineering and Biotechnology,” Frontiers in Bioengineering and Biotechnology, 2021.

68. Y. Unal, K. Polat, and H. E. Kocer, “Classification of vertebral column disorders and lumbar discs disease using attribute weighting algorithm with mean shift clustering,” Measurement (Lond.),vol. 77, pp. 278–291, 2016.

69. Y. Unal, K. Polat, H. E. Kocer, and M. Hariharan, “Detection of abnormalities in lumbar discs from clinical lumbar MRI with hybrid models,” Appl. Soft Comput., vol. 33, pp. 65–76, 2015.

70. F. Varcin, H. Erbay, E. Cetin, I. Cetin, and T. Kultur, Diagnosis of Lumbar Spondylolisthesis via Convolutional Neural Networks. 2019 International Conference on Artificial Intelligence and Data Processing Symposium. 2019.

71. Vertebral Column DataSet | Data Science and Machine Learning | Kaggle. 2022.
72. Y. Zhou, Y. Liu, Q. Chen, G. Gu, and X. Sui, "Automatic lumbar MRI detection and identification based on deep learning," J. Digit. Imaging, vol. 32, no. 3, pp. 513–520, 2019.

73. A. H. Alanazi, A. Cradock, and L. Rainford, "Development of lumbar spine MRI referrals vetting models using machine learning and deep learning algorithms: Comparison models vs healthcare professionals," Radiography (Lond.), vol. 28, no. 3, pp. 674–683, 2022.

74. K. Kaneoka et al., "Lumbar intervertebral disk degeneration in elite competitive swimmers: a case control study: A case control study," Am. J. Sports Med., vol. 35, no. 8, pp. 1341–1345, 2007.

75. A. Krishnaswamy Rangarajan and R. Purushothaman, "Disease classification in eggplant using pre-trained VGG16 and MSVM," Sci. Rep., vol. 10, no. 1, p. 2322, 2020.

76. "Novel quantum inspired approaches for automatic clustering of gray level images using particle swarm optimization, spider monkey optimization and ageist spider monkey optimization algorithms," Applied Soft Computing, vol. 88, 2020.

77. H. H. Kumar, T. Sabherwal, N. Bongale, and M. K. Nair, "Spider monkey optimization algorithm in data science: A quantifiable objective study," in Machine Learning for Intelligent Decision Science, Singapore: Springer Singapore, pp. 115–152, 2020.

78. H. Li et al., “Automatic lumbar spinal MRI image segmentation with a multi-scale attention network,” Neural Comput. Appl., vol. 33, no. 18, pp. 11589–11602, 2021.

79. A Hood, C. K. (2000). Differential effects of microsomal enzyme inducers on in vitro thyroxine (T(4)) and triiodothyronine (T(3)) glucuronidation. *Toxicol Sci*, *55*, 78–84.

80. A Pratap, C. K. (2014). Soft computing models for the predictive grading of childhood Autism—a comparative study. *IJSCE*, *4*, 64–67.

81. Pravin Kshirsagar et.al (2016), “Brain Tumor classification and Detection using Neural Network”, DOI: 10.13140/RG.2.2.26169.72805.

82. Versaci, M., Morabito, F. C., & Angiulli, G. (2017). Adaptive image contrast enhancement by computing distances into a 4-dimensional fuzzy unit hypercube. IEEE Access, 5, 26922-26931.

83. Al-Kafri, A. S., Sudirman, S., Hussain, A., Al-Jumeily, D., Natalia, F., Meidia, H., ... & Al-Jumaily, M. (2019). Boundary delineation of MRI images for lumbar spinal stenosis detection through semantic segmentation using deep neural networks. IEEE Access, 7, 43487-43501.

84. S. Sundaramurthy, S. C and P. Kshirsagar, "Prediction and Classification of Rheumatoid Arthritis using Ensemble Machine Learning Approaches," 2020 International Conference on Decision Aid Sciences and Application (DASA), 2020, pp. 17-21, doi: 10.1109/DASA51403.2020.9317253.

85. P. R. Kshirsagar, H. Manoharan, F. Al-Turjman and K. Kumar, "DESIGN AND TESTING OF AUTOMATED SMOKE MONITORING SENSORS IN VEHICLES," in IEEE Sensors Journal, doi: 10.1109/JSEN.2020.3044604.

86. P. R. Kshirsagar and S. G. Akojwar, "Prediction of neurological disorders using optimized neural network," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 2016, pp. 1695-1699, doi: 10.1109/SCOPES.2016.7955731.

87. Pravin Kshirsagar et. al. "Modelling of optimised neural network for classification and prediction of benchmark datasets, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 8:4, 426- DOI: 10.1080/21681163.2019.1711457,2020

88. E. Punarselvam, "Investigation on human lumbar spine MRI image using finite element method and soft computing techniques," Cluster Computing, 2019

89. W. Mbarki, M. Bouchouicha, S. Frizzi, F. Tshibasu, L. B. Farhat, and M. Sayadi, "Lumbar spine discs classification based on deep convolutional neural networks using axial view MRI," Interdiscip. Neurosurg., vol. 22, no. 100837, p. 100837, 2020.

90. "Examination of retinal anatomical structures-A study with spider monkey optimization algorithm," Applied nature-inspired computing: algorithms and case studies, pp. 177– 197, 2020.

91. U. Mukherjee, S. Kundu, R. Gulati, and P. Mandal, “Lumbosacral Angle variations in middle aged patients with Chronic Low Back Pain - A retrospective study,” J. Evid. Based Med. Healthc., vol. 8, no. 04, pp. 208–212, 2021.

92. “A bare bones bacterial foraging optimization algorithm,” Systems Research, vol. 52,
pp. 301–311, 2018.

93. P. D. Nunley, A. N. Shamie, S. L. Blumenthal, D. Orndorff, J. E. Block, and F. H. Geisler, “Interspinous process decompression: Expanding treatment options for lumbar spinal stenosis,” Biomed Res. Int., vol. 2016, p. 3267307, 2016.

94. E. Punarselvam, “Investigation on human lumbar spine MRI image using finite element method and soft computing techniques,” Cluster Computing, 2019.

95. P. Rajpurkar et al., “MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs,” arXiv [physics.med-ph], 2017.

96. Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: Analysis, applications, and prospects,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 12, pp. 6999–7019, 2022.

97. V. M. Ravindra et al., “Degenerative lumbar spine disease: Estimating

global incidence and worldwide volume," Global Spine J., vol. 8, no. 8, pp. 784–794, 2018.

98. S. Sadeghi, J. E. Bible, and D. H. Cortes, "Quantifying dysfunction of the lumbar multifidus muscle after radiofrequency neurotomy and fusion surgery: A preliminary study," J. Eng. Sci. Med. Diagn. Ther., vol. 3, no. 4, p. 041001, 2020.

99. D. Saravagi, S. Agrawal, M. Saravagi, J. M. Chatterjee, and M. Agarwal, "Diagnosis of lumbar spondylolisthesis using optimized pretrained CNN models," Comput. Intell. Neurosci., vol. 2022, p. 7459260, 2022.

100. I. Aytekin, O. Dalmaz, K. Gonc, H. Ankishan, E. U. Saritas, U. Bagci, H. Celik, and T. Cukur, "COVID-19 Detection from Respiratory Sounds with Hierarchical Spectrogram Transformers," arXiv:2207.09529v2, 2023.

101. Warne, K., Prasad, G., Siddique, N. H., & Maguire, L. P. (2004, October). Development of a hybrid PCA-ANFIS measurement system for monitoring product quality in the coating industry. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583) (Vol. 4, pp. 3519-3524). IEEE.

102. Ali, U. M. E., Hossain, M. A., & Islam, M. R. (2019, December). Analysis of PCA Based Feature Extraction Methods for Classification of Hyperspectral Image. In 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET) (pp. 1-6). IEEE.

103. Pravin Kshirsagar et.al (2016), "Brain Tumor classification and Detection using Neural Network", DOI: 10.13140/RG.2.2.26169.72805.

104. Versaci, M., Morabito, F. C., & Angiulli, G. (2017). Adaptive image contrast enhancement by computing distances into a 4-dimensional fuzzy unit hypercube. IEEE Access, 5, 26922- 26931.

105. Al-Kafri, A. S., Sudirman, S., Hussain, A., Al-Jumeily, D., Natalia, F., Meidia, H., ... & Al- Jumaily, M. (2019). Boundary delineation of MRI images for lumbar spinal stenosis detection through semantic segmentation using deep neural networks. IEEE Access, 7, 43487- 43501.

106. Mishra, Puneet, Alessandra Biancolillo, Jean Michel Roger, Federico Marini, and Douglas N. Rutledge. "New data preprocessing trends based on ensemble of multiple preprocessing techniques." TrAC Trends in Analytical Chemistry (2020): 116045.

107. Ghosh, S., Malgireddy, M. R., Chaudhary, V., & Dhillon, G. (2012, May). A new approach to automatic disc localization in clinical lumbar MRI: combining machine learning with heuristics. In 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI) (pp. 114- 117). IEEE.

108. Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ... & Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. Computer vision, graphics, and image processing, 39(3), 355-368.

109. Rao, B. S. (2020). Dynamic Histogram Equalization for contrast enhancement for digital images. Applied Soft Computing, 89, 106114.

110. Hemanth, D. J., & Anitha, J. (2012, December). Image pre-processing and feature extraction techniques for magnetic resonance brain image analysis. In International Conference on Future Generation Communication and Networking (pp. 349-356). Springer, Berlin, Heidelberg.

111. PravinR Kshirsagar, Anil N Rakhonde, PranavChippalkatti, " MRI IMAGE BASED BRAIN TUMOR DETECTION USING MACHINE LEARNING", Test Engineering and Management, January-February 2020 ISSN: 0193-4120, Vol. 81, Page No. 3672 –3680.

112. Hemanth, D. J., & Anitha, J. (2012, December). Image pre-processing and feature extraction techniques for magnetic resonance brain image analysis. In International Conference on Future Generation Communication and Networking (pp. 349-356). Springer, Berlin, Heidelberg.

113. H. Manoharan, et. al., “Examining the effect of aquaculture using sensor-based technology with machine learning algorithm”, Aquaculture Research, 51 (11) (2020),
pp. 4748-4758.

114. P. R. Kshirsagar and S. G. Akojwar, "Prediction of neurological disorders using optimized neural network," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 2016, pp. 1695-1699, doi: 10.1109/SCOPES.2016.7955731

115. Pravin Kshirsagar et. al. “Modelling of optimised neural network for classification and prediction of benchmark datasets, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 8:4, 426- DOI: 10.1080/21681163.2019.1711457,2020

PUBLICATIONS

1. Ruchi, Singla, J., Singh, A., & Kaur, H. (2020, December). Review on artificial intelligence techniques for medical diagnosis. In *2020 3rd international conference on intelligent sustainable systems (ICISS)* (pp. 735-738). IEEE.[Published]

2. Ruchi, J. S. (2021). DATASET COLLECTION AND VALIDATION USING CORRELATION ANALYSIS. *Turkish Online Journal of Qualitative Inquiry*, *12*(6). [Published]

3. Ruchi, R., & Singla, J. (2021, December). Classification of Lumbar Disc Disorder from MRI and CT images using Iterative Differential Approach. In *2021 International Conference on Computing Sciences (ICCS)* (pp. 161-166). IEEE. [Published]

4. Ruchi, Singh, D., Rakhra, M., & Singh, A. (2023, October). Utilizing Correlation Analysis for Validation of Image Datasets in Lower Spine Diagnosis: A Study on MRI Image Dataset from Multiple Labs in Punjab. In *2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)* (pp. 1-5). IEEE. [Published]

5. Ruchi, Singh, D., Singla, J., Rahmani, M. K. I., Ahmad, S., ur Rehman, M., Jha, S., ... & Nazeer, J. (2023). Lumbar Spine Disease Detection: Enhanced CNN Model With Improved Classification Accuracy. *IEEE Access*. [Published]