Detection of Alzheimer’s Disease using Convolutional Neural Networks

Submitted By

Vishal yadav
E23CSEU0729

Under Supervision of

Dr. Nitin Arvind Shelke

School of Computer Science & Technology

Abstract

Alzheimer's Disease is a progressive and irreversible brain disorder that gradually destroys memory and cognitive skills. With the advent of deep learning, medical diagnostics have significantly advanced. This project focuses on using Convolutional Neural Networks (CNNs), a deep learning approach, for early-stage Alzheimer’s detection through MRI image classification. The model was trained on a publicly available dataset and integrated into a user-friendly web interface that allows real-time predictions. The model achieved exceptional accuracy, highlighting the practical application of AI in healthcare.

1. Introduction

Convolutional Neural Networks have revolutionized image analysis, enabling machines to learn spatial hierarchies in images without manual feature extraction. In the healthcare domain, CNNs are instrumental in detecting diseases from X-rays, MRIs, and CT scans. Alzheimer’s is one such disease where early diagnosis is crucial for patient care. This project presents a CNN-based classification model to categorize brain MRIs into four Alzheimer’s stages: Mild, Moderate, Non-Demented, and Very Mild. A web application was also developed to demonstrate the model’s functionality interactively.

---

2. Background

2.1 Problem Motivation

Alzheimer’s detection traditionally relies on neuropsychological assessments and manual MRI interpretation, which are time-consuming and subjective. Automation using deep learning addresses these challenges by offering speed, consistency, and precision.

2.2 CNN Overview

A CNN architecture consists of convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification. Key elements include activation functions, optimizers, and loss functions that collectively train the model.

2.3 Libraries Used

- **TensorFlow & Keras**: For building and training the CNN model
- **NumPy & PIL**: For image preprocessing
- **Matplotlib**: For visualizing accuracy and loss
- **Flask**: For web application deployment

---

3. Deep Learning Concepts

3.1 Artificial Neural Networks (ANN)

ANNs are the foundation of deep learning. They consist of interconnected neurons in layers that transform input data into meaningful outputs through weighted connections and activation functions.

3.2 Convolutional Neural Networks (CNN)

CNNs are specialized for processing image data. They use filters (kernels) that slide over images to detect patterns such as edges and textures. Pooling layers then reduce the dimensionality, allowing for faster and more robust training.

3.3 Activation Functions

- **ReLU (Rectified Linear Unit)**: Introduces non-linearity and speeds up training

- **Softmax**: Converts raw scores into class probabilities

3.4 Optimization & Loss

- **Adam Optimizer**: Combines the benefits of AdaGrad and RMSProp
- **Categorical Crossentropy**: Measures the performance of a multi-class classification model

---

4. Methodology

4.1 Problem Statement

Develop a multi-class classification model using CNN that can classify MRI scans into Alzheimer’s stages with high accuracy and deploy it via a web interface.

4.2 Dataset Description

- **Source**: Kaggle
- **Number of Classes**: 4 (Mild, Moderate, Non-Demented, Very Mild)
- **Image Format**: JPEG
- **Preprocessing**: Images resized to 224x224, normalized between 0 and 1
- **Split**: 80% training, 20% testing

4.3 Model Architecture

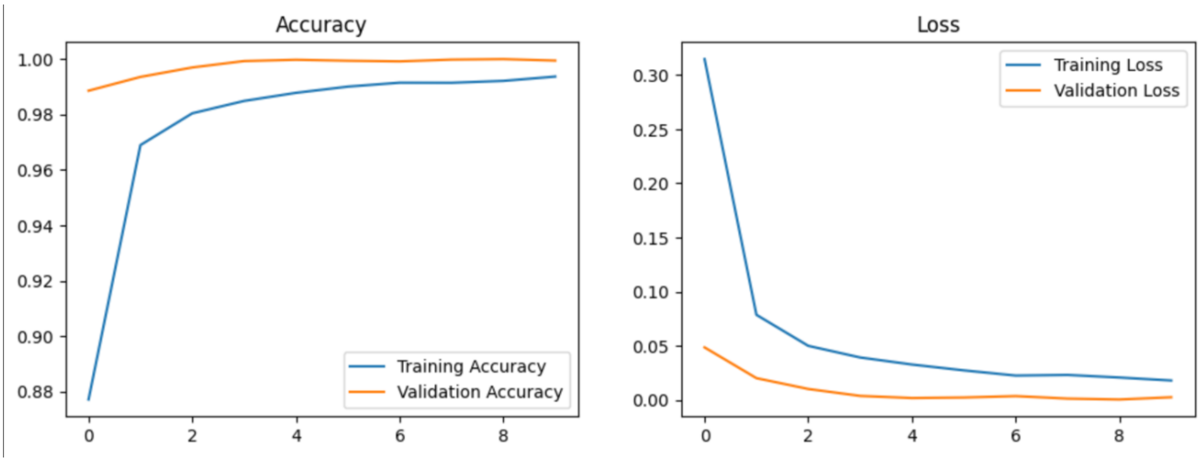
Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_4 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_5 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_1 (Flatten)	(None, 86528)	0
dense_2 (Dense)	(None, 128)	11,075,712
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 4)	516

```
Total params: 11,169,476 (42.61 MB)
Trainable params: 11,169,476 (42.61 MB)
Non-trainable params: 0 (0.00 B)
```

- **Layer 1**: Conv2D (32 filters) + ReLU + MaxPooling
- **Layer 2**: Conv2D (64 filters) + ReLU + MaxPooling
- **Layer 3**: Conv2D (128 filters) + ReLU + MaxPooling
- **Flatten Layer**: Converts output to 1D
- **Dense Layer**: Fully connected with 128 neurons
- **Dropout Layer**: Dropout(0.5) to prevent overfitting
- **Output Layer**: Dense(4) + Softmax

4.4 Training Configuration

- **Epochs**: 10
- **Batch Size**: 32
- **Loss Function**: Categorical Crossentropy
- **Optimizer**: Adam (learning rate = 0.001)



---

5. Results and Discussion

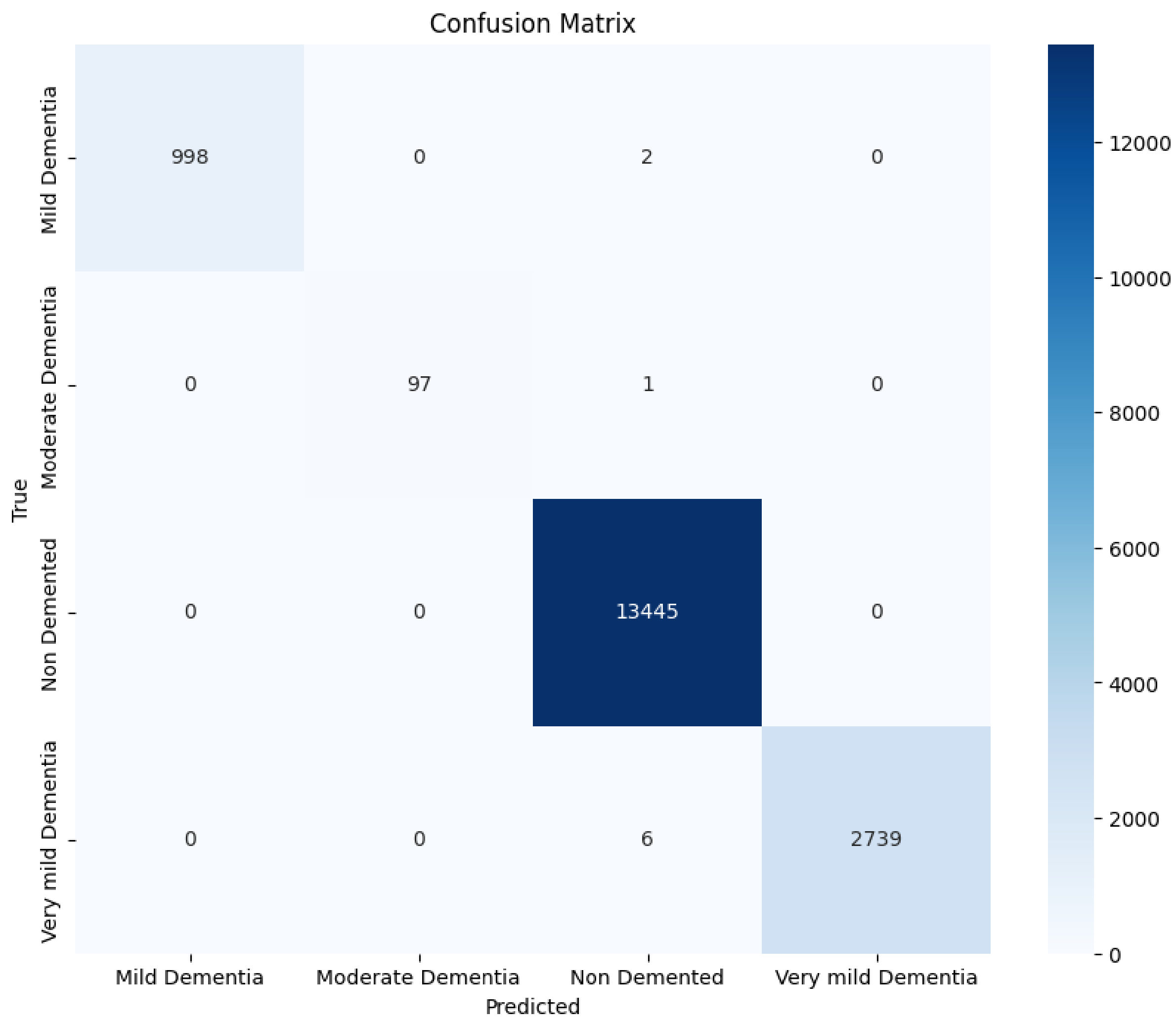
The trained CNN model achieved:

- **Test Accuracy**: 99.95%
- **Test Loss**: 0.0024

Classification Report:

- Mild: Precision 1.00, Recall 1.00
- Moderate: Precision 1.00, Recall 0.99
- Non-Demented: Precision 1.00, Recall 1.00
- Very Mild: Precision 1.00, Recall 1.00

Confusion Matrix confirms accurate predictions across all classes with minimal misclassification. The model's performance demonstrates the potential of deep learning in automating medical diagnosis.



6. Web Application (Frontend + Backend)

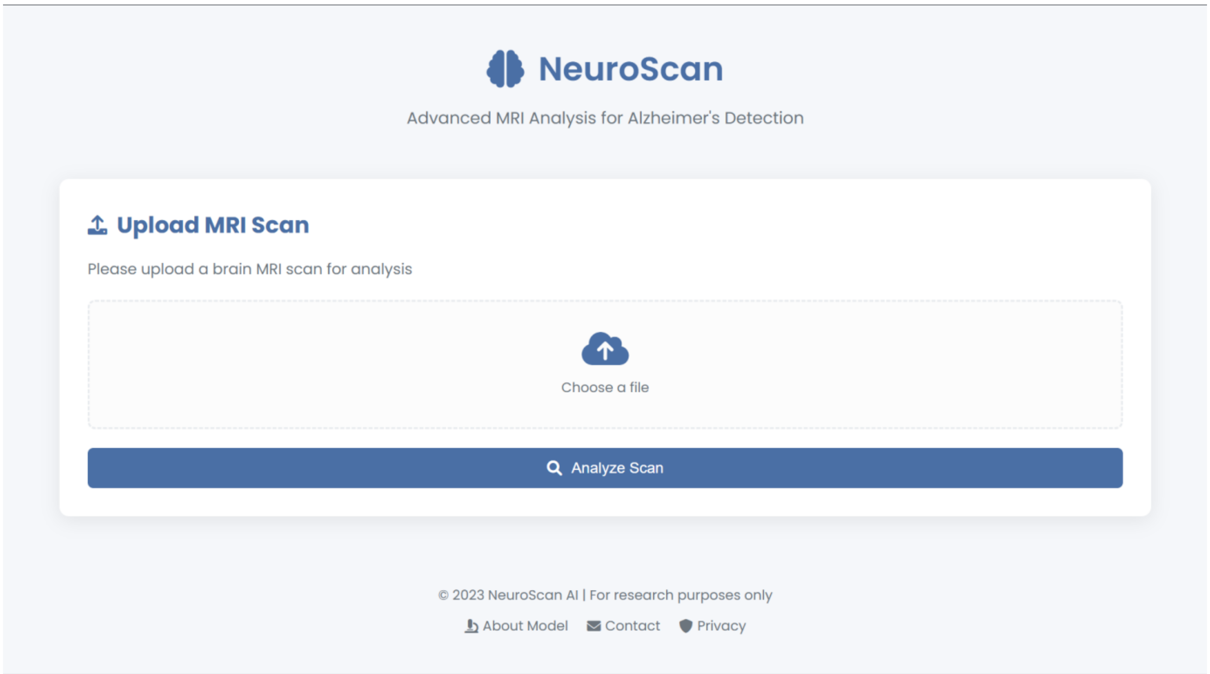
The model was integrated into a web application using Flask. Users can upload MRI images and receive a diagnosis instantly.

6.1 Frontend

- **HTML/CSS**: User interface for file upload
- **JavaScript**: For interactivity and displaying filenames

6.2 Backend

- **Flask App** (app.py): Handles file upload, image preprocessing, prediction, and returns results
- **Model Integration**: Uses model.predict() to generate predictions
- **Outputs**: Displays predicted class and confidence score on the frontend



---

7. Conclusion and Future Scope

This project successfully applies CNN to Alzheimer's MRI classification and shows excellent accuracy and usability. Future enhancements include:

- Using larger datasets
- Applying Transfer Learning (e.g., VGG16, ResNet)
- Mobile App deployment for clinicians
- Real-time scanning integration with hospital systems

---

8. References

1. Kaggle Dataset: https://www.kaggle.com/datasets/ninadaithal/imagesoasis
2. TensorFlow Documentation: https://www.tensorflow.org
3. Keras API: https://keras.io
4. Flask Documentation: https://flask.palletsprojects.com
5. Online Research Papers on CNN in Medical Imaging
