RESEARCH ARTICLE

Using 3D Convolutional Neural Networks to Classify 3D MRI Images

Waseem Akram[1,*], **Neeraj Kumar**[1] **and Ajay Lakhnotra**[1]

[1] Department of Computer Science & IT, Bhaderwah Campus, University of Jammu, Bhaderwah 182222, India

Abstract

Brain MRI uses a strong static magnetic field to align the nuclei in the region of interest, a gradient field to localize them, and a radiofrequency (RF) pulse to raise them to higher energy states. When the pulse is switched off, the nuclei relax and emit energy, which a coil records to map their positions. MRI scanners are common in hospitals and research centers and are central to the diagnosis of brain disorders. Detecting brain abnormalities requires a reliable way to decide whether a patient has a given condition, and image classification is useful when diseases are hard to separate by eye. In this study, MRI scans of people with and without Alzheimer's disease (AD) are analyzed, and a 3D convolutional neural network (CNN) is trained to tell the two groups apart. The model uses the 3D MRI volumes of 199 participants (139 with AD and 60 cognitively normal) for binary classification, and accuracy and loss were recorded after every epoch. Because of hardware limits, the model was trained on down-sampled volumes. Even with data augmentation and preprocessing, the validation curves remained unstable, so a larger dataset, higher-resolution images and more computing power are needed for more reliable results.

Keywords: convolutional neural networks, deep neural networks, artificial neural networks, Alzheimer's disease, TensorFlow, pooling.

1 Introduction

Alzheimer's disease (AD) is an irreversible, degenerative form of dementia that mainly affects older people and is the most common type of dementia [1]. It begins slowly with disorientation and memory loss and then spreads to other cognitive functions. Patients often become more sensitive to nonverbal cues and lose the ability to speak.

The brain changes markedly in AD: it shrinks, the ventricles enlarge, and the cells that produce acetylcholine, a neurotransmitter essential for memory, are damaged. Lower acetylcholine levels and the accumulation of plaques and tangles in the brain follow. The progression is usually divided into seven stages whose symptoms worsen over time: (1) no impairment; (2) very mild decline in cognitive function; (3) mild decline; (4) moderate decline; (5) moderately severe impairment; (6) severe decline; and (7) very severe impairment. Two forms are recognized:

Submitted: 28 July 2026
Revised: 17 October 2026
Accepted: 17 November 2026
Published: 26 November 2026

Vol. 2, **No.** 4, 2026.
10.62762/BISH.2026.482168

***Corresponding author:**
✉ Waseem Akram
akrambtt101@gmail.com

Citation
Akram, W., Kumar, N., & Lakhnotra, A. (2026). Using 3D Convolutional Neural Networks to Classify 3D MRI Images. *Biomedical Informatics and Smart Healthcare*, 2(4), 191–200.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

early-onset AD, which is rare and appears before the age of 65, and late-onset AD, which is the most common and appears after 65. The disease is more prevalent in aging populations, and the number of cases grows as people live longer.

There is no proven cure. Drugs can reduce the rate at which the disease progresses, but treatment is mostly aimed at supporting the patient and the family.

Python is a high-level, easy-to-use programming language that has become a standard environment for scientific computing, because its numerical and array-processing libraries make it practical to load, transform and analyze large datasets such as imaging volumes [2].

TensorFlow is an open-source library created for machine learning in numerical computing. It supports data collection, model training, prediction and the refinement of results. It offers a simple Python front-end API and uses high-performance C++ underneath. TensorFlow is used to train and run deep neural networks for tasks such as classifying handwritten digits, recognizing images, producing word embeddings, building recurrent networks, developing sequence-to-sequence models for natural language processing and machine translation, and even simulating systems described by partial differential equations (PDEs). The same models can be used for both training and real-time prediction in large-scale production [3].

Deep learning is a branch of artificial intelligence (AI) in which models learn representations of data through several layers [4]. Its main goal is to automate prediction. Unlike classical machine-learning algorithms, which usually rely on hand-crafted features, deep learning algorithms are stacked in layers of increasing complexity and abstraction. The training data may be, for example, a set of photos labeled by people as “dog” or “not dog”. From these labels the model builds a prediction model and a set of features for recognizing dogs.

Artificial neural networks (ANNs) are computing systems loosely modeled on biological neural networks, although they cannot reproduce them exactly. They “learn” from examples instead of following predetermined rules. An ANN is made of interconnected artificial neurons, and each connection, like a synapse, carries signals between neurons. After a neuron has processed a signal, it can pass it on to the neurons connected to it.

A convolutional neural network (CNN) is a deep network designed to improve on the multilayer perceptron, whose fully connected architecture often leads to overfitting [5]. Instead of linking every neuron to every neuron of the next layer, a CNN looks for patterns and hierarchies in the data and gradually assembles simple patterns into larger ones. Its structure is less complex and less densely connected than that of a fully connected network, so it generalizes better to new data. A CNN has an input layer, an output layer and several hidden layers. The hidden layers usually include convolutional layers, which apply operations such as multiplication or dot products to the input, followed by an activation function such as ReLU and by further layers such as pooling, fully connected and normalization layers. They are called hidden because the activation function and the subsequent operations mask their inputs and outputs. The weights of the network are adjusted by back-propagation to improve prediction accuracy [6]. Although these operations are called convolutions, they are in fact cross-correlations (sliding dot products), which affects how the matrix indices are interpreted and how the weights are computed for particular data points. CNNs have become a standard tool in medical image analysis [7, 8].

A 3D CNN extends this design to volumetric data such as stacks of MRI slices or series of 2D frames, and it is especially good at finding patterns in such data [9].

2 Related Work

A system that detects AD from MRI scans could improve both clinical practice and medical research, and interest in the topic has grown [1]. The task is difficult, mainly because neighboring stages of the disease show overlapping symptoms. Deep 3D CNNs have shown considerable potential for such problems. Obtaining high-quality MRI volumes in the .nii format is one of the main obstacles and can affect performance.

Earlier work has applied CNNs to MRI-based AD classification in several ways. Basaia et al. [10] classified AD and mild cognitive impairment (MCI) that converts to AD from a single structural MRI, and El-Assy et al. [11] proposed a CNN architecture for early detection and classification of AD from MRI, reporting accuracies above 99%. Ebrahimi and Luo [12] compared slice-based and 3D voxel-based approaches. Hippocampus-based 3D models include DenseCNN [13] and the 3D CNN on sMRI and MD-DTI of Khvostikov et al. [14]. Liu et al. [15]

tested the generalization of a 3D CNN across datasets, and Chen et al. [16] used a 3D-ResNet. Wen et al. [17] reviewed over 30 CNN studies and showed that results are hard to compare because of differences in participant selection and data-leakage control, and Zia-ur-Rehman et al. [18] reviewed deep learning for AD detection in 3D imaging.

In this work, a 3D CNN with data augmentation is used to separate AD patients from cognitively normal (CN) subjects. The study makes two contributions: first, a 3D CNN classification pipeline for MRI volumes; second, a description of how random-rotation augmentation is applied to the training volumes.

2.1 Problems and Weaknesses of Current Systems

Reproducible evaluation of CNNs for AD classification is still difficult [17], and simple, openly documented Python pipelines that classify raw 3D MRI volumes with TensorFlow are rarely described in detail.

2.2 Problem Identification

Without a physician, it is hard to tell whether a patient has AD, so the presence of AD has to be recognized from the patient's MRI scan with machine learning. Identifying the stage of the disease is a harder problem and is left for future work.

2.3 Objective

The aim is a simple, user-friendly tool with one main function: deciding whether a brain MRI scan shows AD or not. The system takes a dataset of brain MRI scans as input, and users can also supply their own scans as test data. It preprocesses the data, passes it to the trained model and returns the predicted class.

3 Process

The model is built with supervised machine learning, in which a model is trained on input data and the matching output labels. Depending on the method, the procedure can vary. Here we concentrate on neural networks, and the model-building process usually has the following steps.

3.1 Model Initialization and Layers

The model is first created with `Sequential()`. Layers are then added one by one with calls such as `Type_of_layer()`.

3.2 Model Compilation

After the layers are added, the model is compiled. TensorFlow and Keras work together to build the model at this step, and the loss function and optimizer are specified as `compile(loss=..., optimizer=...)` [19]. The loss function measures how accurate each prediction is [6]. The data must be scaled properly before training. The console shows the progress of training, and the final accuracy is reported when it finishes.

3.3 Testing and Use

Once trained, the model is tested on data that were not used for training. In this study the validation subset serves this purpose and gives an estimate of the accuracy; an independent test set is left for future work. If training and testing give dependable results, the model can be saved with `model.save("name_of_file.h5")` and later used in real-world applications to evaluate new data.

4 Project Management

4.1 Process Model

The essential phases of the project are collecting requirements, planning, implementing with unit testing, integrating with system testing, and running the system (Figure 1). With an incremental approach, flaws in the requirements are often found early, when the related software component is added to the system, and are therefore easier to correct. This approach is more adaptable than the traditional waterfall method. It splits development into several cycles, which gives a "multi-waterfall" structure, and each cycle passes through requirements, design, implementation and testing in smaller iterations. The Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) are two common ways of planning and scheduling software projects.

4.2 Hardware and Software Requirements

The software environment consists of the Anaconda platform, Jupyter Notebook and the Python programming language. The hardware was an Intel Core i3-8100 CPU at 3.60 GHz with 16 GB or 32 GB of RAM, running 64-bit Windows 10 on an x64-based processor.

5 System Design

5.1 Description of the Algorithm

The project uses a 3D CNN to classify three-dimensional images. As described in the

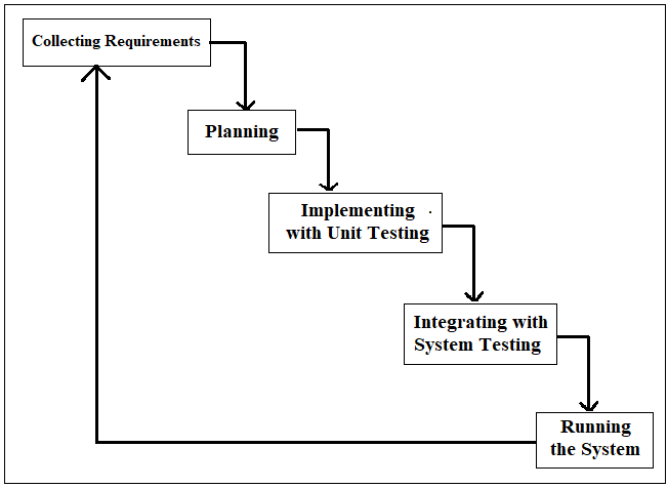


Figure 1. Phases of project development.

Introduction, a CNN has an input layer, hidden layers and an output layer. The convolutional layers scan the input with small filters called kernels, which typically use the Frobenius inner product to detect patterns. An activation function such as the rectified linear unit (ReLU) then adds non-linearity, and the kernel produces a feature map that serves as the basis for later layers. Normalization layers keep training stable, fully connected layers combine features into predictions, and pooling layers reduce the spatial size of the data.

5.2 Function of the System

The use case diagram is shown in Figure 2.

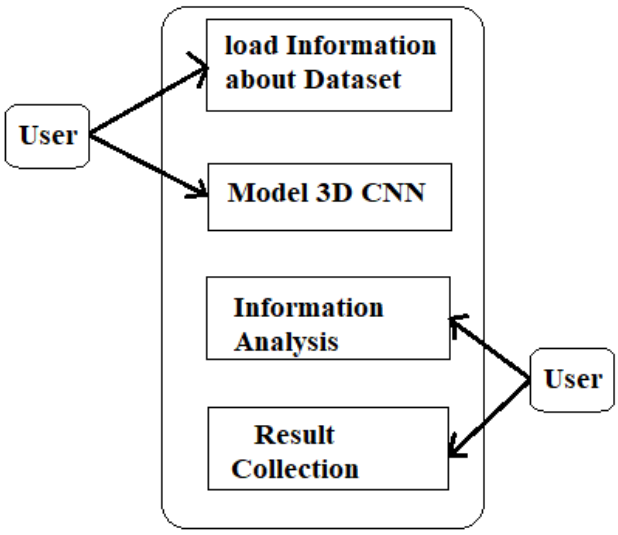


Figure 2. Use case diagram of the system.

5.3 Data Modeling

5.3.1 Methodology

Data modeling has four main steps. Data are first collected from various sources. Preprocessing then organizes and cleans the data and removes errors and inconsistencies. Feature extraction identifies the most informative content and simplifies the data (in the 3D CNN this is done by the convolutional layers). Finally, in the classification stage, the algorithm assigns labels to the data. Figure 3 shows these steps.

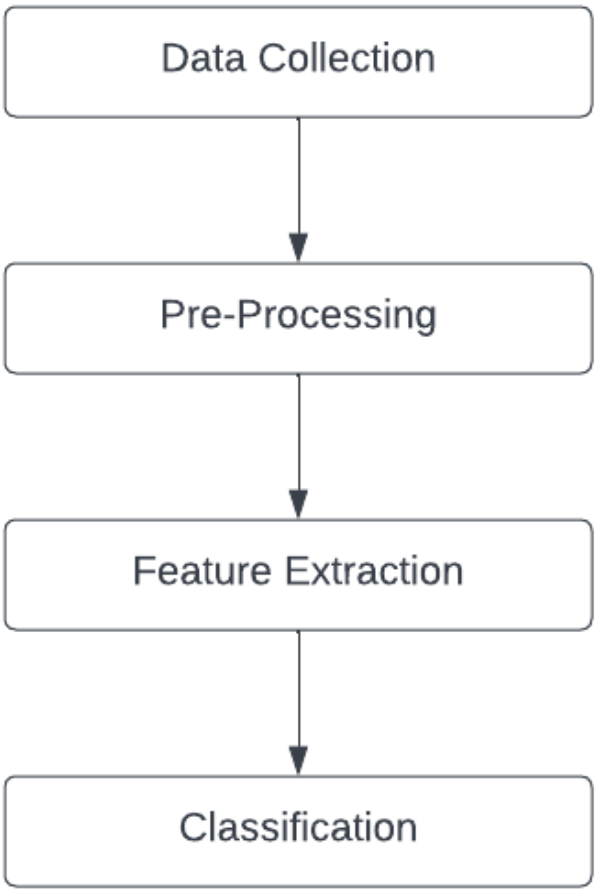


Figure 3. Stepwise methodology for data modeling.

5.3.2 Data Preparation

The preparation steps follow the 3D CT classification example of Zunair [20], adapted here to MRI. The volumes are rotated by 90 degrees so that they have a consistent orientation. The MRI intensities are normalized to the range 0 to 1. The width, height and depth are then adjusted to the required size.

5.3.3 Augmentation and Channel Dimension

During training the volumes are rotated at random to augment the MRI scans. Each volume is a rank-3 tensor with the shape (height, width, depth). To allow 3D convolutions, an extra dimension of size 1 is added

along the fourth axis, so that a batch has the shape (samples, height, width, depth, 1).

5.3.4 *Splitting the Data*

The image arrays and their labels are divided into two parts: a training set (70%), which is used to train the model, and a validation set (30%), which is used to check the trained model on volumes that were not used for fitting.

5.3.5 *Sequence Diagram*

The sequence of operations is given in Figure 4.

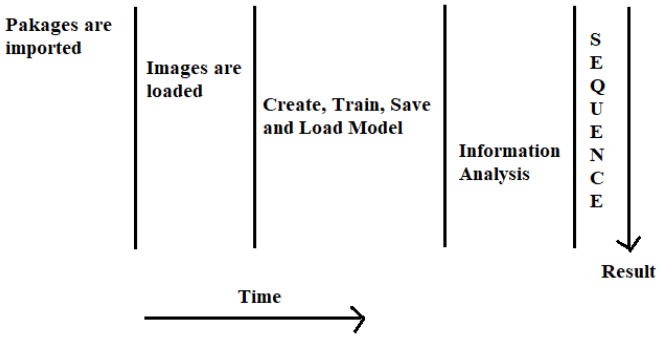


Figure 4. Sequence diagram of the operations performed.

5.3.6 *Data Flow Diagram*

A data flow diagram (DFD) shows how data move through a system: the different steps, the source of the data and the next destination. Data entering the system appear at the start of the diagram, followed by the processing stages and the final destination. Figures 5 and 6 show the level 0 and level 1 DFDs, respectively.

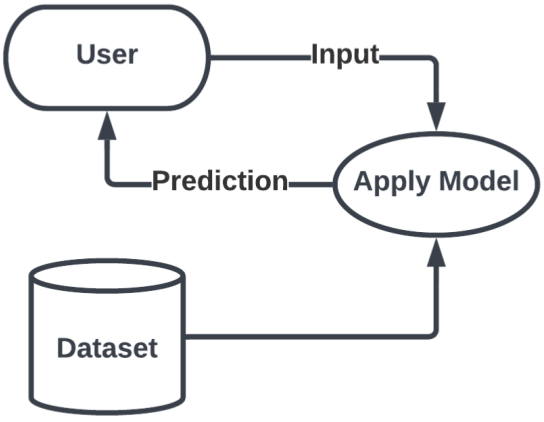


Figure 5. Data flow diagram (level 0).

5.3.7 *Dataset*

The Alzheimer's Disease Neuroimaging Initiative (ADNI) provides standardized, multi-site structural MRI data related to Alzheimer's disease [21]. The

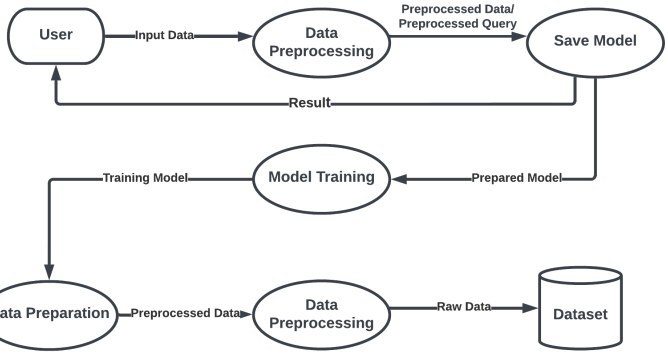


Figure 6. Data flow diagram (level 1).

volumes are distributed in NIfTI or DICOM format. The 3D MRI volumes used in this work come from the ADNI baseline data and are stored in NIfTI format. They fall into two groups: CN subjects and subjects with AD. The study used 199 subjects, 139 with AD and 60 CN.

5.4 3D CNN Architecture and Training Configuration

A 3D CNN was built with TensorFlow and its Keras API for the binary classification of AD and CN subjects. The original MRI volumes had a spatial resolution of about $166 \times 256 \times 256$ voxels. To reduce the computational requirements and allow training on local machines and Google Colab, each volume was resized to $128 \times 128 \times 64$ voxels and normalized so that the intensity values are consistent across the dataset. The scans, stored in NIfTI format, were loaded with the NiBabel library. A single-channel dimension was then added to each volume, which gives an input shape of $128 \times 128 \times 64 \times 1$. The data were divided into training and validation subsets with a 70:30 ratio. The training subset was used to learn the model parameters, and the validation subset was used to monitor how well the model generalizes during training. Augmentation with random rotations was applied only to the training data, so that validation was performed on unmodified MRI volumes.

The network consists of three-dimensional convolutional, batch normalization and pooling layers, followed by fully connected layers. The convolutional layers use the ReLU activation function, and batch normalization after each convolutional layer improves the stability of training. Three-dimensional max-pooling layers progressively reduce the spatial dimensions of the feature maps and the computational cost. The extracted features are flattened and passed to the fully connected layers. The final dense layer has a single neuron with a sigmoid activation, which gives

a probability between 0 and 1. The architecture can be summarized as: input $(128 \times 128 \times 64 \times 1) \rightarrow$ three blocks of [3D convolution $\rightarrow$ batch normalization $\rightarrow$ ReLU $\rightarrow$ 3D max pooling] $\rightarrow$ flatten $\rightarrow$ dense layer $\rightarrow$ output dense layer (sigmoid).

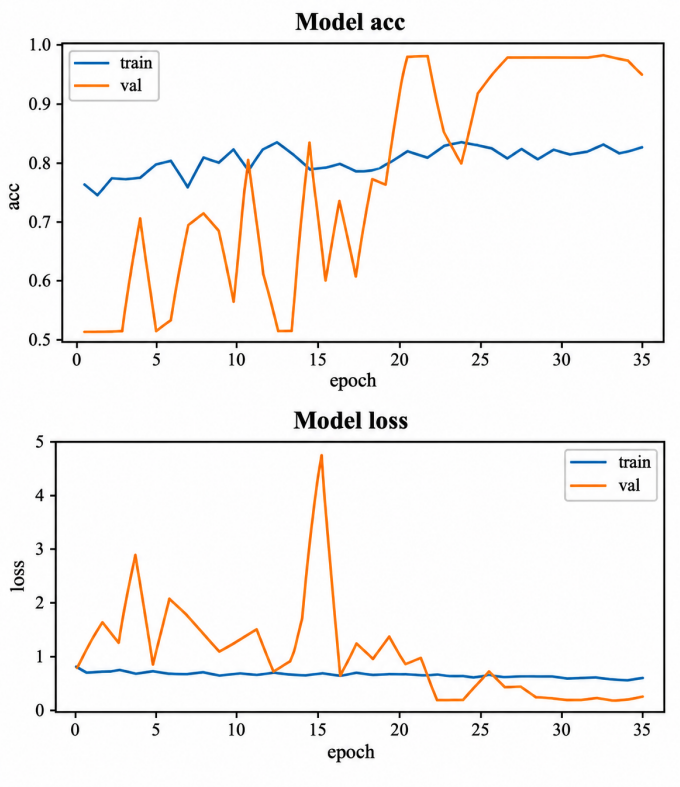


Figure 7. Training and validation accuracy (top) and loss (bottom) per epoch.

The model was set to train for 100 epochs, although the curves in Figure 7 cover only about 35. Binary cross-entropy was used as the loss function because the task is binary classification, and the network parameters were updated with the Adam optimizer [22]. Model checkpoints were saved during training so that the best-performing model could be selected. Training and validation accuracy and loss were recorded after each epoch and plotted with Matplotlib [23]. The trained model then predicted the class of the validation volumes. The sigmoid output was read as the probability of AD, and the predicted class was obtained with a threshold of 0.5.

In Figure 7, training accuracy rose from about 0.76 to 0.83 and training loss fell from about 0.8 to 0.6. Validation accuracy and loss fluctuated strongly: during the first 20 epochs validation accuracy varied between about 0.51 and 0.83 and then rose to about 0.98, ending near 0.95, and the validation loss peaked at about 4.75 near epoch 15 before dropping to about 0.2. These curves do not show a clear overfitting trend. Validation accuracy above training accuracy suggests that the validation set is small, and the initial validation accuracy of about 0.51 indicates that its class proportions differ from those of the full dataset (139 AD and 60 CN), so its composition should be checked.

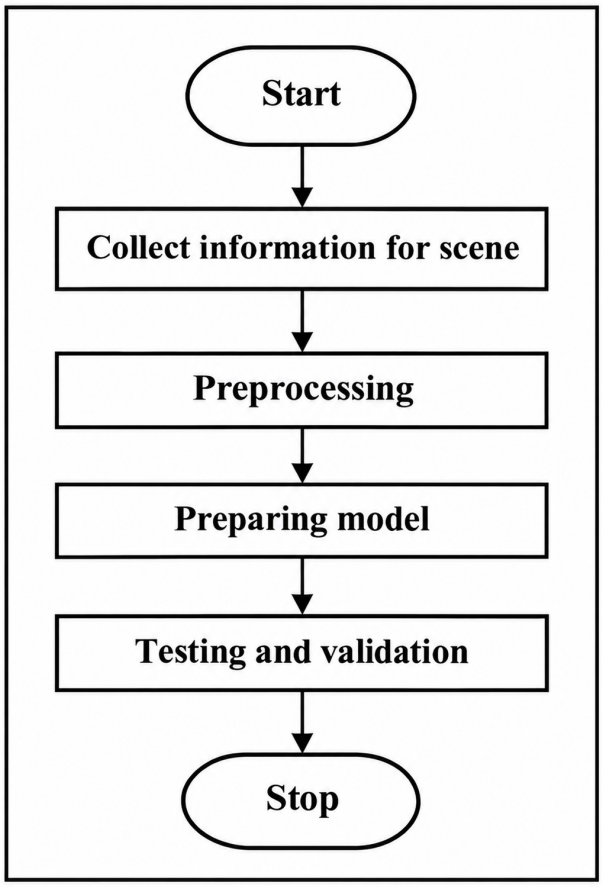


Figure 8. Flowchart of prediction from the trained model.

The complete model-building and prediction workflow is shown in Figures 8 and 9. The workflow starts with MRI data collection, followed by NIfTI loading, preprocessing, resizing, normalization, dataset splitting, augmentation of the training data, model training and validation. For a new patient's scan, the same preprocessing is applied before the scan is passed to the trained model. The model then returns a probability score and the corresponding binary prediction, which can be used to generate the prediction report.

6 Discussion

Collecting data and choosing an algorithm that suits them are the most demanding steps of this kind of research, and researchers spend much time and effort testing algorithms. The tool described here is meant to reduce that effort and to help researchers, physicians

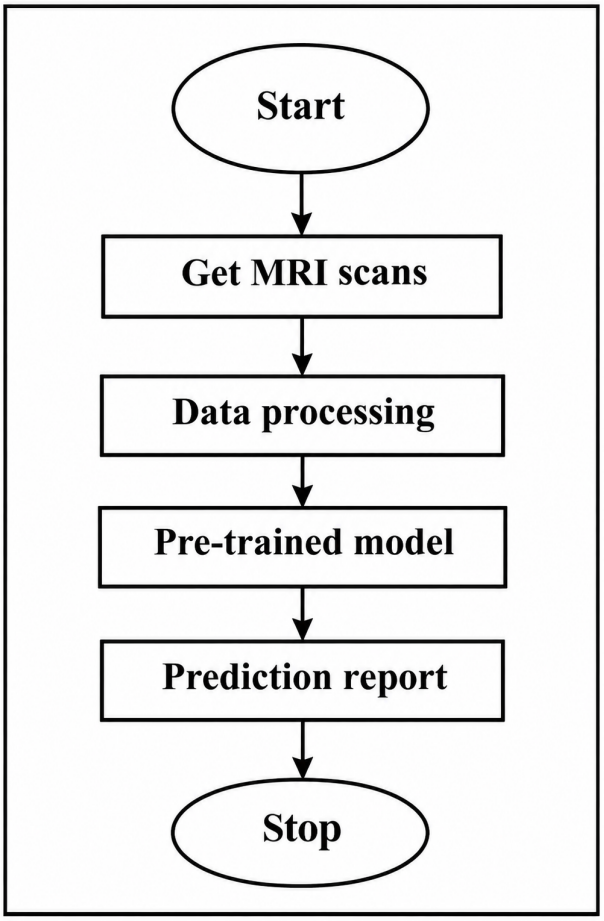


Figure 9. Flowchart of the model-building process.

and health officials screen MRI scans for AD. Its accuracy is limited by the small dataset, and it has not been validated for clinical use.

7 Testing

7.1 Test Cases and Results

Correctness is the degree to which software operates without errors, and testing is necessary for any system to succeed. System testing checks whether the whole system meets the requirements and expectations of the user, on the principle that the system works as intended if each of its components does. Program testing aims to find and correct logical and syntactic flaws by comparing actual and expected outputs, and it also checks that wrong input is handled properly. Minor errors that are ignored can grow into serious problems.

The data augmentation function was applied only to the training data loader. Random rotations were used to increase the variability of the training samples, whereas the validation volumes were evaluated without random augmentation. Both subsets were normalized before being passed to the model. The

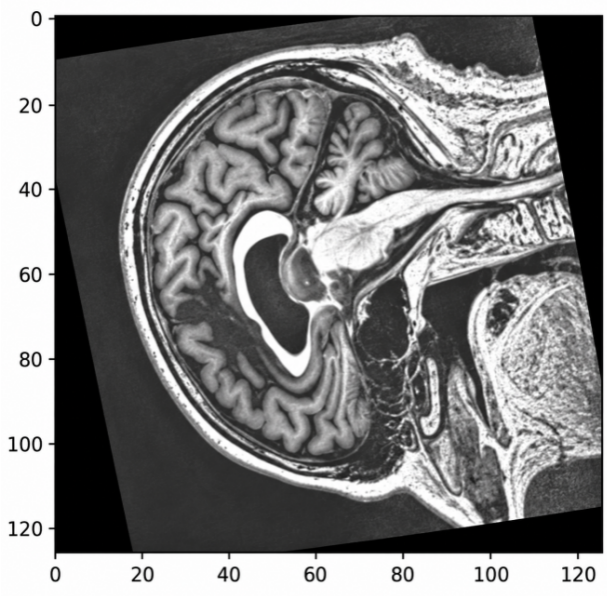


Figure 10. A slice of an augmented (randomly rotated) MRI volume.

intensity values of both sets are already between 0 and 1. An MRI scan in the data loader has the dimensions (128, 128, 64, 1), and Figure 10 shows one augmented slice. A montage of the slices of one volume, with 4 rows and 10 columns (40 slices), is shown in Figure 11.

7.2 Model Performance

Training and validation accuracy and loss are shown in Figure 7. Accuracy is a reliable measure only if the classes in the validation set are balanced. With 139 AD and 60 CN subjects in total, this has to be verified for the validation set, and sensitivity, specificity and the area under the ROC curve should be reported as well.

After training, the best weights are loaded with `model.load_weights`, and `model.predict` is used on one sample from the validation set. The output is a single sigmoid probability that the sample shows AD, and the probability of CN is its complement. The two values are combined with the labels “CN” and “AD” to give a readable result. For the example scan, the model was 0.00% confident that the scan was CN and 100% confident that it was AD. One sample only illustrates the output and does not measure performance.

8 Future Scope

The project gives a basis for further work, and its framework can be extended by researchers and students. Possible directions are as follows.

- The model currently separates two classes. With larger datasets it could be extended to more

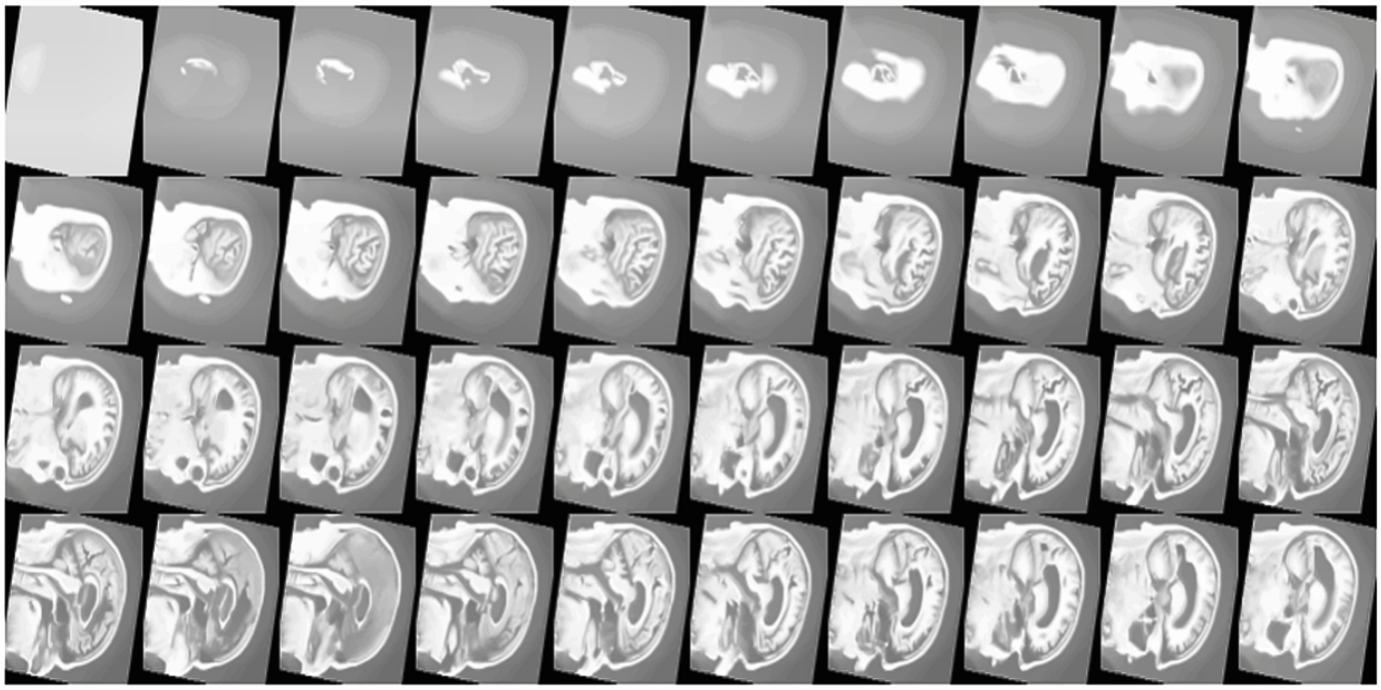


Figure 11. Montage of 40 slices from one MRI volume.

classes, conditions and scan types, which would allow a more thorough study.
- The 3D CNN architecture is rather simple, and more advanced methods and new designs may improve its performance.
- A larger and more diverse MRI dataset would improve the accuracy, robustness and generalizability of the model across patient groups.
- If datasets that describe the stages of the disease become available, the model could predict stages by using softmax activation and categorical cross-entropy loss.
- Connecting the model directly to MRI scanners could allow real-time analysis and prediction while scans are being acquired.

9 Conclusion

This study presented a 3D CNN for classifying brain MRI volumes of AD patients and cognitively normal subjects. The model was built with TensorFlow and Keras and trained on down-sampled volumes of 199 ADNI participants. Training accuracy reached about 0.83, and the validation curves were unstable, so the results do not yet support a reliable accuracy estimate. The study made no comparison with other methods. Future work will use a larger dataset, higher-resolution images, a subject-level split with an independent test set, and metrics beyond accuracy. The authors also plan to isolate specific image characteristics and feed them to the model to improve classification.

Data Availability Statement

The MRI data analyzed in this study are available from the ADNI database (https://adni.loni.usc.edu) after a data-use application.

Funding

This work was supported without any funding.

Acknowledgments

Data collection and sharing for ADNI was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (https://adni.loni.usc.edu). The ADNI investigators contributed to the design and implementation of ADNI and/or provided data but did not participate in the analysis or writing of this report. A complete listing of ADNI investigators is available at http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

This study used de-identified MRI data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database, which were accessed after a data-use application and in accordance with the ADNI data use agreement. ADNI was approved by the institutional review boards of all participating sites, and written informed consent was obtained from all participants or their authorized representatives. No new data were collected from human participants or animals for this study.

References

[1] Jo, T., Nho, K., & Saykin, A. J. (2019). Deep learning in Alzheimer's disease: Diagnostic classification and prognostic prediction using neuroimaging data. *Frontiers in Aging Neuroscience, 11*, 220. [CrossRef]

[2] Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering, 9*(3), 10-20. [CrossRef]

[3] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: a system for Large-Scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283). https://dl.acm.org/doi/10.5555/3026877.3026899

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. [CrossRef]

[5] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324. [CrossRef]

[6] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging, 9*(4), 611-629. [CrossRef]

[7] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60-88. [CrossRef]

[8] Yu, H., Yang, L. T., Zhang, Q., Armstrong, D., & Deen, M. J. (2021). Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing, 444*, 92-110. [CrossRef]

[9] Payan, A., & Montana, G. (2015). Predicting Alzheimer's disease: A neuroimaging study with 3D convolutional neural networks. *arXiv preprint arXiv:1502.02506*. [CrossRef]

[10] Basaia, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., ... & Alzheimer's Disease Neuroimaging Initiative. (2019). Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. *NeuroImage: Clinical, 21*, 101645. [CrossRef]

[11] El-Assy, A. M., Amer, H. M., Ibrahim, H. M., & Mohamed, M. A. (2024). A novel CNN architecture for accurate early detection and classification of Alzheimer's disease using MRI data. *Scientific Reports, 14*(1), 3463. [CrossRef]

[12] Ebrahimi, A., & Luo, S. (2021). Convolutional neural networks for Alzheimer's disease detection on MRI images. *Journal of Medical Imaging, 8*(2), 024503. [CrossRef]

[13] Wang, Q., Li, Y., Zheng, C., & Xu, R. (2020). DenseCNN: A densely connected CNN model for Alzheimer's disease classification based on hippocampus MRI data. *AMIA Annual Symposium Proceedings, 2020*, 1277-1286. https://pmc.ncbi.nlm.nih.gov/articles/PMC8075423/

[14] Khvostikov, A., Aderghal, K., Benois-Pineau, J., Krylov, A., & Catheline, G. (2018). 3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies. *arXiv preprint arXiv:1801.05968*. [CrossRef]

[15] Liu, S., Masurkar, A. V., Rusinek, H., Chen, J., Zhang, B., Zhu, W., ... & Razavian, N. (2022). Generalizable deep learning model for early Alzheimer's disease detection from structural MRIs. *Scientific Reports, 12*(1), 17106. [CrossRef]

[16] Chen, D., Yang, H., Li, H., He, X., & Mu, H. (2025). MRI-based diagnostic model for Alzheimer's disease using 3D-ResNet. *Biomedical Physics & Engineering Express, 11*(3), 035031. [CrossRef]

[17] Wen, J., Thibeau-Sutre, E., Diaz-Melo, M., Samper-González, J., Routier, A., Bottani, S., ... & Alzheimer's Disease Neuroimaging Initiative. (2020). Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation. *Medical Image Analysis, 63*, 101694. [CrossRef]

[18] Zia-ur-Rehman, Awang, M. K., Ali, G., & Faheem, M. (2024). Deep learning techniques for Alzheimer's disease detection in 3D imaging: A systematic review. *Health Science Reports, 7*(9), e70025. [CrossRef]

[19] Chollet, F. (2021). Deep Learning with Python (2nd ed.). Manning Publications. https://www.manning.com/books/deep-learning-with-python-second-edition

[20] Zunair, H. (2020). 3D image classification from CT scans. Keras Code Examples. https://keras.io/examples/vision/3D_image_classification/

[21] Jack Jr, C. R., Bernstein, M. A., Fox, N. C., Thompson,

P., Alexander, G., Harvey, D., ... & Weiner, M. W. (2008). The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, *27*(4), 685-691. [CrossRef]

[22] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations* (*ICLR*). [CrossRef]

[23] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90-95. [CrossRef]



Waseem Akram completed his M.Tech. in Information Technology at the School of Engineering & Technology, Central University of Kashmir, J&K, India, in 2019. His expertise lies in computer science and technology, contributing to academics and research. (Email: akrambtt101@gmail.com)



Neeraj Kumar received his Ph.D. from the Department of Computer Science & IT, University of Jammu, J&K, India, in 2022. His research interests include machine learning, deep learning, medical diagnosis, image analysis and classification, and GANs. (Email: katal.niraj@gmail.com)



Ajay Lakhnotra completed his MCA degree at the Department of Computer Science & IT, University of Jammu, J&K, India, in 2012, and is currently pursuing his Ph.D. in the same department. His research interests include drone technology, machine learning, deep learning, and image processing. (Email: ajay_lakhnotra@jammuuniversity.ac.in)