VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Belagavi, Karnataka-590 018

A FINAL YEAR PROJECT REPORT ON

"DETECTION OF SPINE INJURIES"

Submitted in partial fulfillment of the requirements of the Award of the degree

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
Submitted By:

Bhuvan U Kadlas	**[4VV20CS020]**
Dheemanth Gowda S M	**[4VV20CS034]**
Abhishek L	**[4VV20CS004]**
D S Yashwanth	**[4VV20CS026]**

UNDER THE GUIDANCE OF

Dr. Chethana H T

Associate Professor

Dept. of CS&E

VVCE

2023-24

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VIDYAVARDHAKA COLLEGE OF ENGINEERING

MYSURU-570002

VIDYAVARDHAKA COLLEGE OF ENGINEERING

GOKULAM III STAGE, MYSORE- 570 002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VVCE

CERTIFICATE

This is to certify that the final year project report entitled **"Detection of Spine Injuries"** is a bonafide work carried out by **Bhuvan U Kadlas** (4VV20CS020), **Dheemanth Gowda S M** (4VV20CS034), **Abhishek L** (4VV20CS004) and **D S Yashwanth** (4VV20CS026) students of 8th-semester Computer Science and Engineering, **Vidyavardhaka College of Engineering, Mysuru** in partial fulfillment for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering** of the **Visvesvaraya Technological University, Belagavi**, during the academic year **2023-2024**. It is certified that all the suggestions and corrections indicated for the internal assessment have been incorporated in the report deposited in the department library. The report has been approved as it satisfies the requirements in respect of project work prescribed for the said degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
______________________	______________________	______________________
(Dr. Chethana H T)	**(Dr. Pooja M R)**	**(Dr. B Sadashive Gowda)**

Name of the Examiners **Signature with Date**

1)

2)

ACKNOWLEDGEMENT

If words are considered as the tokens of acknowledgment, then the words play the heralding role of expressing our gratitude.

With proud gratitude we thank God Almighty for all the blessings showered on us and for completing our project successfully.

We owe our gratitude to The Principal, Dr. B Sadashive Gowda for his whole hearted support and for his kind permission to undergo the project.

We wish to express our deepest gratitude to **Dr. Pooja M R**, Professor and Head, of the Department of Computer Science and Engineering, VVCE, for her valuable suggestions and guidance throughout this report.

We wish to enunciate our special thanks to our paradigmatic and relevant Project coordinators Dr. Ayesha Taranum Associate Professor, Gayana J Kumar Assistant Professor, Kavitha D N Assistant Professor Harshitha Suresh Assistant Professor and internal guide, **Dr. Chethana H T** Associate Professor, Department of Computer Science and Engineering, VVCE, Mysuru for her encouragement and advice throughout the course of the project work. My numerous discussions with her were extremely helpful. I hold her in esteem for the guidance, encouragement, and inspiration received from her.

In the end, we are eager to offer our sincere thanks to our family members and friends for their valuable suggestions and encouragement.

Bhuvan U Kadlas **4VV20CS020**
Dheemanth Gowda S M **4VV20CS034**
Abhishek L **4VV20CS004**
D S Yashwanth **4VV20CS026**

ABSTRACT

Spinal injuries pose significant diagnostic and therapeutic hurdles due to their correlation with debilitating conditions such as myelitis, an inflammation of the spinal cord requiring prompt identification to prevent irreversible neurological damage. This survey paper explores the efficiency of Deep Learning (DL) models in detecting myelitis within the context of spine injuries. Drawing upon diverse clinical records and imaging data sourced from Kaggle datasets, Deep Learning algorithms are used accurately to predict the presence of myelitis. These datasets encompass a comprehensive set of patient demographics, injury types and clinical presentations, providing a robust foundation for model training and validation. This research survey delves into various Deep Learning architectures and methodologies, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models, addressing preprocessing techniques, feature extraction, and model optimization strategies tailored to the unique challenges inherent in spine injury datasets. Furthermore, assessment of various performance metrics of these Deep Learning models, comparing sensitivity, specificity, and accuracy against traditional diagnostic methods. This survey aims to furnish a comprehensive resource for researchers and healthcare practitioners interested in deploying Deep Learning techniques for spine injury diagnosis, specifically in the realm of myelitis detection, with the overarching objective of enhancing patient outcomes and streamlining clinical decision-making processes. Comparative study of detection of spine injuries with the existing works are discussed. It is observed that CNN provides a recognition accuracy of 92 percent and outperforms better than existing algorithms for myelitis detection. The detection model shows an accuracy of 96.62% in detecting the myelitis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Spinal injuries, encompassing conditions such as myelitis, pose substantial diagnostic challenges due to their potential for irreversible neurological damage and debilitating consequences. Myelitis, characterized by inflammation of the spinal cord, necessitates prompt identification to optimize patient outcomes and prevent long-term disability. Traditional diagnostic methods, including clinical assessments and standard imaging techniques, often face limitations in accurately and efficiently detecting myelitis, leading to delayed interventions and compromised patient care. Moreover, the intricate nature of myelitis, influenced by various factors such as infectious agents, autoimmune processes, genetic predisposition, and environmental triggers, underscores the need for advanced computational techniques to enhance diagnostic accuracy and streamline clinical decision-making processes.

In recent years, deep learning (DL) models have emerged as promising tools for addressing the diagnostic complexities associated with myelitis detection in spinal injuries. DL algorithms, a subset of artificial intelligence (AI) methodologies, offer the potential to autonomously learn and discern complex patterns and features from extensive medical imaging data, surpassing traditional manual feature extraction methods. By leveraging diverse clinical records and imaging datasets, DL models can accurately predict the presence of myelitis, enabling earlier interventions and more effective management of spinal injuries linked to myelitis. However, despite the potential benefits of DL-based approaches, there remains a need for comprehensive research and exploration into the efficiency, reliability, and applicability of these models in real-world clinical settings.

Therefore, the primary aim of this research project is to conduct a comprehensive review and analysis of DL techniques for myelitis detection in the context of spinal injuries. This project seeks to explore the efficiency of various DL architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models.

1.1.1 Myelitis

Myelitis is a rare but serious neurological condition characterized by inflammation of the spinal cord. This inflammation disrupts the normal functioning of the spinal cord, leading to a wide range of symptoms depending on the severity and location of the inflammation. Common symptoms of myelitis include weakness or paralysis of the limbs, sensory disturbances such as numbness or tingling, bladder and bowel dysfunction, pain, and in severe cases, respiratory complications.

There are several potential causes of myelitis, including viral infections such as herpes simplex virus, varicella-zoster virus, and enteroviruses; autoimmune disorders such as multiple sclerosis and neuromyelitis optical bacterial infections such as tuberculosis and syphilis; and non-infectious inflammatory conditions like transverse myelitis.

Diagnosis of myelitis typically involves a thorough medical history, physical examination, and diagnostic tests such as MRI scans, lumbar puncture (spinal tap), and blood tests to identify the underlying cause of the inflammation.

Treatment for myelitis depends on the underlying cause and severity of symptoms. In cases of viral myelitis, antiviral medications may be prescribed to reduce viral replication and inflammation. For autoimmune-related myelitis, corticosteroids and immunosuppressive therapies are often used to suppress the immune response and prevent further damage to the spinal cord. Physical therapy and rehabilitation are also important components of treatment, helping patients regain strength, mobility, and function.

Myelitis is considered a rare condition, but its exact prevalence is challenging to determine due to its various causes and presentations. Certain risk factors may predispose individuals to develop myelitis, including a history of v The prognosis for myelitis varies depending on the cause, extent of spinal cord damage, and promptness of treatment. Early detection and intervention are crucial for optimizing patient outcomes and minimizing long-term disability. However, some individuals may experience persistent symptoms or complications despite treatment.

Myelitis can manifest in different forms depending on the underlying cause and location of inflammation. For example, longitudinally extensive transverse myelitis (LETM) involves inflammation extending over three or more vertebral segments, while acute flaccid myelitis (AFM) predominantly affects the gray matter of the spinal cord, leading to sudden muscle weakness or paralysis.

Ongoing research is exploring novel treatment strategies for myelitis, including targeted immunotherapies, stem cell transplantation, and neuroprotective agents. Clinical trials are underway to evaluate the efficacy and safety of these interventions in improving outcomes for individuals with myelitis

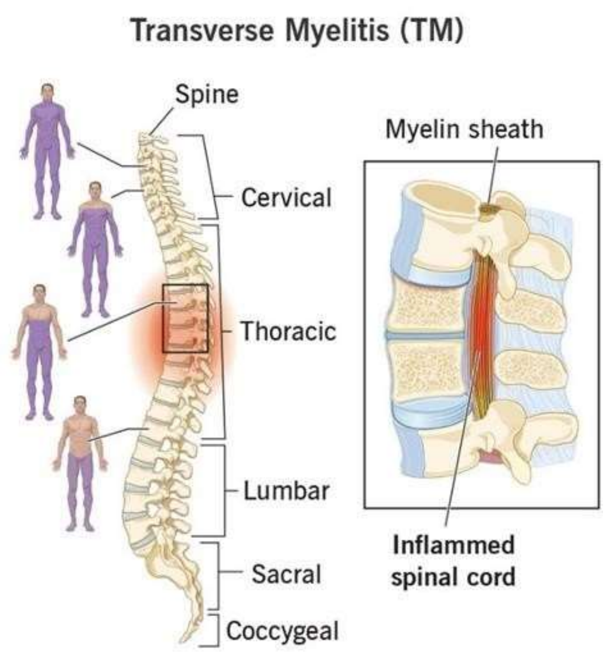

Fig 1.1: Myelitis

The above figure 1.1 depicts the bisectional view of the spinal cord.

1.1.2 Factors Affecting Myelitis

Myelitis, characterized by spinal cord inflammation, can result from various influences including infectious agents, autoimmune processes, genetic predisposition, environmental triggers, and demographic factors. Understanding these conditions is the vital role for comprehending myelitis mechanism and developing effective treatment and prevention strategies.

- Infectious Agents: Human T-cell lymphotropic virus (HTLV), herpes simplex virus, varicella-zoster virus, enteroviruses, TB, Lyme disease, and bacteria can all cause inflammation of the spinal cord. These pathogens can harm tissues and disrupt neurological function either by directly infecting the spinal cord or by inducing an immune response.
- Autoimmune Processes: The central nervous system, especially the spinal cord, is the focus of immune system dysfunction in disorders such as multiple sclerosis (MS), neuromyelitis Optica (NMO), acute
- Disseminated encephalomyelitis (ADEM), and transverse myelitis. Demyelination and inflammation can occur in autoimmune myelitis when immune cells and cytokines target myelin, axons, or other parts of the nervous system.
- Genetic Predisposition: Genetic factors influence susceptibility to myelitis and spinal cord-affecting autoimmune diseases. Certain genetic variations contribute to the risk of conditions like MS, NMO, and hereditary spastic paraplegia, with myelitis as a primary symptom.
- Environmental Triggers: Factors such as viral infections, toxin exposure, smoking, diet, and stress can impact myelitis development. These triggers may interact with genetics and

immune dysfunction, exacerbating inflammation and influencing disease onset or progression.

- Demographic Characteristics: Factors like age, sex, ethnicity, and location affect myelitis incidence and presentation. Autoimmune disorders like MS and NMO show prevalence variations among different demographic groups. Additionally, disparities in environmental factors and healthcare access may influence disease outcomes and treatment effectiveness.
- Neurological and Systemic Diseases: Underlying neurological conditions, systemic autoimmune disorders, and comorbidities can heighten myelitis risk or worsen existing inflammation. Conditions such as systemic lupus erythematosus, sarcoidosis, Behçet's disease, and vasculitis may involve the spinal cord, contributing to myelitis pathogenesis.
- Understanding these influences is crucial for accurate diagnosis, prognosis, and treatment planning in myelitis patients. Multidisciplinary approaches integrating clinical, immunological, genetic, and environmental factors are essential for tailoring therapeutic interventions to individual patient needs. Ongoing research efforts into mechanisms and therapeutic targets offer hope for improving outcomes and quality of life for those with myelitis.

Understanding these factors that contribute to myelitis formation is crucial in adopting preventive measures, such as maintaining proper hydration and dietary modifications.

The paper is organized into the following sections. Section 2 discusses about existing techniques using predictive algorithms to analyze numerous features. Literature Survey is discussed in section 3. Section 4 discusses the advantages of using DL in predicting and detection of myelitis.

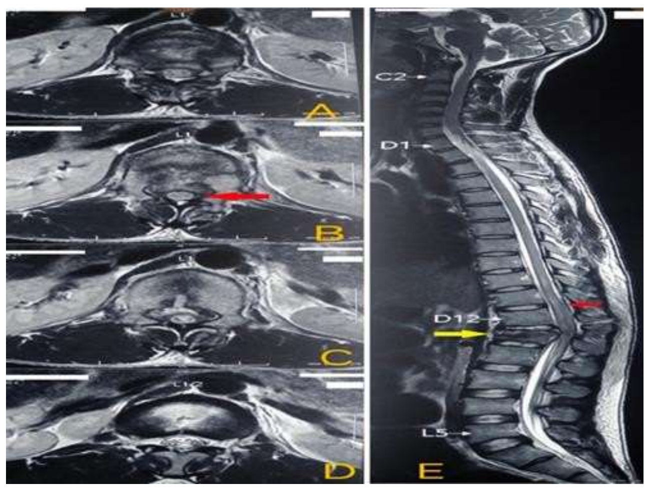

Fig 1.2: Cross-Sectional view of Spine

The above figure depicts the Cross-Sectional view of the Spinal Cord where the myelitis can be detected from the cross-section L1 to L5.

1.2 Motivation

Developing a detection system for Myelitis can be motivated by several factors:

- Clinical Significance: Spinal injuries, particularly those involving myelitis, present significant challenges in diagnosis and management due to their potential for irreversible neurological damage. Improving the accuracy and efficiency of myelitis detection is crucial for optimizing patient outcomes and reducing long-term disability.
- Diagnostic Challenges: Traditional diagnostic methods for myelitis, such as clinical assessments and standard imaging techniques, may face limitations in promptly and accurately identifying spinal cord inflammation. There is a pressing need for advanced computational techniques to enhance diagnostic accuracy and streamline clinical decision-making processes.
- Advancements in Deep Learning: With the rapid advancements in deep learning (DL) technologies, there is increasing interest in leveraging DL models for medical image analysis. DL algorithms offer the potential to autonomously learn and discern complex patterns from extensive medical imaging data, surpassing traditional manual feature extraction methods.
- Potential of DL Models: DL models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated promising results in various medical imaging tasks, including lesion detection and classification. Exploring the potential of DL models for myelitis detection could revolutionize diagnostic practices and improve patient care outcomes.
- Cost Reduction: Myelitis impose a significant economic burden on healthcare systems due to hospitalizations, surgeries, and follow-up treatments. A predictive and detection system could

help in early intervention, potentially reducing the need for expensive procedures and hospital stays, thus saving healthcare costs.

- Automation and Efficiency: By automating the identification of myelitis lesions on medical imaging scans, DL models can reduce the reliance on manual interpretation by radiologists and healthcare providers. This automation can lead to quicker turnaround times, enabling timely interventions for patients suspected of having myelitis.
- Quantitative Analysis: DL models enable quantitative analysis of myelitis lesions, offering objective metrics such as lesion volume, distribution, and spatial extent. This quantitative assessment facilitates longitudinal monitoring of disease progression, treatment response evaluation, and detection of subtle changes in lesion morphology over time.
- Clinical Impact: Implementing DL-based approaches for myelitis detection could have a profound impact on clinical practices, enabling earlier interventions, personalized treatment planning, and continual monitoring of myelitis patients. These advancements have the potential to enhance patient outcomes, minimize neurological deficits, and prevent irreversible spinal cord damage.

1.3 Objectives

- The objectives of a Myelitis detection and prediction model can be multifaceted, aiming to address various aspects of the condition:
- Evaluate DL Techniques: Assess the efficiency and performance of deep learning (DL) techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for detecting myelitis lesions on medical imaging scans.
- Compare with Traditional Methods: Compare the sensitivity, specificity, and accuracy of DL-based myelitis detection models with traditional diagnostic methods, such as clinical assessments and standard imaging techniques, to evaluate the superiority of DL approaches.
- Explore Model Architectures: Investigate different DL model architectures and methodologies, including variations in network depth, convolutional layers, and optimization algorithms, to determine the optimal configuration for myelitis detection in spinal injuries.
- Address Data Preprocessing Challenges: Address challenges related to data preprocessing, including noise reduction, standardization, and feature extraction, to enhance the quality and reliability of input data for DL model training and validation
- Optimize Model Performance: Optimize DL model performance through rigorous validation procedures, parameter tuning, and optimization strategies to ensure high accuracy and reliability in diagnosing spinal cord

- Assess Clinical Impact: Evaluate the clinical impact of DL-based myelitis detection systems by assessing their effectiveness in facilitating early interventions, personalized treatment planning, and improved patient outcomes in real-world clinical settings.
- Continuous Improvement: Implement mechanisms for continuous model refinement and validation using real-world data. By continuously updating the model with new insights and evidence, its predictive accuracy and clinical utility can be enhanced over time, ensuring its relevance in clinical practice.
- Investigate Longitudinal Monitoring: Explore the potential of DL models for longitudinal monitoring of myelitis progression, treatment response evaluation, and detection of subtle changes in lesion morphology over time to support informed clinical decision-making and patient management.
- Identify Future Research Directions: Identify areas for future research and development in DL-based myelitis detection, including the integration of multimodal data sources, implementation of interpretability techniques, and validation in diverse patient populations, to further enhance diagnostic capabilities and clinical utility

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey of Detection Model

In this section, a brief summary of existing techniques for myelitis detection is discussed.

The survey discusses about the various methodologies used for myelitis prediction and detection. In this section a brief discussion about the existing literature review on myelitis prediction and detection is examined. Here are some common types of algorithms and approaches used for predicting myelitis:

- Deep Learning Models: CNNs and RNNs analyze MRI scans to detect myelitis by learning patterns automatically from imaging data. They excel in capturing complex features indicative of spinal cord abnormalities
- Biomarker Analysis and biomarkers enhances the accuracy of myelitis diagnosis by integrating multiple information sources. Integrated systems offer comprehensive insights into patient health status.
- Unsupervised Learning Techniques: Clustering algorithms explore data patterns without labeled examples, aiding in identifying patient subgroups related to myelitis. Unsupervised techniques enable the discovery of hidden structures within the data.
- Ensemble Methods: AdaBoost and bagging improve prediction

accuracy by combining multiple models, enhancing robustness in myelitis detection. Ensemble methods leverage the diversity of individual models to achieve superior performance.

- Feature Selection and Reduction Techniques: Identifying relevant features from data enhances the efficiency of myelitis prediction models by reducing dimensionality. Feature selection methods focus on extracting the most informative features for accurate diagnosis.
- Convolutional Neural Networks (CNNs): CNNs are widely used in medical image analysis tasks due to their ability to automatically learn hierarchical features from image data. They have been applied to myelitis
- Detection tasks using various architectures, including both 2D and 3D CNNs. CNNs can learn to detect the presence of myelitis in medical imaging modalities such as MRI scans or ultrasound images
- U-Net: U-Net is a popular architecture for semantic segmentation tasks, where the goal is to assign a class label to each pixel in an image. U-Net and its variations have been applied to myelitis segmentation tasks, where the model learns to delineate the boundaries of myelitis in medical images, such as MRI

- Cytokine levels are examined as potential indicators of myelitis, offering insights into underlying inflammatory mechanisms. Biomarker analysis provides valuable information for understanding disease progression and treatment response.
- Pattern Recognition Approaches: Texture analysis identifies spinal cord inflammation by capturing specific features of myelitis lesions on medical images. These approaches enhance the characterization of abnormalities in imaging data.
- Integrated Diagnostic Systems: Combining clinical data, imaging,

- DenseNet: DenseNet is an architecture that emphasizes feature reuse and connectivity between layers. It has been used in various medical image analysis tasks, including myelitis detection. DenseNet's dense connectivity pattern allows for efficient feature propagation through the network, which can be beneficial for tasks with limited training data or complex image structures.
- ResNet: ResNet (Residual Neural Network) is another widely used architecture, known for its deep network design with skip connections. ResNet and its variants have been applied to myelitis detection tasks, leveraging their ability to train very deep networks effectively. The skip connections help mitigate the vanishing gradient problem, enabling training of deeper architectures.

2.1.1 Existing Deep Learning Models for Myelitis Detection

E Bulut and T Shoemaker et al. [3] proposed an approach to detection of recurrence after an initial event of transverse myelitis helps to guide preventive treatment and optimize outcomes. Our aim was to identify MR imaging findings predictive of relapse and poor outcome in patients with acute transverse myelitis of unidentified etiology.
Cord expansion, contrast enhancement, and the presence of bright spotty lesions could be used as early MR imaging predictors of relapse in patients with acute transverse myelitis of unidentified etiology. Collaborative studies with a larger number of patients are required to validate these findings.
In the initial evaluation of acute transverse myelitis, a specific underlying cause of the immunologic attack cannot be determined in many circumstances. The early prediction of recurrence in such cases helps to guide preventive treatment, which, in turn, may improve long-term prognosis. Alternatively, features that favor monophasic disease may save the patients from unnecessary long-term immunosuppression. Accordingly, several independent demographic and laboratory risk factors for recurrent disease after acute transverse myelitis have been determined, including female sex, African American race, vitamin D insufficiency, and serum antibodies such as anti-aquaporin 4 antibodies, anti-Ro/SS-A antibodies, and a high (≥1:160) antinuclear antibody titer.1 Previous studies also suggested clinical factors associated with poor functional recovery, such as symmetric motor dysfunction at onset, sphincter dysfunction, and spinal cord shock-like symptoms.
Sinan Tatli and Gulay Macin et al. [1] proposed an approach to offer the best SVM parameters for data categorization. By comparing it to

the most recent approaches, researcher can see how well the suggested algorithm performs. In terms of accuracy, the suggested technique outperformed other than the current algorithms by 3.37% and 9.17%, respectively, and by a substantial margin of 34.12%.

In order to find the optimal combinations that might improve accuracy and detection rate, Linta Antony et al. [1] suggested a technique to apply unsupervised algorithms and compare their performances. The five unsupervised algorithms used in this study are Autoencoder, DB-Scan, I-Forest, and K-Means Clustering. The clinical data of CKD and non-CKD were classified with an overall accuracy of 99% by integrating them with various feature reduction and selection approaches with the K-Means Clustering algorithm. MRI scans are analyzed from 1008 people, with a mean age of 37.7 years ± 9.7 with 730 of those people being women. Out of 519 individuals, at least one had an amplifying lesion. Across all five test sets, the average sensitivity for slice-wise prediction was 78% ± 4.3 and the average specificity was 73% ± 2.7. There was a 72% ± 9.0 and a 70% ± 6.3 corresponding to the participants. The area under the curve (AUC) for slice-wise enhancement prediction was 0.82 ± 0.02 and for participant-wise enhancement prediction it was 0.75 ± 0.03. Potential textural elements that give tissue properties of the spine area in ultrasound images can be extracted from the de speckled pictures. Two features are extracted: the Grey Level Co-occurrence Matrix (GLCM) and the run length texture. The k-Nearest Neighbor classifier (k-NN) are used which divide the pictures into two categories: normal and cystic. For picture categorization into cystic and normal states, the GLCM derived characteristics are very important

Fig 2.1: Classification Accuracy for Various Classification Methods

The performance analysis of existing approaches with the proposed framework is shown in the Figure 2.1. In this plot, value in X-axis denotes different methods and a value in Y-axis denotes the recognition accuracy and determines the performance measurements for various classification methods involving metrics such as accuracy, precision, recall, and F1-score, assessing the model's ability to correctly classify instances across different classes. These evaluations help to quantify the effectiveness and reliability of classification algorithms, aiding in the selection and optimization of models for specific applications.

Table 1: Measurements of Performance of Various Classification Methods

S. no.	Model name	Accuracy	Sensitivity	Specificity	Recall	Precision	Standard deviation
1	Simple tree	90.7	99.1	0	91.4	99.1	38.20
2	Medium tree	89.0	96.5	6.9	91.8	96.5	34.73
3	Complex tree	87.2	94.3	9.3	91.8	94.3	33.14
4	Logistic regression	90.6	99.1	0.0	91.4	99.1	38.19
5	Linear SVM	91.5	1.0	0.0	91.5	1.0	44.50
6	Quadratic SVM	89.5	96.3	11.6	92.1	96.3	32.88
7	Cubic SVM	88.3	95.0	16.2	92.4	95.0	30.68
8	Fine Gaussian SVM	91.1	99.1	4.6	91.8	99.1	36.43
9	Medium Gaussian SVM	91.5	1.0	0.0	91.5	1.0	44.50
10	Coarse Gaussian SVM	91.5	1.0	0.0	91.5	1.0	44.50
11	Ensemble boosted tree	90.9	98.2	11.6	92.3	98.2	33.45
12	Ensemble bagged tree	91.5	99.5	4.6	91.8	99.5	36.55
13	Ensemble RUS boosted tree	63.1	62.9	65.1	95.1	62.9	12.66
14	Artificial neural network	95.3	95.9	60	99.2	95.9	14.69

Table 1 determines the measurements of performance for different classification methods involving metrics like accuracy, precision, recall, and F1-score by evaluating the model's ability to classify instances across different classes. These evaluations help quantify the effectiveness and reliability of classification algorithms, aiding in the selection and optimization of models for specific applications

.

Table 2: Measurements of Performance of Various Classification Methods

SL No.	Authors	Methodology Used	Accuracy
1.	J.E. Small (2021)	Deep Learning	77.36%
2.	Elsevier Inc (2023)	CNN	92.75%
3.	Raghavendra et al. (2021)	Crow Search-Rider optimization	88.82%
4.	Naofumi Tomita (2020)	Deep Learning (CNN)	89.28%
5.	A.B. Paul and M. Kunst (2021)	Convolutional Neural Network (CNN	92%
6.	S. Benetos and John Vlamis (2020)	ANN (Artificial neural networks)	85.1%
7.	Showmick Guha Paul (2021)	Deep learning, Computed tomography	90.75%
9.	Saman Ebrahimi	CNN	84%
10.	Sunanda Biradar (2020)	Support Vector Machines (SVM), Random Forests, CNNs	98.5%

It is observed from the Table 3 that CNN provides a recognition accuracy of 92 percent and outperforms better than existing algorithms for myelitis detection.

Classification accuracy measures the effectiveness of different classification methods in correctly assigning labels to instances, with higher accuracy indicating better performance. The accuracy will change accordingly based on the dataset used, the nature of problem, and the quality of the features used as shown in Figure 2.1. Different algorithms may perform better or worse in different scenarios.

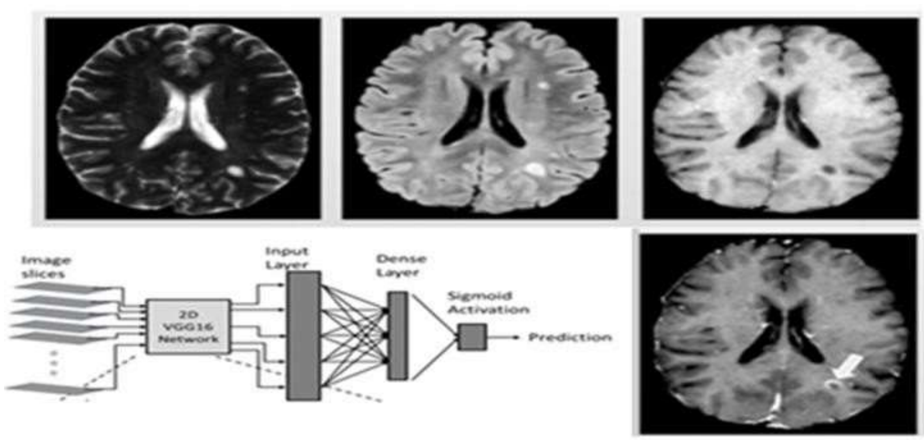

Fig 2.2: Illustration of the Training of the Six Shallow Machine Learning Model

Figure 2.2 shows the illustration of training six shallow machine learning models involves initial steps of feature extraction and selection, followed by hyperparameter tuning to optimize model performance. The process concludes with model validation, ensuring robustness and effectiveness in handling diverse datasets and tasks.

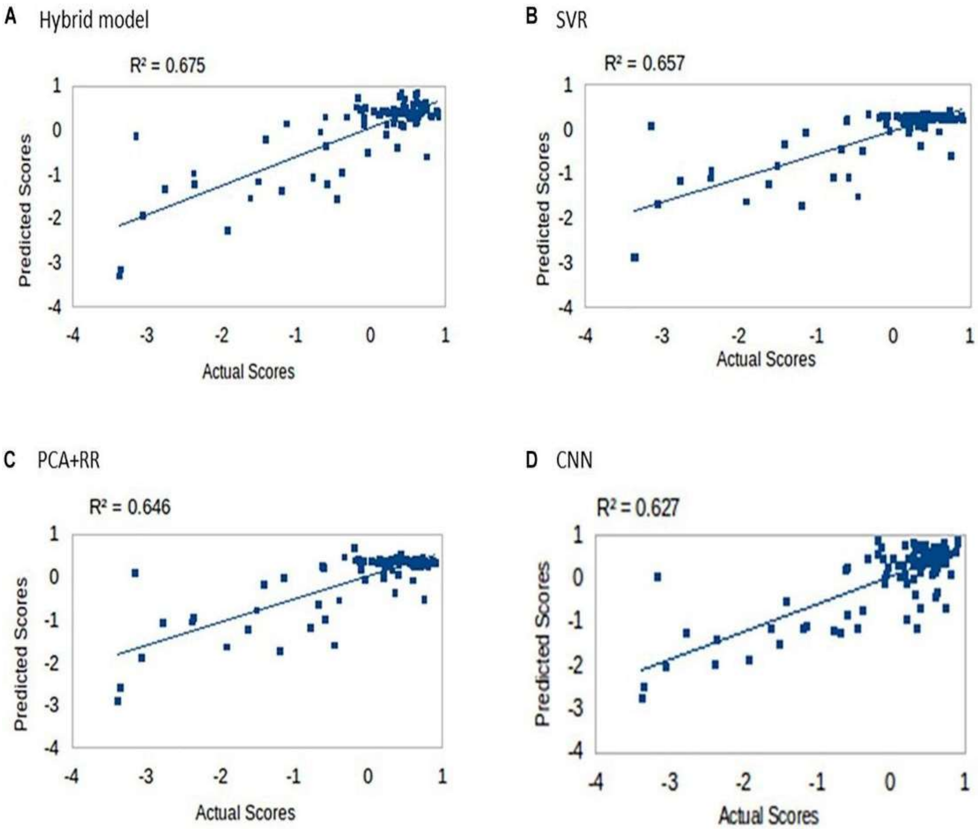

Fig 2.3 Language Cores Detected by the Four Competing Models: (A)Hybrid model, (B) SVR, (C) PCA + RR, and (D) CNN.[4]

Classification accuracy measures the effectiveness of different classification methods in correctly assigning labels to instances, with higher accuracy indicating better performance is depicted in the figure 2.3. The metrics are essential tools for evaluating the performance of classification models and are often used in conjunction to gain a comprehensive understanding of a model's effectiveness.

In this section existing datasets for myelitis Detection is discussed and same is tabulated in Table 3

2.1.2 Existing Datasets for Myelitis Detection

In this section existing datasets for Kidney-stone Prediction and Detection is discussed and same is tabulated in Table 3.

Table 3: Existing Datasets

SL No.	Name	Source	No. of Samples
1	Myelitis Images	Kaggle	12388
3	CKD Dataset	Kaggle	400

The above table interprets the datasets containing MRI images that are used in training the model. Around 30% of the datasets are used for testing purposes. The datasets are sourced from GitHub and Kaggle.

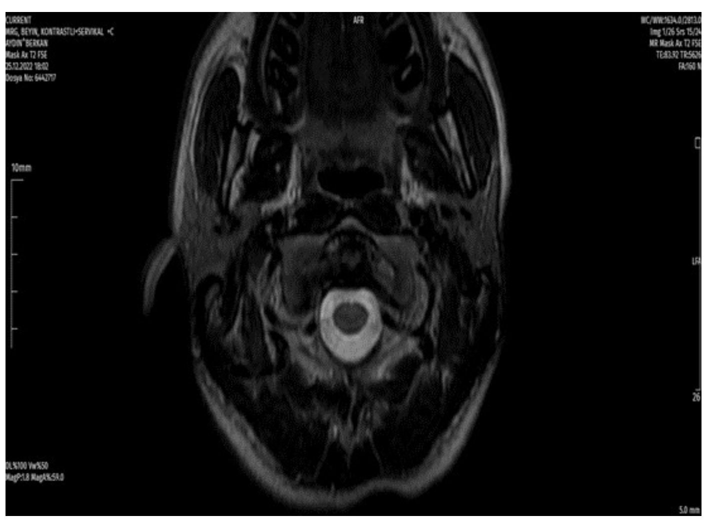

Fig 2.4: Sample MRI Scans from the Dataset

Sample MRI scans from a dataset comprises cross-sectional images generated by computed tomography method as shown in Figure 2.4. These scans serve as crucial input data for training and analyze the evaluation of the DL model.

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 System Requirements

- **Operating System:** The system must support Windows (7 or higher), macOS (10.12 or higher), and popular Linux distributions like Ubuntu (16.04 LTS or higher), CentOS (7 or higher), etc.
- **Memory:** Ensure a minimum of 8GB RAM for smooth execution of DL models and the web application. Optimal performance requires 16GB or higher RAM.
- **Processor:** A multi-core processor (Intel Core i5 or higher, AMD Ryzen 5 or higher) is essential for efficient computational tasks, especially during model training.
- **Storage:** Allocate at least 5GB of free disk space for datasets, trained models, and project files. Utilize Solid State Drives (SSDs) for faster read/write speeds.
- **Web Browsers:** Conduct compatibility testing with modern web browsers (Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge), covering both desktop and mobile versions. Support the latest stable versions, along with one or two previous versions for broader compatibility.

3.2 Hardware Requirements:

- **Devices Capable of Accessing a Web Browser (e.g., Laptop, Phone):** Users will require devices with sufficient processing power and display capabilities to effectively interact with the system's web-based interface. This includes laptops, desktop computers, tablets, or smartphones with modern web browsers. The devices should support HTML5, CSS3, and JavaScript for seamless rendering of the user interface elements and responsive interaction.
- **CPU:** A modern processor with multiple cores (quad-core or higher) is recommended to accelerate computation tasks, particularly for training deep learning models.
- **GPU (optional):** High-end graphics cards (GPUs) are specialized hardware components designed for accelerating graphics rendering and parallel processing tasks. In the context of the real-time person tracing and tracking system, GPUs can be utilized to offload computationally intensive tasks from the CPU, such as deep learning inference, feature extraction, and object detection. GPUs with CUDA (Compute Unified Device Architecture) support from NVIDIA or similar technologies from other manufacturers are preferred for maximizing performance gains.
- **Memory:** It is recommended to have 8GB RAM or higher for loading datasets into memory and running DL models efficiently. Higher memory capacity may be required for handling larger datasets.
- **Storage:** Ensure adequate storage space to store large datasets, trained models, and other project-related files. Consider using External Hard Drives for additional storage capacity.

3.3 Software Requirements

- **Python 3.x:** The project relies on Python for implementing the DL models and backend server. Ensure Python 3.x is installed on the system, along with package management tools such as pip or conda. Python is a high-level, interpreted programming language known for its simplicity and readability. Version 3.11 introduces new features and improvements over previous versions. It will serve as the primary language for developing the myelitis detection and prediction system, providing access to libraries, tools, and frameworks necessary for implementing algorithms, managing backend logic, and integrating various components of the system.
- **Flask Web Framework:** Flask is a lightweight and flexible web framework for Python. It provides tools and libraries for building web applications and APIs. Flask is chosen for its simplicity, modularity, and ease of use. It will be utilized to create the user interface of the real-time person tracing system, handling HTTP requests, rendering HTML templates, and managing session data. Flask's extensibility allows for easy integration with other Python libraries and frameworks.
- **HTML, CSS, JavaScript:** Frontend technologies for designing and developing the user interface. No specific software installation is required, but familiarity with frontend development tools (e.g., Visual Studio Code, Sublime Text) is recommended.
- **TensorFlow, scikit-learn:** Python libraries for implementing the CNN model (TensorFlow) and SVM classifier (scikit-learn). Install these libraries using pip or conda, along with any required dependencies.

- **Other dependencies**: NumPy, Pandas, Matplotlib, etc., are required for data manipulation, visualization, and analysis. Install these dependencies using pip or conda.
- **Virtual Environment:** It is recommended to use a virtual environment to manage project dependencies. Virtual environments isolate project dependencies from system- wide packages, preventing version conflicts and ensuring space efficiency. Tools like virtualenv or conda environments can be utilized for creating and managing virtual environments.

CHAPTER 4

SYSTEM DESIGN AND METHODOLOGY

4.1 System Design

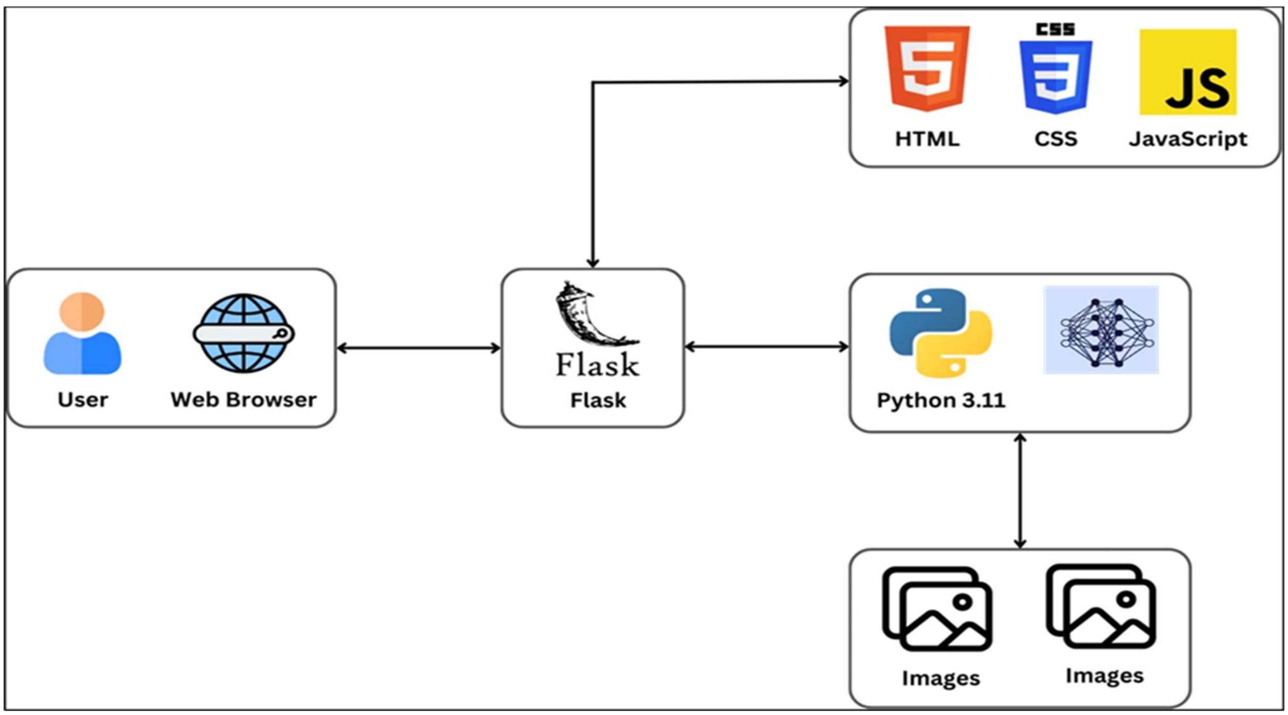

Fig 4.1: System Design Architecture

Flask Server:

- The Flask server serves as the backend infrastructure for the web-based DL application.
- It receives incoming HTTP requests from the client interface and routes them to the appropriate endpoints.
- Flask handles request processing, model inference, and response generation.

DL Model(s):

- DL models are trained offline using historical data and validated to ensure accuracy and reliability.
- Trained models are serialized and stored in a format compatible with Flask's deployment requirements (e.g.,.h5).
- The Flask server loads the pre-trained DL model(s) into memory upon startup, ready for making detections in real-time.

Request Handling and Detection:

- When the Flask server receives a detection request from the client interface, it extracts the input data from the request payload.
- The input data is pre-processed as per the requirements of the DL model(s) (e.g., feature scaling, encoding).
- The pre-processed data is passed to the loaded DL model(s) for inference, which generates predictions based on the learned patterns and relationships in the data.
- Predictions are returned as JSON responses to the client interface, containing relevant information such as detected outcomes, probabilities, or confidence scores.

4.2 Methodology

Detection Model

- Data Collection and Preparation:
    - A dataset of MRI images that depict the presence or absence of myelitis is gathered.
    - Annotate the images to label regions of interest (ROI) containing myelitis and background regions.

  - Images are pre-processed by resizing them to a uniform resolution, normalizing pixel values, and augmenting the dataset to increase diversity and robustness.

- Dataset Splitting:
  - Divide the dataset into training, validation, and testing sets. The split ratio is 70:30.
  - The distribution of images with and without myelitis is balanced across the datasets to prevent bias.
- Model Architecture Design:
  - CNN model is used for 3D medical imaging data such as MRI scans, design architectures that capture spatial and temporal features across multiple slices.
- Model Training:
  - Initialize the CNN model with random weights or pre-trained weights.
  - Train the model using the training dataset and optimize its parameters to minimize a loss function (e.g., binary cross-entropy loss).
  - Utilize techniques like mini-batch gradient descent and backpropagation to update the model weights iteratively.
  - Monitor the model's performance on the validation set to prevent overfitting and adjust the parameters accordingly.
- Model Evaluation:
  - Evaluate the trained CNN model on the held-out testing dataset to assess its performance.
  - Calculate evaluation metrics such as accuracy, precision, recall, F1- score, and area under the receiver operating characteristic curve (ROC-

AUC).

- Visualize model predictions and compare them with ground truth annotations to identify any discrepancies or errors.

- Post-processing:
  - Apply post-processing techniques to refine the model predictions and improve their interpretability.
  - Thresholding: Apply a threshold to the model's output probabilities to classify pixels or regions as either myelitis or background.
  - Morphological Operations: Perform operations like erosion, dilation, and filtering to remove noise and artifacts from the predicted regions.

- Deployment and Integration:
  - Deploy the trained CNN model into a production environment using frameworks like Flask.

4.2.2 General Architecture of Myelitis Detection:

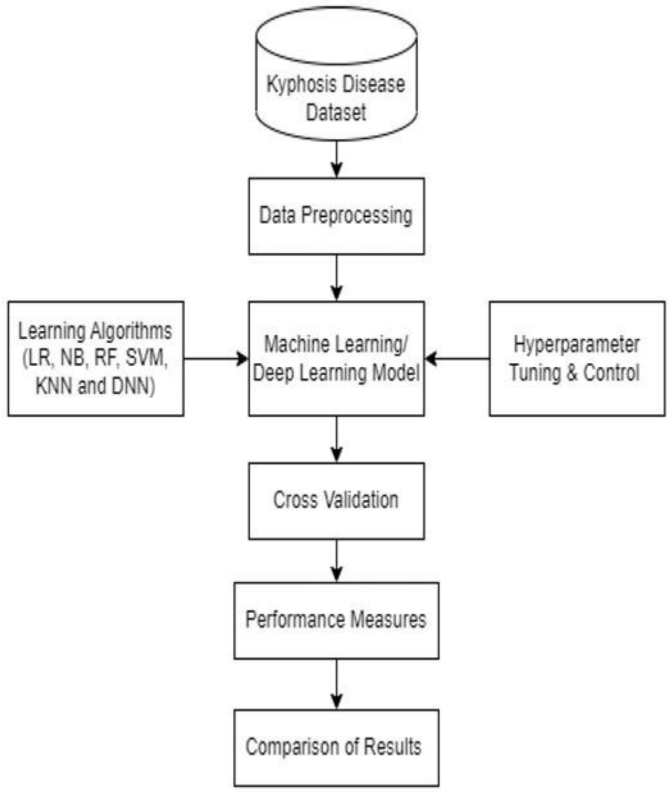

Fig 4.2: Flowchart of General Architecture of Myelitis Detection

The comprehensive design of the detection of the myelitis is presented in Figure 4.3. The steps are as follows:

- Data Collection: Acquire diverse medical imaging datasets comprising MRI scans and related medical data concerning spinal cord injuries and myelitis
- Data Preprocessing: Clean, organize, and standardize the collected data to ensure consistency and prepare it for input into the Deep Learning model, addressing issues such as noise reduction and normalization.
- Feature Extraction: Extract relevant features from the pre-processed data, such as specific patterns or characteristics indicative of spinal cord injuries or myelitis, to enhance the model's ability to make accurate diagnoses.
- Deep Learning Model Development: Design and train the Deep Learning model, incorporating advanced techniques and architectures suited for analyzing medical imaging data and detecting spinal cord issues with precision.
- Model Validation and Optimization: Validate the model's performance using rigorous testing procedures and optimize its parameters to ensure high accuracy and reliability in diagnosing spinal cord problems.
- Real-time Integration and Deployment: Integrate the trained model into a real-time decision support system, ensuring seamless deployment in clinical settings for prompt interventions and aiding healthcare professionals in diagnosing spinal cord issues effectively.

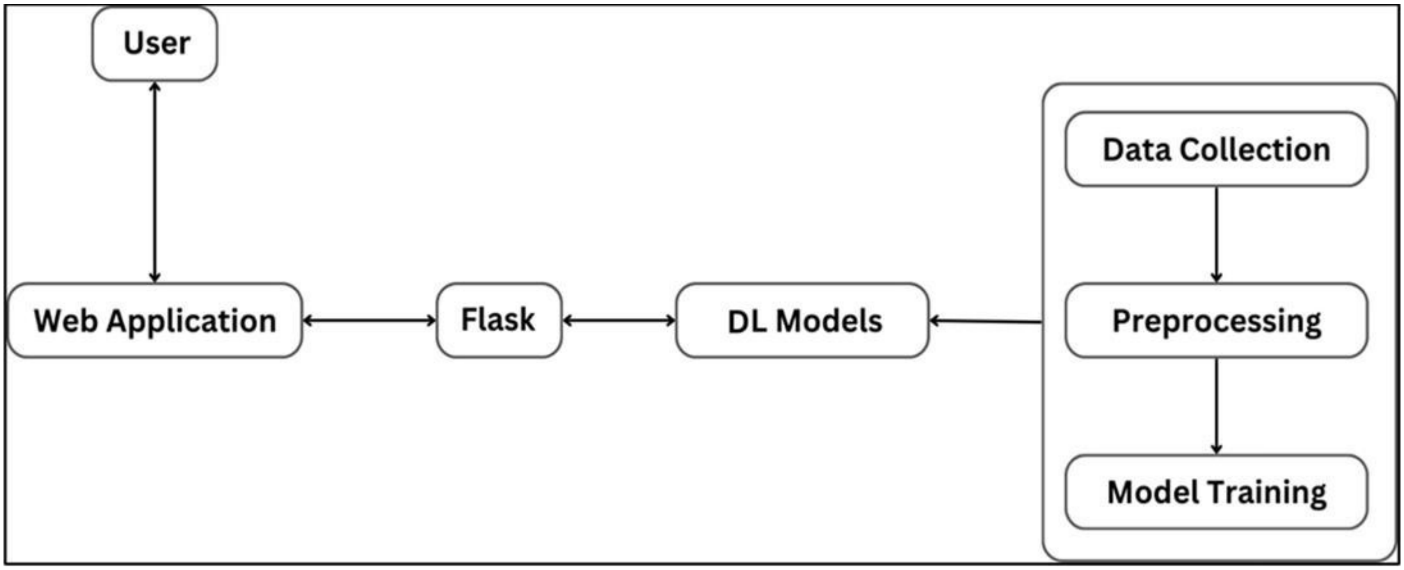

Fig 4.3: Workflow of the Project

CHAPTER 5

IMPLEMENTATION

5.1 Overview

The project aims to create essential tools for healthcare professionals: a myelitis image classification Convolutional Neural Network (CNN) model. These tools are designed to aid in diagnosis and treatment planning. A key feature is a user-friendly web application enabling users to either upload MRI images for detection.

5.2 Architecture

The high-level architecture of the project consists of frontend, backend, and DL model components. User interactions are facilitated through a web interface, with data processed and detects generated by the backend server. DL models are integrated into the backend server, allowing seamless communication between frontend and backend components.

5.3 Implementation Details

- **CNN Model:** The myelitis image classification model is implemented using Convolutional Neural Networks (CNNs) in TensorFlow. The model architecture consists of multiple convolutional layers followed by fully connected layers for classification.
- **Frontend Development:** Frontend development involves he

creation of the user interface (UI) of a web application, focusing on how users interact with and experience the application. Here's an elaboration on each component:

HTML (Hypertext Markup Language): HTML is the standard markup language used to structure the content of web pages. It provides the basic building blocks for creating web pages by defining the structure and layout of elements such as text, images, forms, and links.

CSS (Cascading Style Sheets): CSS is used to style and format the HTML elements defined in the web page. It allows developers to control the appearance of elements, including layout, colors, fonts, spacing and animations. CSS ensures consistency and enhances the visual appeal of the UI.

JavaScript: JavaScript is a powerful programming language that adds interactivity and dynamic behavior to web pages. It enables developers to create responsive and interactive features such as form validation, user input handling, DOM manipulation, and asynchronous data fetching. JavaScript enhances the user experience by making web applications more engaging and interactive. In frontend development, HTML provides the structure of the UI elements, CSS styles the appearance of these elements, and JavaScript adds functionality and interactivity to the UI. Together, these technologies work seamlessly to create a user-friendly and visually appealing web interface that enhances the overall user experience.

- **Backend Development:** Flask, a lightweight Python web framework, is used for developing the backend server. Flask provides routing mechanisms for handling HTTP requests, along with support for integrating DL models and database operations.
- **Data Management:** Datasets are collected from reliable sources and pre-processed to ensure data quality and consistency. Data augmentation techniques such as rotation, scaling, and flipping may be applied to increase the diversity of training data for DL models.
- **Integration:** DL models were integrated into the backend server using Flask's application structure. Model inference was performed based on user input received through API endpoints. Frontend components interacted with backend APIs using asynchronous JavaScript (AJAX) calls to send requests and receive predictions. JSON (JavaScript Object Notation) format was used for data exchange between frontend and backend components.

  Additionally, Python libraries such as pickle and Pillow were employed in saving and loading the CNN and SVM classifier models. Pickle facilitated the serialization and deserialization of the models, allowing them to be stored persistently and loaded into memory when required. Pillow was utilized for image processing tasks, enabling the manipulation and preprocessing of myelitis images before feeding them into the CNN model for classification.

CHAPTER 6
TESTING

6.1 Unit Testing

Comprehensive unit testing is conducted for individual components, including DL models, frontend UI elements and backend APIs.
Test cases are designed to cover various scenarios and edge cases to ensure robustness and reliability of the system.
Tools such as Postman were utilized for testing Flask API endpoints. Postman offers a user- friendly interface for sending HTTP requests to the backend server and inspecting the responses. This approach allows for thorough testing of API functionality and ensures that the endpoints behave as expected, facilitating the validation of data transmission, error handling, and overall, API performance.

6.2 Integration Testing

End-to-end integration testing is performed to validate system functionality and ensure seamless interaction between different components.
Test scenarios are designed to cover user interactions, data processing, and model predictions, simulating real-world usage scenarios.
Manual Testing for frontend and tools such as Postman for API testing is used to automate test execution and validate system behavior.

6.3 User Acceptance Testing (UAT)

UAT is conducted with end-users to gather feedback on system usability, functionality, and performance. Test cases are executed by end-users in a controlled environment, with feedback collected through surveys interviews, or feedback forms.
User feedback is analyzed and incorporated into the final version of the application to address usability issues and improve overall user satisfaction.

6.4 Performance Testing

Performance testing was conducted to evaluate system responsiveness and scalability under various load conditions. Metrics such as response time, throughput, and resource utilization were measured to identify performance bottlenecks and optimize system performance. In addition to assessing system performance, DL model accuracy testing was crucial for evaluating the effectiveness of the myelitis image classification CNN model. Testing involved comparing the performance of the models on test data against training data to ensure they generalized well to unseen data. Additionally, the accuracy of the models on data found in open sources, such as images sourced from platforms like Google, was evaluated to assess their robustness and real-world applicability.

6.5 Security Testing

Security testing measures are implemented to identify and mitigate potential vulnerabilities in the system. Penetration testing, vulnerability scanning, and code review are conducted to identify security weaknesses and enforce best practice.

CHAPTER 7

RESULT AND ANALYSIS

7.1 Detection of Myelitis

Detecting myelitis in MRI scans using a Convolutional Neural Network (CNN) involves preprocessing the images to extract relevant features and training a CNN model to classify the presence of myelitis. The CNN learns to identify patterns indicative of myelitis, such as their shape, size, and density, enabling detection in MRI scans.

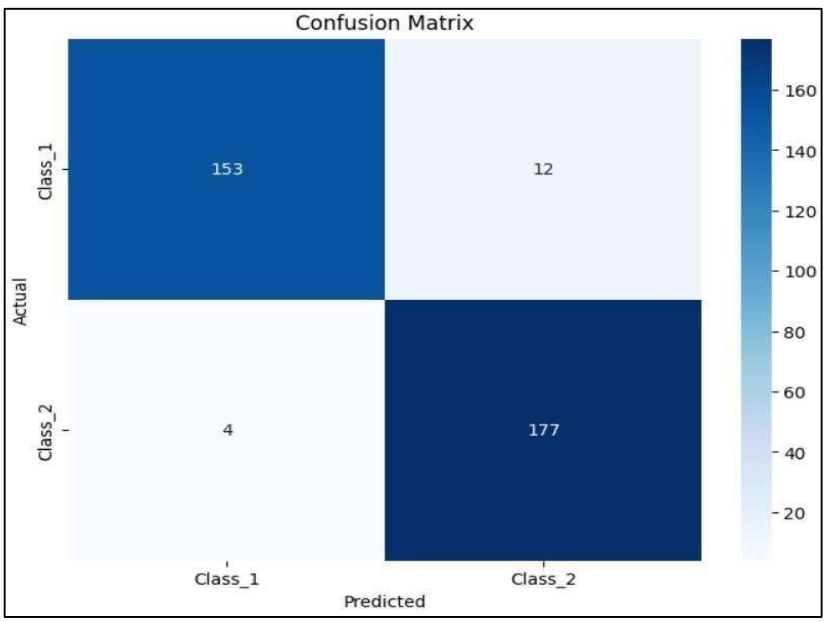

Fig 7.1: Confusion Matrix of the Detection Model

7.1.1 Performance Evaluation of Detection of Myelitis

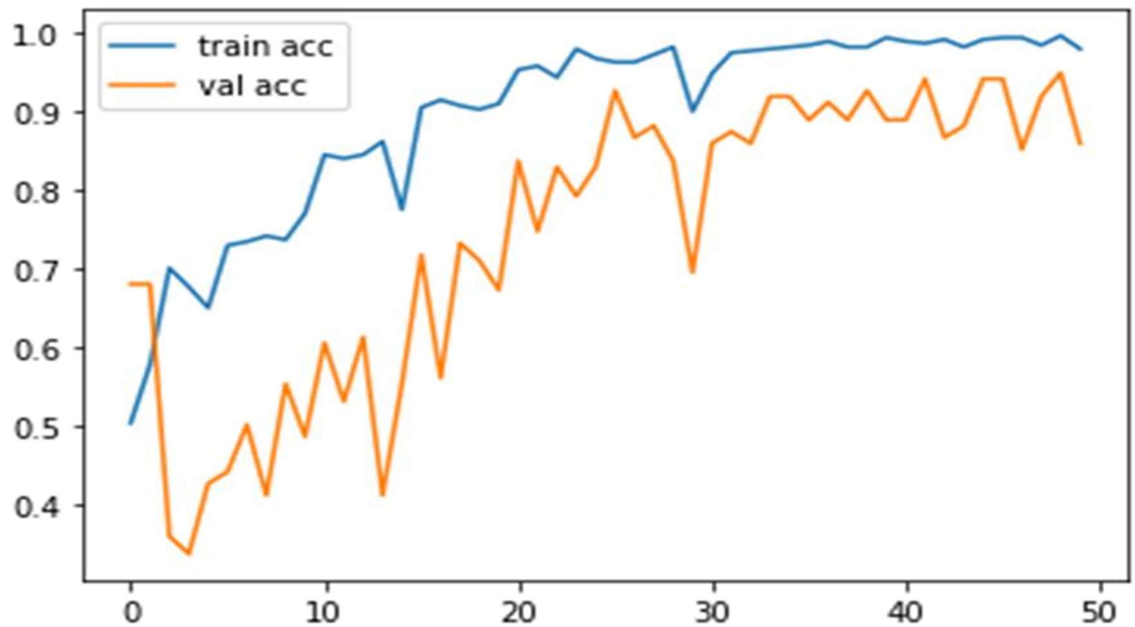

Fig 7.2: The Trained Model's Accuracy

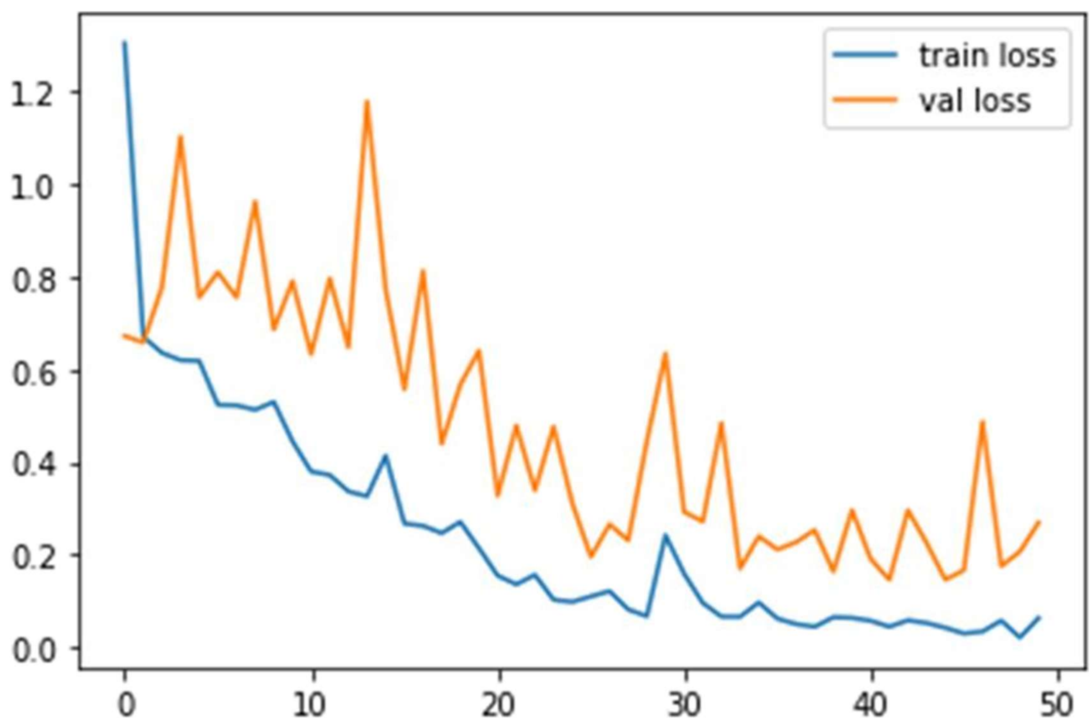

Fig 7.3: The Model's Loss at each Epoch

Input: MRI

Output

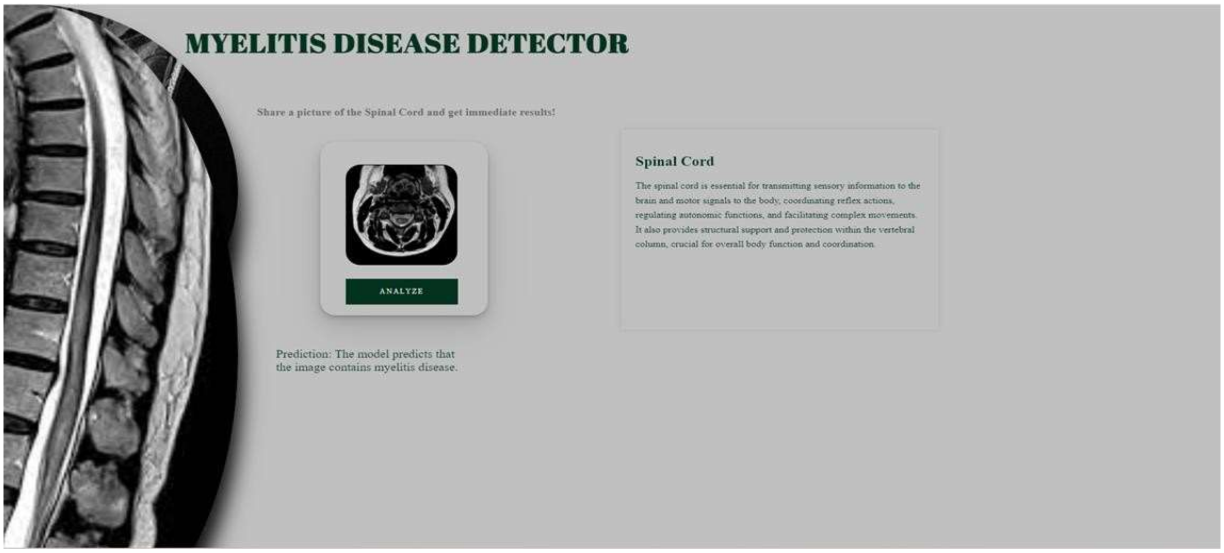

Fig 7.4: Detection of Myelitis Based on MRI Images

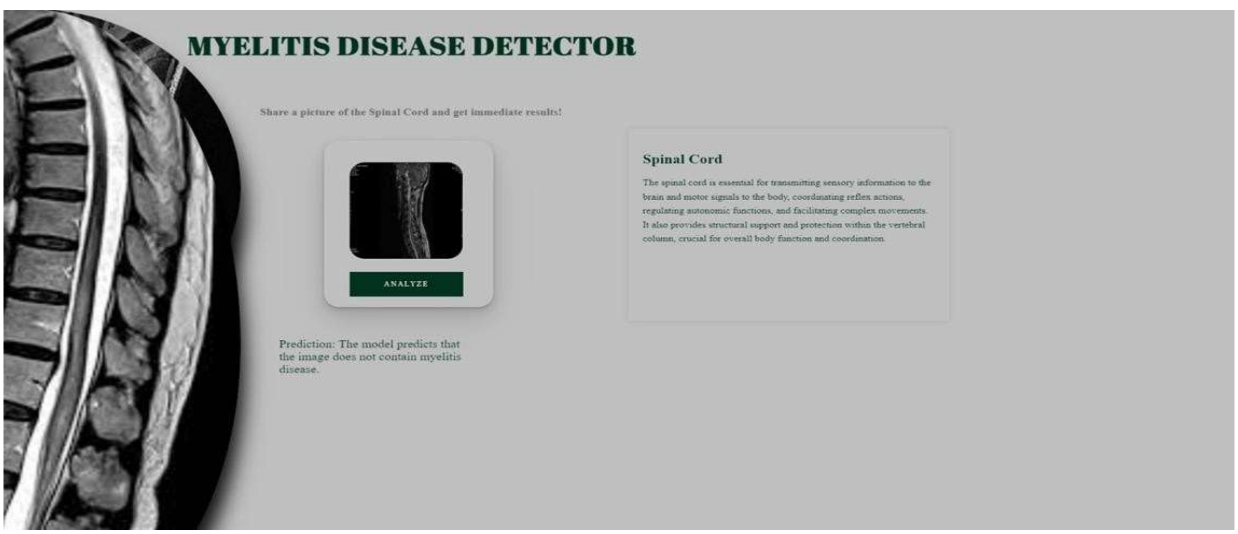

Fig 7.4: Detection of Not Myelitis Based on MRI Images

CHAPTER 8

CONCLUSION

The incorporation of deep learning models in myelitis detection marks a significant progression in medical imaging and diagnostic methodologies. These models present automated, highly sensitive, and specific approaches to identify inflammatory lesions within the spinal cord, thereby simplifying diagnostic processes and facilitating prompt interventions. Deep learning's ability to offer quantitative analysis and personalized insights enables early detection, individualized treatment planning, and continual monitoring of myelitis patients. Additionally, insights derived from deep learning contribute to our comprehension of myelitis pathophysiology and guide research endeavors aimed at refining diagnostic and therapeutic approaches. As deep learning algorithms continue to advance, their role in myelitis detection is expected to broaden, leading to improved clinical decision-making, optimized patient care, and ultimately better outcomes for those affected by spinal cord inflammation. Sustained research efforts and collaborative initiatives are vital to further develop and validate these models for widespread clinical. This research paper showcases the transformative potential of AI and DL in revolutionizing medical diagnostics, improving patient outcomes, and enhancing efficiency of the healthcare.

FUTURE ENHANCEMENTS

- Validation and Generalizability: Further research is necessary to validate the performance and applicability of machine learning models, such as SVM-CNN, in detecting myelitis across diverse patient populations and healthcare settings. This includes assessing model performance on different demographics, disease subtypes, and imaging modalities to ensure robustness and generalizability.
- Integration of Additional Data Sources: Incorporating additional data sources, such as genetic information, cerebrospinal fluid biomarkers, or clinical history features, may improve the predictive accuracy and specificity of machine learning algorithms for myelitis detection. These complementary data sources could provide valuable insights into disease mechanisms and enhance diagnostic capabilities.
- Advancements in Deep Learning and Imaging Technology: Continued advancements in deep learning techniques and medical imaging technology hold promise for further refinement and optimization of CNN-based detection models for myelitis. Innovations in image acquisition, preprocessing, and feature extraction methods could enhance the sensitivity and specificity of CNNs in identifying characteristic patterns of spinal cord inflammation.
- Integration into Clinical Decision Support Systems: Integrating machine learning models for myelitis detection into clinical decision support systems (CDSS) could streamline diagnostic workflow

REFERENCES

[1]. J E Small, P Osler, A B Paul, M Kunst,” CT Cervical Spine Fracture Detection Using a Convolutional Neural Network”, Apr,2021.

[2]. Showmick Guha Paul, Arpa Saha, Md Assaduzzaman, “A real-time deep learning approach for classifying cervical spine fractures”, Dec,2023.

[3]. Munavar Jasim, Thomas Brindha, “Spinal cord segmentation and injury detection using a Crow Search- Rider optimization algorithm”, Dec,2020.

[4]. Naofumi Tomita, Yvonne Y Cheung, Saeed Hassanpour, “Deep neural networks for automatic detection of osteoporotic vertebral fractures on CT scans”, May,2018.

[5]. Themistoklis Tragaris, Ioannis S Benetos, John Vlamis, and Spyridon Pneumaticos, “Deep Learning Applications in Spine Surgery”, Oct,2023.

[6]. Anu Jacob, Brian G. Weinshenker, “An Approach to the Diagnosis of Acute Transverse Myelitis”, Dec,2018.

[7]. Lee YJ, Terbrugge KG, Saliou G, Krings T. “Clinical features and outcomes of spinal cord arteriovenous malformations”, Oct,2019.

[8]. Samir Alkabie, Courtney S. Casserly, Sarah A. Morrow, Juan M. Racosta, “Identifying specific myelopathy etiologies in the evaluation of suspected myelitis”, Jul,2023.

[9]. Yo Huh, Eun-Jung Park, Ju-Won Jung, Sungho Oh, Sang-Cheon Choi, “Clinical insights for early detection of acute transverse myelitis in the emergency department”, Mar,2018.

[10]. V P Chandnani, J Beltran, C S Morris, S N Khalil, C F Mueller, J M Burk, W F Bennett, P B Shaffer, “Acute experimental osteomyelitis and abscesses: detection with MR imaging versus CT”, Mar,2019.

[11]. Divyanshu Dubey, MD1,2; Sean J. Pittock, MD1,2; Karl N. Krecke, “Radiologic, Features of Myelitis Associated with Myelin Oligodendrocyte Glycoprotein Autoantibody”, Dec,2018.

[12]. Zajicek JP, Scolding NJ, Foster O, “Central nervous system sarcoidosis: diagnosis and management”. Mar,2020.

[13]. Birnbaum J, Petri M, Thompson R, Izbudak I, Kerr D, “Distinct subtypes of myelitis in systemic lupus erythematosus”, May,2018

[14]. Gailloud P, Gregg L, Galan P, Becker D, Pardo C. Periconal, “Arterial anastomotic circle and posterior lumbosacral watershed zone of the spinal cord”, Jun,2015.