Enhancing the Detection of Alzheimer's Disease Using Magnetic Resonance Imaging Data through Convolutional Neural Networks

JUIHSUAN WONG, Earlham College, USA

Alzheimer's disease (AD) is a significant global health challenge. This study harnesses convolutional neural networks (CNNs) to improve the accuracy of Alzheimer's detection using magnetic resonance imaging (MRI) data. By optimizing CNNs, we aim to refine diagnostic processes and contribute to early and accurate AD diagnosis.

Additional Key Words and Phrases: Alzheimer's disease, MRI, convolutional neural networks, deep learning, machine learning

1 Introduction

Alzheimer's disease (AD) is one of the most significant global health challenges of our age, affecting millions worldwide. As a progressive neurodegenerative disorder, AD manifests through cognitive decline and memory loss, severely impacting the quality of life of patients and their families. The early and accurate diagnosis of AD is not merely a medical necessity; it is critical for effective treatment planning and improving patient outcomes. In this context, magnetic resonance imaging (MRI) plays an essential role. MRI scans provide detailed images of the brain, enabling the detection of early signs of AD before severe symptoms appear. These early interventions can potentially slow the progression of the disease, highlighting the importance of precision in the diagnostic process [6].

This project seeks to harness the power of convolutional neural networks (CNNs), a sophisticated deep learning technique, to enhance the precision of AD detection using MRI data [3, 7]. Despite the advancements in machine learning (ML) and its application in medical imaging, challenges remain in achieving the high accuracy needed for clinical use. By applying CNNs, known for their effectiveness in image recognition and classification [2], this project aims to refine diagnostic processes. We will explore how CNNs can be optimized to better identify the characteristic patterns of AD in brain images, thus supporting earlier and more accurate diagnoses than currently possible. This initiative not only aims to leverage cutting-edge technology but also to contribute significantly to the field by improving diagnostic accuracy, which is paramount for the timely and effective treatment of AD.

2 Background

2.1 Current state of AD detection

MRI plays a crucial role in diagnosing AD, allowing for the identification of characteristic brain patterns linked to the disease. While traditional diagnostic methods rely on visible symptoms and MRI imaging, the integration of ML offers a path to earlier and more accurate diagnoses. Recent advancements in ML and deep learning have significantly improved the accuracy of AD classification from MRI data, addressing some limitations of earlier techniques such as subjectivity in image interpretation and variability in diagnostic criteria.

Author's Contact Information: JuiHsuan Wong, Earlham College, Richmond, IN, USA, swong21@earlham.edu.

2.2 Machine learning in medical imaging

Various ML algorithms have shown promise in enhancing AD detection through MRI. Support vector machines (SVMs) [5], CNNs, decision trees [1], and ensemble methods like random forests are particularly noted for their effectiveness in classifying complex patterns in MRI data. These algorithms are capable of processing high-dimensional data and extracting subtle features that may not be discernible through traditional analysis. However, challenges remain, including the need for substantial training data, computational intensity, and the risk of overfitting, which necessitates careful tuning and validation of the models.

2.3 Related works

Studies have demonstrated the superior performance of deep learning algorithms, particularly CNNs, in detecting AD by analyzing MRI scans. These models excel at capturing intricate spatial hierarchies in imaging data, leading to high classification accuracy. Notable research has explored various architectures and hybrid models that combine the strengths of multiple learning algorithms to further enhance diagnostic accuracy. Despite these advancements, the field continues to face challenges related to the generalizability of models across different populations and imaging protocols, highlighting an area for ongoing research and development [4, 8].

3 Methodology

3.1 Data collection and preprocessing

Our study leverages MRI images from the *Alzheimer's Dataset (4 Class of Images)* from Kaggle, organized into binary classification:

- **Healthy**: Non-Demented
- **Unhealthy**: Very Mild, Mild, and Moderate Demented

We divided the data into training and testing sets, ensuring consistent partitioning for reproducibility across all models. We used `ImageDataGenerator` to load images, with the following preprocessing steps:

- **Rescaling**: Each pixel intensity value was normalized to the range [0, 1] by dividing by 255.
- **Image resizing**: All images were resized to 224×224 pixels to match the input size required by models like VGG16 and EfficientNet.
- **Batch processing**: The data was processed in batches of 32 images to optimize memory usage during training.
- **Consistent shuffling**: Shuffling was disabled to maintain consistent ordering across models, particularly for machine learning algorithms like SVM and Random Forest.

```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    directory=train_dir,
```

Table 1. Pros and Cons of Various Machine Learning Algorithms

Algorithm	Pros	Cons
SVM	High accuracy with clear margin separation; effective in high-dimensional spaces	Requires careful parameter tuning; may perform poorly with noisy datasets
CNNs	Exceptional at capturing spatial hierarchies in image data; automatically detects important features	Requires large amounts of labeled data; computationally intensive
Random Forest	Handles high-dimensional data well; provides insights into feature importance	Can be computationally intensive; risk of overfitting without proper tuning
Decision Trees	Can handle both numerical and categorical data; easy to interpret	Prone to overfitting with complex trees; may not capture complex relationships effectively
Gradient Boosting	High accuracy by combining multiple weak models; flexible, optimizing different loss functions	Prone to overfitting if not controlled; time-consuming to train
Naive Bayes	Fast and efficient for large datasets; performs well with independent features	Assumes feature independence, reducing accuracy in complex scenarios; struggles with mixed data types
Linear Regression	Simple to implement and understand; useful for analyzing feature relationships	Assumes linear relationships, unsuitable for complex patterns; sensitive to outliers
XGBoost	High performance and scalability; supports handling missing data	Complex to tune due to many hyperparameters; may overfit with deep trees
Voting Classifier	Improves accuracy by combining multiple models; reduces overfitting risk	Results in a complex, less interpretable model; requires careful model selection

```
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)

test_generator = test_datagen.flow_from_directory(
    directory=test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)
```

Listing 1. Data preprocessing with ImageDataGenerator

3.2 Models implemented

We experimented with a range of models to determine the best-performing one for this binary classification task. Each model was evaluated using accuracy, precision, recall, and F1-score to ensure a balanced assessment of both healthy and unhealthy predictions.

3.2.1 Logistic regression. We flattened the input images and trained a logistic regression model using the maximum iteration limit of 1000 to ensure convergence.

```
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train_flat, y_train.argmax(axis=1))
```

Listing 2. Training logistic regression model

3.2.2 Random forest (RF). A Random Forest Classifier with 100 trees was used to capture non-linear patterns. The model provided insights into feature importance, which can be helpful for interpretability.

```
rf_model = RandomForestClassifier(n_estimators=100)
rf_model.fit(X_train_flat, y_train.argmax(axis=1))
```

Listing 3. Training Random Forest model

3.2.3 Support vector machine (SVM). We used a linear kernel SVM to classify MRI features. This approach works well for binary classification tasks but required careful tuning to avoid overfitting.

```
svm_model = SVC(kernel='linear')
svm_model.fit(X_train_flat, y_train.argmax(axis=1))
```

Listing 4. Training SVM model

3.2.4 Deep learning architectures. Several pre-trained models were fine-tuned on the MRI dataset, including VGG16, VGG19, EfficientNetB0, ResNet50, ResNet101, InceptionV3, and DenseNet121. The base models were loaded with ImageNet weights, and additional layers were added to adapt them for binary classification.

```
vgg16 = VGG16(weights='imagenet', include_top=False,
    input_shape=(224, 224, 3))
vgg16_model = create_model(vgg16, train_generator.
    num_classes)
```

Listing 5. Fine-tuning VGG16 model

Each model used the Adam optimizer with a learning rate of 1×10^{-4} and binary cross-entropy loss. Early stopping was implemented to prevent overfitting by monitoring the validation loss.

4 Preliminary design

Our system architecture is designed as a seamless, end-to-end pipeline for classifying AD from MRI scans, as illustrated in Figure 1. The

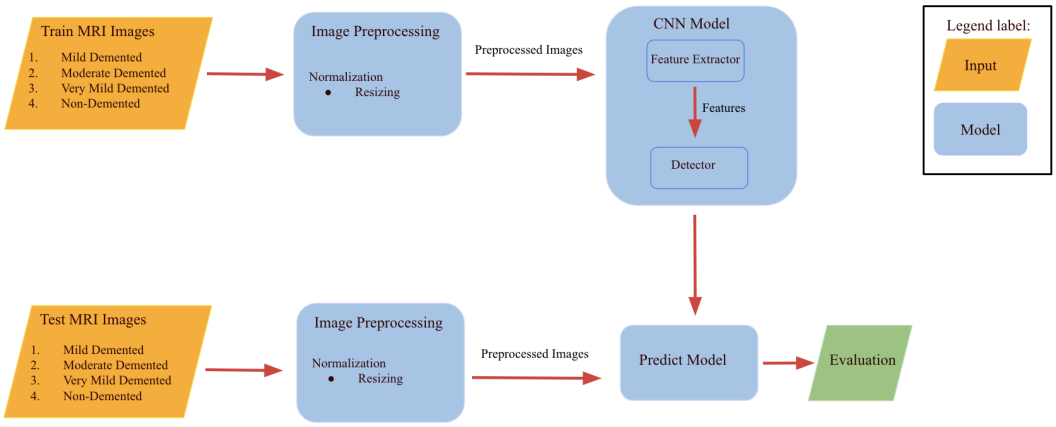

Fig. 1. The architecture of the convolutional neural network used in this study.

process begins with MRI scans being fed into the system, categorized into four distinct classes: mild demented, moderate demented, very mild demented, and non-demented. These images then undergo essential preprocessing steps, where normalization adjusts pixel intensity values, and resizing ensures uniform dimensions across all inputs.

The preprocessed images are subsequently fed into a CNN model, which acts as the core of the system. The CNN's feature extractor identifies and extracts critical patterns from the MRI scans, which are vital for distinguishing between the different stages of AD. The extracted features are then analyzed by the detection component, enabling the model to predict the appropriate category for each input.

This architecture not only handles the training phase but also supports testing, where the model's predictions are evaluated for accuracy and generalizability. The model's performance is rigorously assessed to ensure that it can reliably classify new, unseen data, making it a powerful tool for early diagnosis and treatment planning of AD.

Training and validation of the model utilize cross-validation techniques to fine-tune parameters and optimize generalization. The integration of this system into existing diagnostic workflows enhances current capabilities without requiring significant modifications, thereby providing a reliable and accurate disease classification system. Python, with the support of TensorFlow and Keras, serves as the foundation for developing and training this model, ensuring robust performance and adaptability in clinical settings.

5 Evaluation plan

The plan for evaluating the model designed to recognize AD includes a simple and flexible approach. We aim to check whether our model works well and can be trusted for practical use.

5.1 Metrics

In assessing the performance of our CNN model, the primary metric is its accuracy—specifically, its ability to correctly classify MRI scans in terms of whether they suggest the presence of AD. Accuracy is determined by comparing the model's predictions against known outcomes in the test set. This metric is crucial for evaluating the reliability of the model in clinical settings, where high precision is necessary to support medical decisions. Additionally, we may consider other performance metrics such as sensitivity and specificity, which provide deeper insight into the model's diagnostic capabilities, particularly its ability to detect true positive cases of AD and correctly identify negatives where the disease is not present.

5.2 Testing protocol

To ensure the robustness and generalizability of our CNN model, we employ a standard three-way split in our dataset: training, validation, and testing. The training set is used to fit the model parameters, while the validation set helps in tuning and selecting the best model parameters to prevent overfitting. The distinct testing phase involves evaluating the finalized model on a completely separate set of data that it has never encountered during the training or validation phases. This step is critical as it simulates real-world application, providing an unbiased evaluation of the model's effectiveness in new and varied clinical scenarios.

6 Risk analysis

With any ML project, numerous risks must be carefully managed to ensure the integrity and utility of the developed models. Among the technical challenges, one significant concern is the risk of overfitting—where the model learns the training data too well and fails to generalize to new, unseen data. To mitigate this risk, we employ techniques such as cross-validation and regularization during the model training phase.

Another technical concern involves biases that may arise from the data itself or the model's processing capabilities. Data bias can occur due to the way data is collected, processed, or selected for training. Since our dataset is sourced from publicly available data on Kaggle, it may not comprehensively represent all demographics or stages of AD, potentially leading to biased predictions against underrepresented groups. We aim to address this by further enriching our dataset with more diverse data sources in future iterations of the project.

Model bias is another area of concern, particularly with deep learning models like CNNs, which are inherently complex and often operate as "black boxes." There is a risk that the model might develop hidden biases against certain patterns or features in the data, which might not be immediately apparent. Efforts to increase the transparency and interpretability of the model, such as implementing model explanation tools or techniques like SHAP (SHapley Additive exPlanations) values or LIME (Local Interpretable Model-agnostic Explanations), are crucial to identify and mitigate these biases.

Operationally, the challenges lie in effectively deploying the model and ensuring it is adopted by end-users, which often requires integrating the model into existing clinical workflows without significant disruptions. To address these operational challenges, we will engage in thorough testing and conduct training sessions for healthcare professionals who will use the system.

7 Results

We evaluated multiple models, including traditional machine learning algorithms and deep learning architectures, to determine their effectiveness in classifying Alzheimer's disease from MRI images. The following table summarizes the performance of each model across key metrics: accuracy, precision, recall, and F1-score.

Table 2. Performance metrics for each model (sorted by accuracy)

Model	Accuracy	Precision	Recall	F1-Score
YOLO8	98%	70%	72%	75%
YOLO11 CIS	85%	68%	67%	67%
Random Forest	73%	73%	73%	72%
Logistic Regression	66%	68%	66%	65%
SVM	67%	69%	67%	67%
DenseNet121	52%	25%	50%	33%
ResNet50	50%	25%	50%	33%
VGG16	50%	25%	50%	33%
VGG19	50%	25%	50%	33%
EfficientNetB0	50%	25%	50%	33%
InceptionV3	50%	25%	50%	33%
ResNet101	49%	37%	50%	33%

7.1 Analysis of results

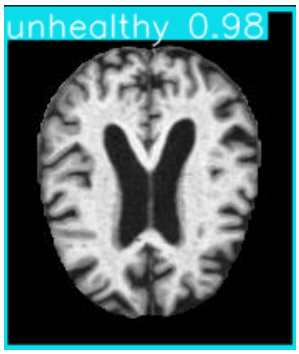

Fig. 2. Unhealthy classification (Confidence: 0.98).

Fig. 3. Healthy classification (Confidence: 0.96).

7.1.1 *Traditional machine learning models.*

- **Random Forest** achieved the highest accuracy among all models at 73%, with balanced precision and recall, indicating a good performance in classifying both healthy and unhealthy cases.
- **Support Vector Machine (SVM)** showed moderate performance with an accuracy of 67%. The model had higher precision for classifying unhealthy cases but struggled with healthy cases.
- **Logistic Regression** performed the lowest among traditional models with an accuracy of 66%, serving as a baseline for comparison.

7.1.2 *Deep learning models.*

- **YOLO8** demonstrated the highest accuracy among deep learning models at 98%, significantly outperforming YOLO11 CIS, which achieved 85%. YOLO8 also exhibited greater stability in classification tasks, with consistent precision and recall scores.
- YOLO8's performance is complemented by its efficiency, achieving speeds of approximately 2.2–2.5ms for preprocessing, 2.5–3.3ms for inference, and 2.9–3.9ms for non-maximum suppression (NMS) per image, making it suitable for real-time applications.
- **YOLO11 CIS**, while achieving a respectable accuracy of 85%, had slightly lower precision and recall scores compared to YOLO8, indicating less stable performance in distinguishing between healthy and unhealthy cases.
- All other deep learning models (VGG16, VGG19, EfficientNetB0, ResNet50, ResNet101, InceptionV3, DenseNet121) achieved an accuracy around 50%, comparable to random guessing in a binary classification task. This indicates challenges in generalizing to the dataset with these architectures.
- The low performance of other deep learning models may be attributed to overfitting, insufficient training epochs, or inappropriate learning rates.
- **DenseNet121** showed a slightly higher accuracy at 52%, but this was not significantly better than chance, highlighting the need for further optimization.

7.1.3 Discussion. The traditional machine learning models outperformed the deep learning architectures in this study. The Random Forest model, in particular, demonstrated better generalization on the test data. The deep learning models failed to learn meaningful patterns from the data, possibly due to:

- **Data quantity and quality**: Deep learning models require large amounts of data to train effectively. The dataset may not have been sufficient for these complex models.
- **Training parameters**: The number of epochs (5-10) might have been too low for the models to converge. Additionally, other hyperparameters like learning rate may need adjustment.
- **Class imbalance**: If there is an imbalance in the dataset between healthy and unhealthy classes, it can affect the training of deep learning models more significantly.
- **Model complexity**: The deep learning models may be too complex for the dataset, leading to underfitting or overfitting.

7.1.4 Recommendations. To improve the performance of deep learning models in future work, consider the following:

- Increase the size of the dataset through data augmentation techniques.
- Train the models for more epochs and use early stopping to prevent overfitting.
- Experiment with different learning rates and optimization algorithms.
- Use techniques like transfer learning with fine-tuning of pre-trained models.
- Address any class imbalance with resampling methods or adjusted loss functions.

8 Future work

Future work will focus on several key areas to enhance the model's applicability and accuracy:

(1) **Model interpretability**: To address the complexity and "black box" nature of deep learning models, we will explore methods like SHAP values or LIME to increase the transparency and interpretability of the CNN model. This will help clinicians understand the decision-making process of the model and increase trust in its predictions.
(2) **Operational integration**: Future efforts will also focus on integrating the model into existing clinical workflows. This includes collaborating with healthcare professionals to ensure the model's seamless adoption and conducting extensive testing in real-world settings to validate its performance.
(3) **Continuous learning and updating**: We aim to implement a system for continuous monitoring and updating of the model to ensure it remains accurate and relevant as new data becomes available and medical practices evolve.
(4) **Exploration of hybrid approaches**: Further exploration into hybrid models that combine CNNs with other ML techniques, such as ensemble methods, could potentially yield even higher accuracy rates and better diagnostic capabilities.

Acknowledgments

We thank the Earlham College CS department for their support.

References

[1] Maria Teresa Climent et al. 2018. Decision Tree for Early Detection of Cognitive Impairment by Community Pharmacists. *Frontiers in Pharmacology* 9 (2018). https://doi.org/10.3389/fphar.2018.01232
[2] Shaker El-Sappagh et al. 2021. A Multilayer Multimodal Detection and Prediction Model Based on Explainable Artificial Intelligence for Alzheimer's Disease. *Scientific Reports* 11, 1 (2021), 2660. https://doi.org/10.1038/s41598-021-82098-3
[3] Yasmina M Elgammal et al. 2022. A New Strategy for the Early Detection of Alzheimer Disease Stages Using Multifractal Geometry Analysis Based on K-Nearest Neighbor Algorithm. *Scientific Reports* 12, 1 (2022). https://doi.org/10.1038/s41598-022-26958-6
[4] Lawrence V Fulton et al. 2019. Classification of Alzheimer's Disease with and without Imagery using Gradient Boosted Machines and ResNet-50. *Brain Sciences* 9, 9 (2019), 212. https://doi.org/10.3390/brainsci9090212
[5] Ashir Javeed et al. 2023. Early Prediction of Dementia Using Feature Extraction Battery (FEB) and Optimized Support Vector Machine (SVM) for Classification. *Biomedicines* 11, 2 (2023), 439. https://doi.org/10.3390/biomedicines11020439
[6] Anup Juganavar et al. 2023. Navigating Early Alzheimer's Diagnosis: A Comprehensive Review of Diagnostic Innovations. *Cureus* 15, 9 (2023). https://doi.org/10.7759/cureus.44937
[7] Khandaker Mohammad Mohi Uddin et al. 2023. A Novel Approach Utilizing Machine Learning for the Early Diagnosis of Alzheimer's Disease. *Biomedical Materials & Devices* (2023), 1–17. https://doi.org/10.1007/s44174-023-00078-9
[8] Juan Yang et al. 2022. Random-Forest-Algorithm-Based Applications of the Basic Characteristics and Serum and Imaging Biomarkers to Diagnose Mild Cognitive Impairment. *Current Alzheimer Research* 19, 1 (2022), 76–83. https://doi.org/10.2174/1567205019666220128120927