

A Web-Integrated Convolutional Neural Network Framework for MRI-Based Multiple Sclerosis Screening: Architecture and Evaluation Protocol

Manish Kashyap¹, Jyaneswar Rao A^{2,*}, Bodabanda Srinath³, Lingutla Bhargav⁴, and K N Surya Prakash Y⁵

^{1,2,3,4,5}Department of Computer Science & Engineering (AIML), Parul University, Vadodara, Gujarat, India

¹2403031469002@paruluniversity.ac.in, ²2303031460006@paruluniversity.ac.in,

³2303031460026@paruluniversity.ac.in, ⁴2303031460104@paruluniversity.ac.in,

⁵2303031460225@paruluniversity.ac.in

Under the Guidance of: **Prof. Sheeka Raina**, Department of Computer Science & Engineering (AIML), Parul University

* Corresponding Author: Jyaneswar Rao A

Abstract—Multiple sclerosis (MS) is a chronic demyelinating disease of the central nervous system whose diagnosis depends heavily on the reading of brain magnetic resonance imaging (MRI), a task that is time-consuming and subject to inter-reader variability. Published deep learning work on MS concentrates on lesion segmentation and classification algorithms, whereas the packaging of such a model into an auditable, deployable decision-support workflow with explicit evaluation safeguards receives less attention. This paper presents the design of Turing Body, a web-integrated framework that classifies brain MRI images as MS or normal using a convolutional neural network (CNN) and returns a report containing the predicted class, a confidence score and a lesion visualisation. The pipeline comprises slice extraction, resizing, intensity normalisation, noise reduction and data augmentation, followed by CNN classification implemented in Python with TensorFlow/Keras. The model is exposed through a Flask inference service, coordinated by a Spring Boot backend, presented through a React interface and persisted in MongoDB. We document the layered architecture, the functional and non-functional requirements, and the current implementation status, in which the backend and model modules are functional while frontend integration remains in progress. Because the project has not yet produced quantitative performance results, no accuracy figures are claimed. Instead, we define a patient-wise evaluation protocol, a metric set and a reporting standard intended to prevent slice-level data leakage and control-cohort confounding. The contribution is a documented reference architecture and a pre-specified, reproducible evaluation plan that subsequent experiments on public MS MRI benchmarks can follow.

Keywords—deep learning, medical image classification, clinical decision support, microservices, data leakage, reproducibility.

I. INTRODUCTION

Multiple sclerosis is a chronic inflammatory disease of the central nervous system in which demyelinating lesions accumulate in the brain and spinal cord [1]. Magnetic resonance imaging is central to its diagnosis: the 2017 revisions of the McDonald criteria use MRI evidence of lesion dissemination in space and time, alongside clinical findings, to establish the diagnosis [2]. Lesions can be small and variable in appearance, and examining every slice of a study is laborious, which motivates computational assistance for lesion identification [3].

Convolutional neural networks (CNNs) have become the dominant approach to image recognition [4], [5] and are

widely applied in medical image analysis [6]. A dedicated review of deep learning for automated MS detection from MRI summarises the methods in use and the open challenges [7]. Community benchmarks show how far automation has progressed. In the MICCAI 2016 challenge, thirteen lesion segmentation methods were evaluated on 53 MS cases from four centres, each annotated by seven experts, and the organisers concluded that automatic algorithms still trailed human expertise in both detection and delineation [8]. Automated tools for MS therefore remain best positioned as support for, rather than replacement of, clinical judgement.

The benchmark literature reviewed here is organised around algorithmic performance on curated data. System-level questions receive less space in these sources: how a trained model is wrapped in a secure, role-based workflow; what a clinician-facing report should contain; and, for classifiers that operate on two-dimensional slices, how an evaluation must be designed so that reported performance is not inflated by leakage between training and test data. This statement refers to the sources reviewed and is not a claim about the whole literature.

This paper describes Turing Body, a student-developed prototype that classifies brain MRI images as MS or normal and presents the result through a web application. The paper is deliberately limited to what the project documentation supports. The project has documented the architecture, requirements, preprocessing steps and the model-development workflow, but it has not yet produced quantitative performance results. We therefore report no accuracy, sensitivity or specificity values. We present the system as a design and specification study, give an explicit evidence-status audit of each capability, and define the evaluation protocol that any future experiment should follow. Section II reviews related work, Section III states the contributions, Sections IV to VI describe the architecture, methodology and implementation status, Section VII specifies the evaluation protocol, and Sections VIII to XII discuss limitations, ethics, future work and conclusions.

II. RELATED WORK

A. Clinical Background and the Role of MRI

The McDonald criteria formalise how MRI lesions contribute to diagnosis, and the 2017 revision extends the use of symptomatic and cortical lesions for demonstrating dissemination [2]. The panel also identified validation in diverse populations and the incorporation of advanced imaging as priorities for further research. Two consequences follow for automated tools. A single-image classifier cannot reproduce a criteria-based diagnosis that depends on clinical context and temporal evolution, so its output is best framed as screening support. Evidence of performance in diverse populations is a requirement for any clinical claim, not an optional extra.

B. Deep Learning for MS Lesion Analysis

Much of the MS deep learning literature addresses lesion segmentation. Valverde *et al.* [9] proposed a cascaded three-dimensional CNN for MS lesion segmentation, and Kamnitsas *et al.* [10] described a multi-scale three-dimensional CNN with a fully connected conditional random field for brain lesion segmentation more generally. The U-Net architecture [11] and residual connections [12] are common building blocks in this area and in medical image classification. Shoeibi *et al.* [7] review the use of deep learning for automated MS diagnosis from MRI and discuss challenges and future directions. These works study model design and accuracy; they do not prescribe a deployment architecture.

C. Public Benchmarks

Two open resources are most relevant for evaluation. The ISBI 2015 longitudinal challenge released data in which white-matter lesions were delineated by two raters across 82 data sets, with training data from five subjects and test data from fourteen subjects [3]. The MICCAI 2016 challenge provided 53 cases from four centres with seven expert annotations per case [8]. Both are segmentation benchmarks in which the subjects have MS. An MS-versus-normal classifier additionally needs healthy-control scans, which these resources do not supply; this affects the design of the evaluation (Section VII).

TABLE I
PRINCIPAL SOURCES USED TO POSITION THIS WORK

Source	Contribution and scale reported	Relevance
[2]	Revised McDonald criteria using MRI evidence of lesion dissemination; international panel consensus	Clinical limits of a single-image classifier
[3]	Longitudinal lesion segmentation challenge; 82 data sets; 5 training and 14 test subjects; two raters	Candidate public data source
[8]	MICCAI 2016 evaluation; 13 methods; 53 cases; 4 centres; 7 experts	Automation trailing experts supports decision-support framing
[9]	Cascaded 3D CNN for MS lesion segmentation	Volumetric lesion modelling (future direction)
[10]	Multi-scale 3D CNN with fully connected CRF for brain lesions	Volumetric context for a 3D extension

Source	Contribution and scale reported	Relevance
[7]	Review of deep learning for automated MS detection from MRI	Overview of methods and open challenges

D. Research Gap

Within the sources reviewed, existing work (a) establishes that MRI lesion evidence is central to MS diagnosis, (b) shows that automatic methods approach but do not reach expert performance on curated benchmarks, and (c) supplies public data for lesion segmentation. What is not addressed in these sources is a documented, requirements-driven reference architecture that places a slice-based MS classifier inside an authenticated, auditable web workflow, together with an evaluation protocol that guards against slice-level leakage and control-cohort confounding. This paper addresses that gap at the level of design and specification, and it states which parts of the system have been validated and which have not.

III. AIM AND CONTRIBUTIONS

The aim of the project is to automate the preliminary classification of brain MRI images as MS or normal and to deliver the result through a web application that supports clinicians. This paper makes three contributions, each limited to what the project evidence supports.

- **Architectural.** A documented four-layer reference architecture (presentation, backend, model serving and persistence) with a defined data flow from MRI submission to report (Section IV).
- **Specification and audit.** A consolidated statement of user classes, functional requirements and non-functional requirements, with an evidence-status audit that separates capabilities that are specified, described or reported as functional (Section VI).
- **Evaluation.** A pre-specified, patient-wise evaluation protocol that addresses slice-level leakage, control-cohort confounding, uncertainty estimation, baselines, ablations and external validation (Section VII).

The paper does not claim a new network architecture, a new dataset or a performance improvement over any published method.

IV. SYSTEM ARCHITECTURE AND REQUIREMENTS

A. Layered Architecture

Turing Body is designed as a standalone web application with four layers (Fig. 1 and Table II). The presentation layer lets users sign in, upload MRI images and view results. The backend validates requests, enforces authentication and authorisation, and brokers communication with the model service. The AI processing layer performs preprocessing and CNN inference behind a Flask API. The persistence layer stores user records, uploaded images, predictions and history in MongoDB. Components communicate through RESTful APIs, which allows each layer to be developed and replaced independently.

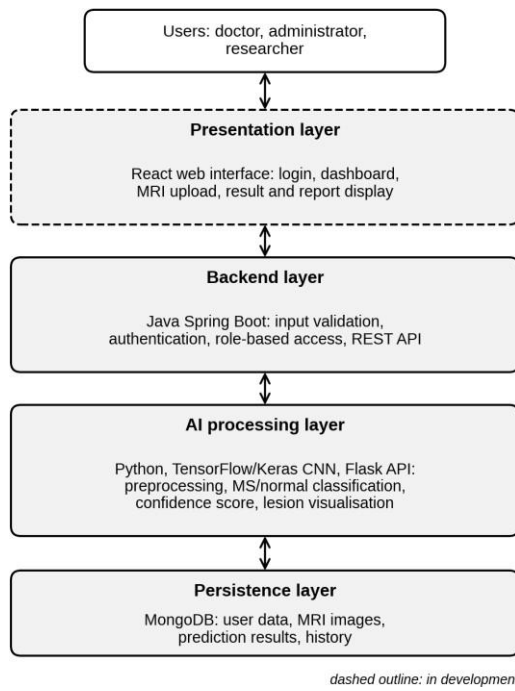


Fig. 1. Layered architecture of Turing Body. The presentation layer is still under development.

TABLE II
LAYERS OF THE TURING BODY ARCHITECTURE

Layer	Technology	Responsibility and documented status
Presentation	React; HTML, CSS, JavaScript	Login, dashboard, MRI upload, result and report display. In development.
Backend	Java, Spring Boot, REST APIs	Input validation, authentication, role-based access, routing to the AI service. Reported functional.
AI processing	Python, TensorFlow/Keras, Flask API	Preprocessing, CNN classification, confidence score. Reported functional.
Persistence	MongoDB	User data, MRI images, prediction results, history. Specified.

B. Data Flow and User Classes

An authorised user signs in and uploads an MRI image. The backend validates the file and forwards it to the AI service, which applies the preprocessing steps described in Section V, runs the CNN and returns the predicted class (MS Detected or Normal) with a confidence score and a lesion visualisation. The backend stores the result and returns a report to the interface, and prior records can be retrieved from the database (Fig. 2). Three user classes are defined (Fig. 3): medical users (doctors and radiologists) who upload scans and read results, administrators who manage accounts, monitoring and data, and researchers or developers who analyse model performance and update algorithms.

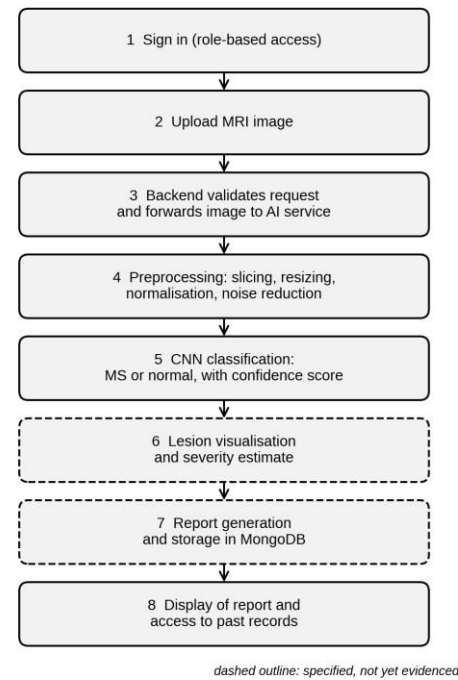


Fig. 2. Request workflow from sign-in to report display. Dashed steps are specified but not yet evidenced.

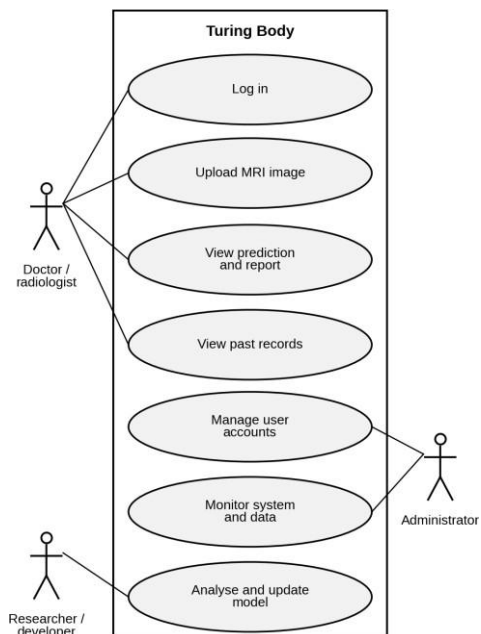


Fig. 3. Use cases by user class, as defined in the requirements specification.

C. Non-Functional Requirements

The requirements specification calls for responsive performance under concurrent use, role-based access control, encrypted transport (HTTPS), protection of stored sensitive data, activity logging, regular backups, modular design and compliance with applicable data-protection rules. These are specified requirements. The project documentation does not report measurements or audits that verify them, and this paper treats them as design intentions.

V. METHODOLOGY

A. Data Preparation

The documented preprocessing chain consists of slice extraction from MRI studies, resizing, intensity normalisation, noise reduction and data augmentation. Slicing converts volumetric studies into two-dimensional inputs suitable for a standard CNN. Normalisation and noise reduction reduce variation caused by acquisition differences, and augmentation increases the effective diversity of the training set. The project documentation does not fix numerical settings such as target image size or augmentation parameters. Section VII therefore requires that every setting be recorded in a versioned configuration file and reported with any results.

B. CNN Classification

A CNN built with TensorFlow/Keras performs feature extraction and binary classification of an image as MS or normal. For a binary classifier with predicted probability p_i and label y_i over N training images, the standard training objective is the binary cross-entropy:

$$L = -(1/N) \sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \quad (1)$$

The layer configuration, optimiser and training schedule are not fixed in the project documentation. They are therefore treated as experimental settings to be recorded and reported, not as properties of the system described here.

C. Lesion Visualisation and Report Generation

The specification requires the report to include lesion visualisation and a severity estimate. The documentation does not state the technique used. For the next implementation iteration we specify gradient-weighted class activation mapping (Grad-CAM) [13], which derives a localisation map from the gradients of the predicted class and needs no lesion-level annotations. Two cautions apply. A Grad-CAM map shows image regions that influenced the classification, so it is not a lesion segmentation and should not be presented as one. A severity estimate such as lesion load requires a validated segmentation step, which is outside the current classifier, so the report should not present a severity value until such a step has been evaluated.

D. Development Process

The project followed a Waterfall model with requirement analysis, system design, implementation, testing, deployment and maintenance phases, which suits a fixed-scope academic project. Testing is described at the unit, integration, system and acceptance levels. The documentation describes these levels in general terms but reports no test cases or outcomes.

E. Statistical Analysis of Results

No inferential statistics are reported because the paper presents no experimental results. The statistical treatment that future results must follow, including subject-level bootstrap confidence intervals, is defined in Section VII.

VI. IMPLEMENTATION STATUS AND EVIDENCE AUDIT

Table III separates what the project documentation specifies, describes or reports as functional, and records what has not been evidenced. The distinction guards against presenting a planned feature as a demonstrated one.

TABLE III

EVIDENCE STATUS OF CAPABILITIES PER PROJECT DOCUMENTATION

Capability	Documented status	Evidence not yet available
MRI upload, authentication, storage	Specified; backend reported functional	Security audit, load test
Preprocessing	Described in methodology	Numerical settings
CNN classification	Described as trained on MRI datasets	Dataset identity, split, architecture, metrics
Lesion visualisation, severity	Specified as report content	Technique, validation
Report generation	Specified	Implementation evidence
Web frontend	Under development; not fully integrated	Completed integration
Testing	Described in general terms	Test cases, outcomes
Cloud deployment	Optional future scope	Not applicable at present

VII. PRE-SPECIFIED EVALUATION PROTOCOL

The protocol in Fig. 4 and Table IV is a plan to be executed and reported by the authors. It is not a description of experiments already performed. Its elements respond to specific ways in which slice-based MS classifiers can produce misleadingly high results.

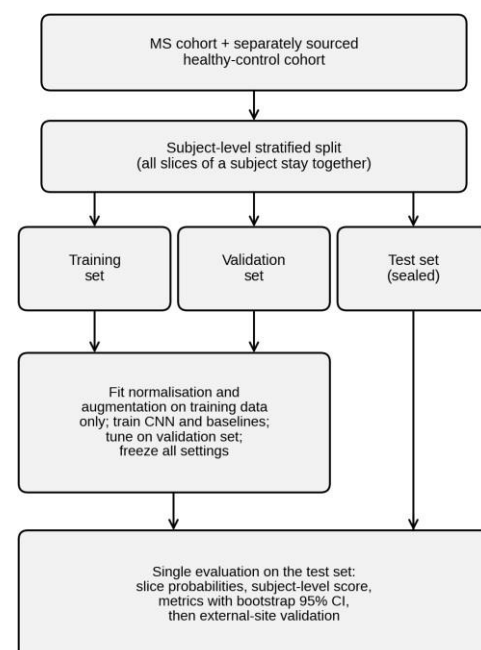


Fig. 4. Pre-specified evaluation workflow with subject-level splitting and a sealed test set.

TABLE IV
EVALUATION PROTOCOL FOR THE TURING BODY

Element	Specification
Data sources	Public MS MRI data such as the ISBI 2015 and MICCAI 2016 resources [3], [8] for MS cases. A healthy-control cohort acquired under comparable protocols must be sourced separately, with licences recorded.
Control-cohort confounding	Report scanner and site distribution per class, and test whether site can be predicted from the images, so that the classifier is not learning acquisition artefacts.
Splitting unit	Subject-level (patient-wise) stratified split. All slices of a subject are placed in one partition only.
Leakage control	Normalisation parameters and augmentation policies are fitted on training data only. The test partition is used once, after all settings are frozen.
Aggregation	Subject score = mean of slice probabilities, Eq. (2). Slice-level and subject-level results are reported separately.
Metrics	Sensitivity, specificity, precision, F1-score, balanced accuracy, ROC-AUC, precision-recall AUC and the confusion matrix, at slice and subject level, Eqs. (3)–(6).
Uncertainty	Subject-level bootstrap with at least 1,000 resamples for 95% confidence intervals of every headline metric.
Baselines	A majority-class predictor and a residual-network baseline [12] trained under identical splits and preprocessing.
Ablations	With and without augmentation; with and without noise reduction; slice sampling strategy.
External validation	Evaluation on at least one data source and site not used in training or tuning.
Explainability check	Compare Grad-CAM maps with expert lesion masks where available, using an overlap measure, and report failure cases.
System measures	End-to-end latency of the deployed workflow on stated hardware, as median and 95th percentile over a stated number of requests.
Reproducibility	Publish the configuration file, random seeds, software versions and, where licences permit, the split lists.

The subject score is the mean of slice probabilities over the slices belonging to that subject:

$$p_s = (1/|S(s)|) \sum_{j \in S(s)} p_j \quad (2)$$

where $S(s)$ is the set of slices of subject s . The classification metrics follow their standard definitions in terms of true and false positives and negatives (TP, FP, TN, FN):

$$\text{Sensitivity} = TP/(TP + FN) \quad (3)$$

$$\text{Specificity} = TN/(TN + FP) \quad (4)$$

$$\text{Precision} = TP/(TP + FP) \quad (5)$$

$$F1 = 2 \cdot \text{Precision} \cdot \text{Sensitivity} / (\text{Precision} + \text{Sensitivity}) \quad (6)$$

VIII. DISCUSSION

The value of the work lies in its explicitness. A student prototype of this kind is easy to describe in terms of “accurate” and “early” detection, yet neither word can be supported without data. The evidence audit makes the current position clear: the architecture and workflow are specified, parts of the backend and model pipeline are

reported as functional, and the performance claim is still open.

Three points shape how the system should be read. First, the word “early” in the project title denotes an aspiration. Showing earlier detection would require cohorts of patients with a clinically isolated syndrome or longitudinal imaging, evaluated against later confirmed diagnoses. A cross-sectional MS-versus-normal classifier does not address this question. Second, slice-level evaluation can overstate performance when slices from one subject appear in both training and test data, because the model can recognise a subject rather than a disease. The subject-level protocol is designed to remove this effect. Third, when MS and control images come from different sources, high accuracy may reflect acquisition differences. The control-cohort checks in Table IV are intended to expose this.

The benchmark finding that automatic algorithms still trailed expert readers on detection and delineation [8] supports the intended role of Turing Body as an adjunct that presents evidence for a clinician to review. The clinical framework [2] reinforces this: the model output can inform, but cannot replace, the integrated clinical and imaging assessment.

IX. LIMITATIONS

- No quantitative performance results are available, so effectiveness, generalisation and robustness are untested.
- The dataset, its size, its class balance, its scanners and the MRI sequences used are not documented in the project report, so the work cannot yet be reproduced.
- The classifier operates on two-dimensional slices and does not use three-dimensional context or longitudinal information.
- The frontend is not fully integrated, so the end-to-end workflow has not been demonstrated as a complete system.
- Lesion visualisation and severity estimation are specified but not validated.
- Security and privacy measures are requirements and have not been audited. No clinical validation or regulatory assessment has been performed.

X. APPLICATIONS AND ETHICAL CONSIDERATIONS

Current application. The system serves as an academic prototype and a reference design for integrating an image classifier into an authenticated web workflow. **Potential applications**, which depend on the validation described above, include triage support, second-reading support and teaching. The system should not be used for clinical decisions in its present state.

Ethical considerations are central because the inputs are medical images. Uploaded scans should be de-identified, including metadata, before storage, and the handling of patient data must follow the data-protection and medical-software rules of the deployment jurisdiction. Performance should be assessed across scanners, sites, ages and sexes to detect bias before any use. Interface design should present

outputs as decision support with their uncertainty, to limit over-reliance on the prediction. Confidence scores should be calibrated before they are shown to clinicians.

XI. FUTURE WORK

The next steps follow directly from the limitations. The first is to execute the protocol in Section VII on public MS data with a properly sourced healthy-control cohort, and to report the results with confidence intervals and external validation. The second is to complete frontend integration and test the end-to-end workflow, including measured latency and a security review. The third is to implement and validate a lesion localisation or segmentation component so that severity estimates are grounded in evaluated outputs. Further directions are three-dimensional and multi-sequence models [9], [10], longitudinal analysis for the early-detection question, integration with hospital systems and electronic health records, and cloud deployment with additional authentication safeguards.

XII. CONCLUSION

This paper presented the design of Turing Body, a web-integrated CNN framework for MRI-based MS screening, covering its layered architecture, requirements, preprocessing pipeline and development process. An evidence audit showed that the backend and model modules are reported as functional, that the frontend remains in progress, and that no quantitative performance results exist yet. We therefore claim no accuracy and instead specify a patient-wise evaluation protocol that addresses slice-level leakage, control-cohort confounding and uncertainty. The principal contribution is a documented reference architecture and a pre-specified, reproducible evaluation plan. The next stage of the project is to execute that plan and report the findings alongside the design presented here.

DECLARATIONS

Author contribution statement. All authors contributed jointly to the conception, development and documentation of the system described in the project report, and to the preparation and approval of this manuscript.

Funding. No external funding for this work is reported in the project documentation.

Conflict of interest. The authors declare that they have no conflict of interest.

Compliance with ethical standards. The project documentation does not describe the collection of new data from human participants, and this paper reports no human-participant or animal experiments. Informed consent and ethical approval are therefore not applicable to the work reported. Any dataset used in future experiments will be used under its published licence and ethics provisions.

Data availability. No datasets were generated or analysed during the study reported here, because the paper presents a system design and evaluation protocol. The public resources discussed in Section II-C are available from their original publishers [3], [8].

AI declaration. Claude (Anthropic) was used to structure, draft and edit this manuscript from the authors' project report and to assist in organising literature. The four figures were drawn programmatically from the architecture documented in the project report and contain no generated imagery or patient data. The authors reviewed the text and figures, verified the cited sources and take full responsibility for the content. No AI tool is listed as an author. The CNN described in the paper is the subject of the research and is separate from the writing assistance.

Acknowledgments. The authors thank their project guide, Prof. Ritu R. Agrawal, and the Department of Computer Science & Engineering (AIML), Parul University, for guidance and support during the project.

REFERENCES

- [1] D. S. Reich, C. F. Lucchinetti, and P. A. Calabresi, "Multiple sclerosis," *N. Engl. J. Med.*, vol. 378, pp. 169–180, 2018.
- [2] A. J. Thompson *et al.*, "Diagnosis of multiple sclerosis: 2017 revisions of the McDonald criteria," *Lancet Neurol.*, vol. 17, no. 2, pp. 162–173, Feb. 2018, doi: 10.1016/S1474-4422(17)30470-2.
- [3] A. Carass *et al.*, "Longitudinal multiple sclerosis lesion segmentation: Resource and challenge," *NeuroImage*, vol. 148, pp. 77–102, Mar. 2017, doi: 10.1016/j.neuroimage.2016.12.064.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [6] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017, doi: 10.1016/j.media.2017.07.005.
- [7] A. Shoeibi *et al.*, "Applications of deep learning techniques for automated multiple sclerosis detection using magnetic resonance imaging: A review," *Comput. Biol. Med.*, vol. 136, Art. no. 104697, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104697.
- [8] O. Commowick *et al.*, "Objective evaluation of multiple sclerosis lesion segmentation using a data management and processing infrastructure," *Sci. Rep.*, vol. 8, Art. no. 13650, Sep. 2018, doi: 10.1038/s41598-018-31911-7.
- [9] S. Valverde *et al.*, "Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach," *NeuroImage*, vol. 155, pp. 159–168, 2017.
- [10] K. Kamnitsas *et al.*, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, 2017, doi: 10.1016/j.media.2016.10.004.
- [11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv. (MICCAI)*, Cham, Switzerland: Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [13] R. R. Selvaraju *et al.*, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.