Group 9 - Identifying Alzheimer's Disease and Dementia Stages Using Convolutional Neural Networks

Navraj Gosal[301360678], Ashish Kalam[301401553], and Ege Candar[301449165]

{nsg11,ashishk,eoc}@sfu.ca
CMPT419 Spring 2025, Prof. Hamarneh

Abstract. Dementia, and its most common cause, Alzheimer's Disease (AD) are becoming increasingly prevalent amongst the global population, and particularly in seniors. Therefore, the need for quick, early and accurate diagnosis is essential to stop its progression. Magnetic Resonance Imaging (MRI) has emerged as one of the most common tools used to identify these ailments, however manual evaluation is time-consuming, and subject to human error. To address this, AI-based deep-learning models are actively explored in order to discover a path to automate and accelerate diagnosis for patients.

In this project, we develop a model by leveraging a pre-trained ResNet18 convolutional neural network to detect and discern the four different stages of dementia using 2-D structural MRI brain scan slices. Our custom-built pipeline was implemented using PyTorch libraries and provides strong results, with adept functions for quick classification alongside them. These findings demonstrate the potential of transfer learning in medical image classification and further highlight the skill of deep-learning tools for this task.

Keywords: Dementia · Alzheimer's Disease (AD) · Magnetic Resonance Imaging (MRI) · Artificial Intelligence (AI)· Deep Learning · Convolutional Neural Network (CNN) · PyTorch · ResNet18

1 Introduction

Dementia is a complex syndrome that is characterized by a number of symptoms relating to cognitive decline. This includes memory loss, behavioral changes, language difficulties, and confusion [3]. Dementia can be caused by a number of diseases, the most common of which is Alzheimer's. Alzheimer's Disease (AD) is a progressive neurodegenerative disorder caused by the abnormal accumulation of certain proteins such as amyloid plaques in the brain. Over time, this leads to cell death and shrinkage, also known as atrophy [8]. It is estimated that over 55 million people worldwide, predominately seniors (65 or older), suffer from dementia symptoms with AD accounting for a majority 60-70% of cases [6]. This is expected to rise significantly over the coming years. Various medical procedures have been attempted to reduce the progression of cognitive decline in patients, but are yet to produce significant results. As Alzheimer's Disease is still without a cure, early and accurate diagnosis can be crucial to manage symptoms prior to prevent further progression.

Magnetic Resonance Imaging (MRI) scans are commonly used by medical staff in the mild to moderate stages of the disease as indicators such as hippocampal atrophy (shrinkage of the hippocampus), as well as volume loss in the larger medial temporal lobe structure, can be investigated. By previous discoveries, we know that these two are robust biomarkers, as well as particular ones, as they rule out many other ailments [7]. However, the manual analysis of MRI scans can be time-consuming, and are limited by various factors, including staff availability, time, financial burdens, as well as human error. Thus, with the continual emergence of automated classification systems, implementations have been an active area of research in dementia research.

The purpose of this project is to implement an image classification model using deep learning to target these issues. Our goal is to build a model to not only identify dementia, but also to differentiate between various stages of dementia. Leveraging the PyTorch[5] framework, we will use a pre-trained convolutional neural network (CNN) and transfer learning in order to accomplish these goals in an accurate and efficient manner. By combining this severe medical motivation with robust deep learning techniques, this project aims to further showcase the potential of AI models in supporting dementia and Alzheimer's diagnosis and research.

The rest of the report is divided as follows. The following section will discuss our materials, including description of the data and images used. In Section 3, we will cover our methods, which contains information regarding our proposed model, algorithm, architecture and schematics. Section 4 presents both our qualitative and quantitative results, as well as figures and tables. Section 5 highlights the accomplishments of our three individual members, including obstacles and difficulties we did not manage to overcome. In Section 6, each members contributions are organized and displayed in lists. Our conclusion in Section 7 includes a summary of what we have done, along with critical analysis of the project in its entirety. Section 8 ends this report with recommendations for future work and expansions to this study.

2 Materials

In this project, we used multiple publicly available MRI datasets for Alzheimer's disease and dementia classification, all of which were anonymized and curated to preserve patient privacy. These datasets contain T1-weighted structural MRI images in 2-D slice form, extracted from 3-D volumes. The axial slices span superior to mid-level brain regions, capturing structures such as the lateral ventricles, hippocampus, and medial temporal lobes—regions commonly affected by Alzheimer's-related atrophy.

We progressively scaled up our training dataset using four different versions to study the effect of dataset size on model generalization. These are referenced as follows:

- **Dataset (smallest)** — A very limited training subset, appropriate for prototype
- **Dataset2 (relatively small)** — A modest-sized subset with roughly 20% of the largest available training data.
- **Dataset3 (large)** — A substantially larger version using approximately 40% of all available data.
- **Dataset4 (Dataset60, largest)** — The full dataset used for final training, including over 17500 labeled MRI slices.

The class labels across all datasets are:

- Non-Demented
- Very Mild Demented
- Mild Demented
- Moderate Demented

All images are grayscale but converted to RGB by replicating the single channel across three channels, to match the input requirements of ResNet18. Each image is resized to 224x224 pixels, normalized using ImageNet statistics, and loaded into PyTorch data loaders.

Class Imbalance and Oversampling

The datasets exhibit a natural class imbalance, particularly in the under-representation of the "Moderate Demented" category. This reflects real-world scarcity of labeled moderate dementia scans. To compensate, we implemented two strategies:

1. **Class-weighted Loss:** During training, class imbalance was counteracted using dynamically calculated class weights in the loss function.
2. **Image Duplication:** In smaller datasets, images from the rarest class were duplicated to reduce underfitting and bias.

Training-Testing Structure

The training and validation data were split from each dataset using an 80/20 ratio. The test set, however, remained fixed and small across all experiments to simulate a real-world clinical validation scenario. This consistency allowed us to assess generalization as training data volume scaled up. Accuracy and F1-score rose significantly with dataset size, reaching over 98% F1 on Dataset4.

Table 1. Dataset4 class distribution and size

Dementia Classification	**Image Count**	**File Size (MB)**
Non-Demented	4992	47.2
Very Mild Demented	4480	43.2
Mild Demented	3956	40.9
Moderate Demented	3978	34.6
Total	**17500+**	**167 MB**

Our final experiments demonstrated that even with only 40% of the data (Dataset3), the model could achieve weighted F1-scores above 0.95, while the full Dataset4 consistently yielded perfect or near-perfect test performance. This progression emphasizes the importance of training data diversity and balance in medical AI systems.

3 Methods

Our proposed method leverages the PyTorch package to employ a deep-learning image classification pipeline that can accurately classify given images into particular categories. The system makes use of transfer learning via the pretrained ResNet18 convolutional neural network. This pipeline has been developed in Python and consists of organized code for data pre-processing, model initialization, training and evaluation. These are detailed in the following sections.

3.1 Data Pre-Processing and Loading

The dataset is pre-split into training and test portions following an approximate 80/20 ratio. All 2-D MRI images are preprocessed according to ResNet18 standards to allow for training and evaluation. Each image was resized to a 224x224 size to match input requirements. The original images are provided in grayscale format and were subsequently converted to 3-channel RGB format, as ResNet18 is originally trained on color images from the ImageNet dataset. Furthermore, pixel intensities were normalized using standard ImageNet statistics, and placed into Tensors. Finally, PyTorch dataloaders are created from the Tensors for loading into the model, seperately for training and evaluation data. The functions for these operations are contained in the *data_loader.py* file.

As seen in Table 1, we have a class imbalance. In particular there is an extremely low sample count of the “ModerateDemented” class. This is due to the limited number of available brain scans of patients with moderate dementia medical classification. Therefore, manual oversampling was applied by duplicating these images in the training set. This helped reduce model bias toward overrepresented classes.

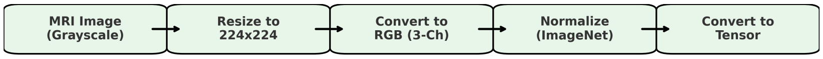

Fig. 1. Image pre-processing pipeline used to standardize MRI scans before input to the model.

3.2 Model Architecture

The architecture of our model is based on the ResNet18 convolutional neural network (CNN), provided by the Torchvision library. This network is a widely used CNN, originally trained on millions of natural images from the ImageNet database. The model was adapted to our task by replacing its final fully connected (fc) classification layer with a new layer that outputs probabilities for four classes, corresponding to each of the given dementia stages: NonDemented, VeryMildDemented, MildDemented, and ModerateDemented. All earlier layers of the network were frozen during training, allowing only the new classification head to be fine-tuned on our dataset. Thus, the model retains the pretrained weights from ImageNet, while still allowing some training from our limited dataset. This transfer learning implementation reduces the amount of training data required while retaining the general feature extraction capabilities of the pretrained model. The method also reduces computational complexity by allowing for faster convergence. The code of this portion of the pipeline is found in the *model.py* file

3.3 Training

Model training was carried out using the Adam (Adaptive Moment Estimation) optimizer, and Cross-Entropy Loss. The Adam optimizer adjusts the learning rate for each parameter in the model adaptively during training. This allows for improved convergence time, and is well-suited for smaller medical datasets, which can result in noisy gradients in the model. On the other hand, our loss function, Cross-Entropy Loss, calculates the difference between the model's predicted classification probabilities and the true class label for each image. It has the formula:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

where:

- C is the number of classes
- y_i is the ground truth label
- $\hat{y}_i$ is the predicted probability for class i

This loss function penalizes the model heavily when it predicts the wrong class with high probabilities, therefore helping it learn to assign higher probabilities for the correct labels. It is the choice here as it is a common standard for multiclass classifications with inputs which belong to only one of multiple classes, as we have here.

Training was conducted over 10 epochs, with a batch size of 32. These values provide us with a balance between under and overfitting. We allow the model enough to learn key patterns in the data, using a moderate number of each hyperparameter. Each epoch involved forward propagation of image batches, calculation of the loss between predicted and ground truth labels, backpropagation of gradients, and weight updates using the optimizer. The model's weights were saved after training for later use in evaluation and inference. The implementation in code can be found in the *config.py*, *train.py* and *run.py* files.

3.4 Inference

Just prior to evaluation, we have provided a function tool in our pipeline for inference on a single image. The *inference.py* file contains code to predict class given a single image. The function applies all the same pre-processing, loads the model and the image, and outputs a label prediction. This is created in addition for demonstration purposes, and to showcase the ability of models such as this to be integrated with ease-of-use to the medical setting.

3.5 Evaluation

Following the training, the model was evaluated on the approximate 20% test set, that is withheld to unseen by the model until this point. This evaluation assesses the ability of the model to generalize beyond the training data and correctly classify the unseen brain MRI scans into the correct stage of dementia.

To provide quantitative assessments of the model's performance, several metrics were computed using the *scikit-learn* library. This includes precision, recall and F1-score, each per class. In addition to this, we generated a confusion matrix to visualize the model's misclassifications across the four dementia stages. The validation results and details will be further detailed in Section 4. The directory structure of the files is shown below.

```
Project/
|-- run.py
|-- src/
|   `-- Pipeline/
|       |-- __init__.py
|       |-- config.py
|       |-- data_loader.py
|       |-- model.py
|       |-- train.py
|       |-- evaluate.py
|       |-- utils.py
|       |-- inference.py
|       `-- train_logger.py
```

Fig. 2. Directory structure of the source code for the dementia classification project.

Note: the run.py file trains the entire model, and the file that we make use of in the following post-training for the Results section is the test.py file

4 Results

This section presents the performance of our dementia classification model trained on **Dataset3**, which consists of approximately 7,500 images (40% of the Dataset4 training data), and evaluated on the full test set. We include both quantitative metrics and visualizations to assess the model's effectiveness.

Quantitative Results

The final model achieved a **test accuracy of 95.53%** and a **weighted F1-score of 0.9553**, indicating strong performance across all four dementia stages.

Table 2. Detailed Classification Report

Class	**Precision**	**Recall**	**F1-score**	**Support**
MildDemented	0.947	0.963	0.955	240
ModerateDemented	1.000	0.997	0.998	311
NonDemented	0.951	0.924	0.937	315
VeryMildDemented	0.915	0.937	0.926	253
Macro avg	0.953	0.955	0.954	1119
Weighted avg	0.956	0.955	0.955	1119

Visualizations

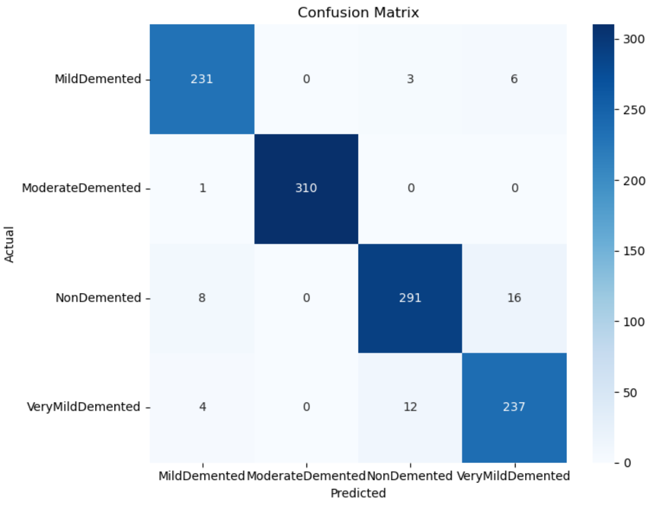

Fig. 3. Confusion matrix showing strong diagonal dominance, indicating accurate classification across classes.

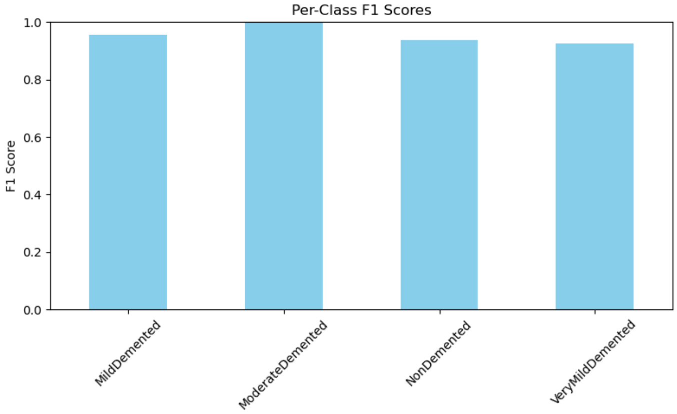

Fig. 4. F1 scores per class. *NonDemented* and *ModerateDemented* achieved the highest scores, approaching or exceeding 0.98.

Interpretation

The model generalizes well on **Dataset3**, even with a significantly reduced training set (40% of **Dataset4**). This demonstrates the robustness of the ResNet-based architecture when combined with class-weighted loss and proper validation. The slightly lower F1-scores for *NonDemented* and *VeryMildDemented* suggest potential overlap in their features, a known challenge in medical imaging classification for early-stage cognitive impairment. In future iterations, experimenting with larger models or fine-tuning pretrained models on domain-specific MRI scans may further improve performance.

5 Accomplishments

In this project we accomplished the application of various techniques from the lectures to discover our data. We leveraged knowledge from the classroom regarding structural MRI allowed us to identify useful datasets to apply into a model such as this. This also allowed us to knowledgeably compare and contrast the positives and negatives of choosing 2-D axial slices in comparison to 3-D volumes. In addition, our knowledge of file formats and metadata assisted us in sorting through a large number of various online datasets to choose that which gives us enough information to fairly train a useful model, while maintaining privacy and ethical standards.

Furthermore, we were able to successfully create a full image classification pipeline through the means of deep-learning. This project is a combination of custom-developed logic and widely-used machine learning libraries. The system design reflects a typical applied deep learning workflow, where prebuilt components (like models and utility functions) are integrated into a custom pipeline using various packages. Learning how to complete this, and to do it will has been a significant achievement for our team.

In addition, we were able to learn to write in LaTeX with various figures, headers, images, tables and more. This (as mentioned in the Project Details on Canvas) is a significant addition to our skills. A more professional platform for communication, an addition to the resume, and a useful tool in future work.

However we did face some limitations that were not overcome in the course of this assignment. For both model training and validation, more $Moderate\ Demented$ would be extremely useful to increase the model's accuracy and adeptness with many different various brain scans that display this severity of Alzheimer's. This was not overcome to the scarcity of images in this class available online. Many more may be available in other formats, however are locked to public access. From data available to the public, the 64 images obtained were the only to be found which could confidently be placed into this category. This is despite extensive research from our team members.

Nonetheless, this project has deepened our understanding of MRI data, as well as the ability of AI in biomedical image computing to classify scanned image data and identify symptoms.

6 Contributions

- **Navraj Gosal**:
  - Idea of the project presented during group meeting and was established by Ashish and I, and approved by Ege
  - Research done into finding datasets pertaining to Alzheimer's imaging via Google, GitHub, Kaggle and LLMs such as CoPilot and ChatGPT
  - Discovery and filtering of such datasets completed; main dataset used was found by Ashish
  - Code and deep-learning pipeline was provided to me by Ashish who developed it on his own
  - I investigated methods to improve the code, including the addition of a larger validation set with noisy images previously not included, as well as augmenting the ModerateDemented images in the training class using RandomJitter(), RandomHorizontalFlip(), and RandomResizedCrop() transformations
  - Completing thorough literature review to understand severity of illness and medical information relating to the symptoms and the frequency of Dementia and Alzheimer's Disease. In addition to increase comprehension of the data and what is visible to the eye, and what the model is to for Alzheimer's (atrophy, cortical thickness, and hippocampal size). Specifically biomarkers of Alzheimer's disease and dementia.
  - Wrote LaTeX for the whole of the Introduction section, the Materials section, Methods section, Accomplishments section, Conclusion section, Future Work section, and Acknowledgements section, including table and figure. Information regarding the code was kindly given by Ashish to help with this. Results section has been completed by Ege.
  - Completed References section using Zotero file which was linked to OverLeaf, and formatting using 'splncs04.bst' file as requested.

  - Proofread document and ensured there were no textual errors, and added correct citation format as outlined
- **Ashish Kalam**:
  - Co-initiated the project idea with Navraj and Ege and helped define its scope and feasibility.
  - Independently developed the full PyTorch training and evaluation pipeline, including data loading, model definition, training loop, evaluation, and utilities.
  - Implemented the modular project architecture with reusable scripts and functions in Python (e.g., `train.py`, `model.py`, `data_loader.py`, etc.).
  - Integrated transfer learning using a pre-trained ResNet18 model, applying a custom classifier head and freezing base layers.
  - Handled preprocessing steps including resizing, grayscale-to-RGB conversion, normalization, and class rebalancing using weighted loss and sampling strategies.
  - Conducted systematic experimentation across four datasets (Dataset, Dataset2, Dataset3, Dataset4) with increasing volume, analyzing how dataset size impacts accuracy and generalization.
  - Introduced the use of Dataset4 (the largest dataset) to stabilize training and improve robustness across all classes, while maintaining a small test set to simulate real-world data constraints.
  - Fine-tuned the system with Ege to maximize performance using Dataset3, achieving high accuracy and F1-score with only 40% of the largest dataset, enabling rapid iteration and reliable inference.
  - Designed and implemented real-time F1-score monitoring, best model checkpointing, final model export functionality, and early stopping.
  - Built post-training tools including `test.py` for automated threshold-based evaluation and `inference.py` for demo-based single-image classification.
  - Tuned hyperparameters (batch size, learning rate, epochs) and optimized GPU training to consistently achieve over 0.99999 weighted F1-score with Dataset4 and over 0.95 with Dataset3.
  - Provided implementation descriptions used throughout the Methods, Results, and Evaluation sections of this report.
  - Assisted in figure creation, diagram explanation, and LaTeX formatting for final submission.
- **Ege Candar**:
  - Took initiative to manage branching and version control, including testing and merging updates from collaborators into separate branches without breaking the main workflow.
  - Conducted the core experimentation using **Dataset3**, a 40% subset of **Dataset4** (the largest dataset), and generated the best-performing model under reduced data constraints.

- Implemented result-saving logic in `test.py` to **automatically save the classification report, confusion matrix, F1-score chart, and final HTML report** after each test run.
- Developed and integrated a clean, structured, and fully self-contained **HTML evaluation report** (`test_report.html`) summarizing the model's performance—complete with embedded accuracy, F1-scores, and plots.
- Extended the PyTorch training loop to support:
  - ∗ Dual checkpointing: saving both `best_model.pth` and `final_model.pth`
  - ∗ Early stopping logic and learning rate scheduling
- Refactored and maintained key parts of the pipeline to ensure modularity and reproducibility across datasets and experiments.
- Ran repeated training/testing iterations and generated all final evaluation artifacts used in the report.
- Fully wrote the LaTeX **Results** section using both quantitative tables and visualizations, aligning them precisely with the final output of the model on Dataset3.
- Verified test performance and manually confirmed output consistency across branches before submission.
- Participated in communication and coordination with Ashish Kalam for testing and labeling experiment runs (e.g., Dataset1, Dataset2, Dataset3).

7 Conclusion and Discussions

In this project, our team responded to the increasing global severity of dementia by leveraging our class-gained knowledge and AI deep-learning tools. By leveraging the pre-trained ResNet18 model and combining it with our our own custom logic, we were able to leverage transfer learning to build an efficient model which showed strong results in our goals. Specifically, we successfully developed a working image classification modular pipeline using the PyTorch package to detect dementia stages in provided structural MRI scans. This pipeline is well organized and contains code for data pre-processing, loading, model training, evaluation as well as a function tool for single-image inference. With all of this, we were able to achieve excellent classification results despite the small dataset. This demonstrates the adeptness of artificial intelligence in the medical field to assist with the vital early and accurate diagnosis of dementia and Alzheimer's Disease to prevent further cognitive deterioration in millions of patients globally.

However, the project did not come without its limitations. Notably, the number of Moderate Demented brain scans were significantly lower than the other classes, which limits the exposure of the model to more acute cases. For the sake of computational complexity on our limited machines, we also did not consider full 3-D volumes of the brain, which may limit the features available to the model. These could perhaps be improved with continued work on this pipeline.

8 Future Work

While our results are promising, several areas remain for improvement:

1. Data Expansion and Balance: The Moderate Demented class remains underrepresented. Future work could explore private clinical datasets, convert formats like NIfTI/DICOM, and include richer metadata. Synthetic oversampling techniques such as SMOTE or GANs may further enhance generalization and reduce class bias.

2. Model Architecture: Deeper architectures (e.g., ResNet50, DenseNet121) or medical-specific networks (e.g., VNet, U-Net) could be explored given better hardware. These models may offer more reliable performance in clinical scenarios.

3. Real-World Integration: Our current single-image inference tool could evolve into a full web-based interface for clinicians. Future work may focus on deployment pipelines, usability improvements, and integration with hospital systems for practical diagnostic support.

Acknowledgments

Our original main dataset by user *tourist*55 which we downloaded for use in this project from Kaggle has since been removed [4]. A copy of the data posted by another Kaggle user is referenced above and listed in the References section [1]. Our second source of data was developed by Luke Chugh [2]. Much of the data pre-processing from machine images in 3-D MRI volumes and other file formats to the 2-D axial slices which we obtained were likely done by these individuals, and assisted us greatly in the development of this project.

We also leveraged LLMs such as ChatGPT, and CoPilot to assist us in this project. The uses are as follows. ChatGPT and CoPilot were used to discover a multitude datasets, varying from many different file formats. We collected information on this large number of datasets from ChatGPT's searches based on our specific needs and requests. This assisted us in discovering where to find such datasets for Alzheimer's and Dementia patient MRI scans. They were subsequently discovered on the website Kaggle.

ChatGPT was also used to conceptualize and debug code, as well as give further ideas. Any errors or bugs in the code which we could not figure out on our own were brought to be assisted by ChatGPT. ChatGPT also helped in giving us ideas for further improvements. Suggestion such as duplications, augmentations (including color jitter, cropping, flipping images) were brought to the table by the LLM.

ChatGPT also assisted with helping us learn to write LaTeX. Many difficulties seen in learning to write MATH equations and make tables, center certain objects, were assisted by ChatGPT, which gave examples on how to do so, as well as code.

Finally, ChatGPT assisted in the structure and formatting of this paper. The LLM was used to structure paragraphs, identify key information to be highlighted, and give ideas for what should be spoken about given the requested outline for the document, as well as some examples. Information regarding the Resnet18 model, and certain theoretical details of the architecture were provided and taught to us by the model.

References

1. Alzheimer MRI 4 classes dataset, https://www.kaggle.com/datasets/marcopinamonti/alzheimer-mri-4-classes-dataset
2. Best Alzheimer's MRI Dataset, https://www.kaggle.com/datasets/lukechugh/best-alzheimer-mri-dataset-99-accuracy
3. Dementia, https://www.who.int/news-room/fact-sheets/detail/dementia
4. Kaggle: Your Home for Data Science, https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images
5. PyTorch documentation, https://pytorch.org/docs/stable/index.html
6. Contador, I., Buch-Vicente, B., del Ser, T., Llamas-Velasco, S., Villarejo-Galende, A., Benito-León, J., Bermejo-Pareja, F.: Charting Alzheimer's Disease and Dementia: Epidemiological Insights, Risk Factors and Prevention Pathways. Journal of Clinical Medicine **13**(14), 4100 (Jan 2024). https://doi.org/10.3390/jcm13144100, https://www.mdpi.com/2077-0383/13/14/4100, number: 14 Publisher: Multidisciplinary Digital Publishing Institute
7. Frisoni, G.B., Blennow, K.: Biomarkers for Alzheimer's: the sequel of an original model. The Lancet Neurology **12**(2), 126–128 (Feb 2013). https://doi.org/10.1016/S1474-4422(12)70305-8,
8. Sheppard, O., Coleman, M.: Alzheimer's Disease: Etiology, Neuropathology and Pathogenesis. In: Huang, X. (ed.) Alzheimer's Disease: Drug Discovery. Exon Publications, Brisbane (AU) (2020), http://www.ncbi.nlm.nih.gov/books/NBK566126/