Cognitive Imaging

Dementia Detection

Binary Classification CNN using 3D Convolutional Blocks

Contributors
Alexander Hajdukiewcz, Michael Batushansky, Amy Margolina

The Problem

The Challenge:

- Dementia affects **50 million people worldwide**, projected to triple by 2050 (WHO).
- Early diagnosis is crucial to slow progression and improve patient outcomes.

The Gap:

- Current methods are subjective, time-consuming, and inaccessible in many areas.
- Require expert interpretation

The Goal:

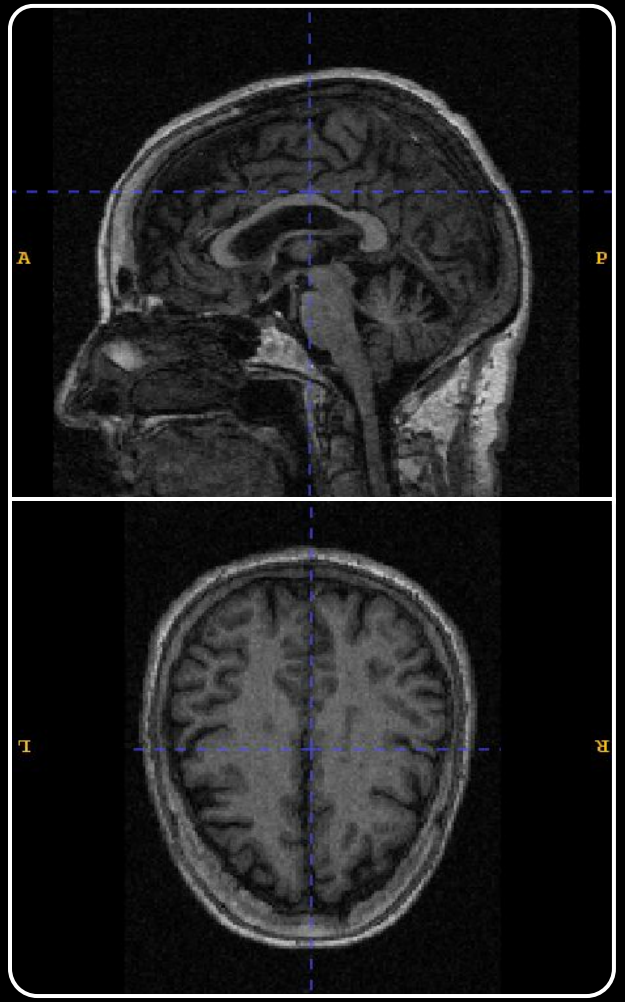
- Create a **machine learning model** to analyze brain images for early dementia detection:
  - Faster, more accurate, and scalable diagnostic aid.
  - Improved patient outcomes through earlier intervention.

Proposal

- identify and detect dementia and cognitive decline based on high resolution MRI images
- early detection of cognitive decline before Mini-Mental State Examination (MMSE)
  - "...did not find evidence supporting a substantial role of MMSE as a stand-alone single-administration test in the identification of MCI patients who could develop dementia." (Mitchel, 2009)
  - "In memory clinic settings the MMSE had ... a positive predictive value (PPV) of 86.3% and a negative predictive value (NPV) of 73.0%

Score	Degree of Impairment	Formal Psychometric Assessment	Day-to-Day Functioning
25-30	Questionably significant	If clinical signs of cognitive impairment are present, formal assessment of cognition may be valuable.	May have clinically significant but mild deficits. Likely to affect only most demanding activities of daily living.
20-25	Mild	Formal assessment may be helpful to better determine pattern and extent of deficits.	Significant effect. May require some supervision, support and assistance.
10-20	Moderate	Formal assessment may be helpful if there are specific clinical indications.	Clear impairment. May require 24-hour supervision.
0-10	Severe	Patient not likely to be testable.	Marked impairment. Likely to require 24-hour supervision and assistance with ADL.

https://pmc.ncbi.nlm.nih.gov/articles/PMC8406467/
https://pubmed.ncbi.nlm.nih.gov/18579155/



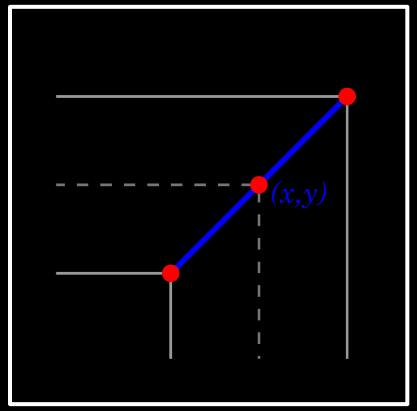
Input and Data Set

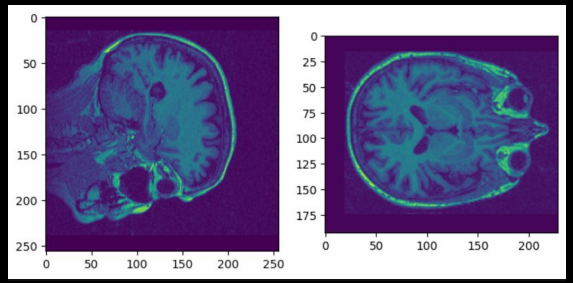
- Open Access Series of Imaging Studies (OASIS)
  - used OASIS-2 data set
  - 3 or 4 individual T1-weighted MRI scans (x,y,z axis views)
  - all MRIs marked Nondemented, Demented, Converted
- NIFTI-1 (Neuroimaging Informatics Technology Initiative)
  - dual file format (.hdr & .img)
  - dimensions: 256 x 256 x 128
- Benefits of OASIS image set for pre-processing:
  - uniform voxel (3D pixel) size: 1 x 1 x 1.25
  - co-registered images (not for the RAW directory we used)
- Limitations of OASIS-2
  - small image set pool

Subject ID	MRI ID	Group	Visit	MR Delay	M/F	Hand	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
OAS2_0001	OAS2_0001_MR1	Nondemented	1	0	M	R	87	14	2	27	0	1987	0.696	0.883
OAS2_0001	OAS2_0001_MR2	Nondemented	2	457	M	R	88	14	2	30	0	2004	0.681	0.876
OAS2_0002	OAS2_0002_MR1	Demented	1	0	M	R	75	12		23	0.5	1678	0.736	1.046
OAS2_0002	OAS2_0002_MR2	Demented	2	560	M	R	76	12		28	0.5	1738	0.713	1.010
OAS2_0002	OAS2_0002_MR3	Demented	3	1895	M	R	80	12		22	0.5	1698	0.701	1.034
OAS2_0004	OAS2_0004_MR1	Nondemented	1	0	F	R	88	18	3	28	0	1215	0.710	1.444
OAS2_0004	OAS2_0004_MR2	Nondemented	2	538	F	R	90	18	3	27	0	1200	0.718	1.462

https://sites.wustl.edu/oasisbrains/home/oasis-2/

Pre-Processing(method 1)





- Extracting z-axis view from NIFTI:
  - SimpleITK, Nibabel Library
    - extremely useful for working with NIFTI
    - we worked with greyscale, but arbitrary channels allowed

1. Images resampled (discrete) ; first image write

- interpolation to continuous grid representation
- easier and smoother transformations

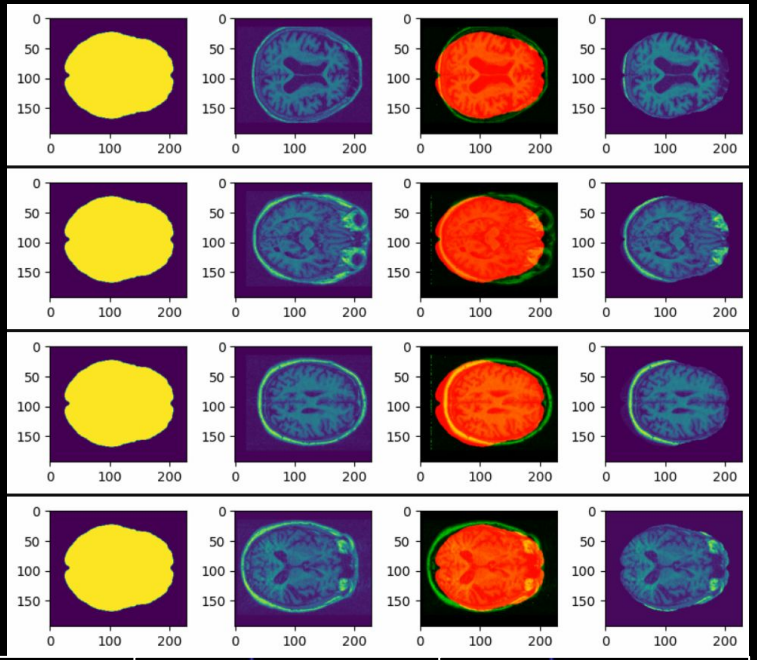
2. Registration alignment; second image write

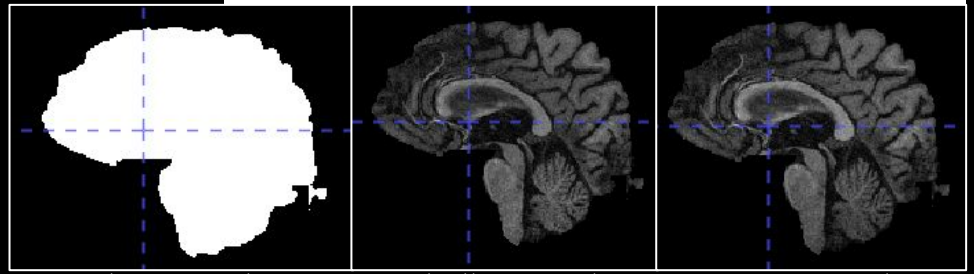
- ICBM 2009c Nonlinear Symmetric template
  - .nii.gz (single file format)
  - axis rotations were required to match input orientation
- Affine registration through Nibabel library

https://nist.mni.mcgill.ca/icbm-152-nonlinear-atlases-2009/
https://nipy.org/nibabel/
https://neuroimaging-data-science.org/content/006-image/003-registration.html
https://www.kaggle.com/code/rasoulisaeid/oasis1-skull-stripping
https://en.wikipedia.org/wiki/Linear_interpolation

Pre-Processing (cont.)

- Skull stripping:
  - isolates brain tissue features
  - second round of image-writing
  - initial issues
    - limitations with template mask
    - alternative: pre-trained or custom NPP
      - more flexibility and accuracy
- Pre-trained skull-stripping:
  - PARIETAL : trained on OASIS images
- Noise Removal:
  - N4 Bias Field correction
    - corrects low frequency intensity non-uniformity
- HDF5 conversion:
  - Image files converted to .h5py for compactness and better management tools
  - commonly used in medical imaging applications





brain mask | skull stripped | N4 corrected

https://www.kaggle.com/code/rasoulisaeid/oasis1-skull-stripping
https://github.com/rasoulisaeid/skull_stripping
https://github.com/sergivalverde/PARIETAL

Model Outline

```
CNNModel(
  (skip_connection_block): SkipConnectionConvBlock(
    (conv_blocks): ModuleList(
      (0): Conv3d(1, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
      (1): Conv3d(16, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
      (2): Conv3d(17, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
      (3): Conv3d(16, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
    )
    (batch_norms): ModuleList(
      (0-3): 4 x BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (maxpool): MaxPool3d(kernel_size=3, stride=3, padding=1, dilation=1, ceil_mode=False)
    (relu): ReLU()
  )
  (resnet_blocks): ModuleList(
    (0): ConvBlock(
      (blocks): Sequential(
        (0): Conv3d(16, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
        (1): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): MaxPool3d(kernel_size=3, stride=3, padding=1, dilation=1, ceil_mode=False)
        (3): ReLU()
      )
    )
    (1): ConvBlock(
      (blocks): Sequential(
        (0): Conv3d(16, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
        (1): BatchNorm3d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): MaxPool3d(kernel_size=3, stride=3, padding=1, dilation=1, ceil_mode=False)
        (3): ReLU()
      )
    )
    (2): ConvBlock(
      (blocks): Sequential(
        (0): Conv3d(32, 64, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))
        (1): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): MaxPool3d(kernel_size=3, stride=3, padding=1, dilation=1, ceil_mode=False)
        (3): ReLU()
      )
    )
  )
  (fc): Linear(in_features=64, out_features=1, bias=True)
  (sigmoid): Sigmoid()
)
```

- borrowing from ResNet-18 structure
  - smaller number of blocks to counteract small image set
- 3D Convolutional blocks
  - more features
  - appending the target image as-is (y-axis orient.)
- Image set (1368)
  - training set: 70%
    validation set: 15%
    test set: 15%
- Other features
  - Optimizer: Adam with momentum(0.1)
    - dynamic learning rate vs SGD
  - ReLU activation
  - Sigmoid (for binary classification)
  - initial learning rate: 1^{-3}
  - Loss function: BCE
    - since we are evaluating binary classification
  - batch normalization
  - max-pooling
    - helped reduce training time

https://www.researchgate.net/figure/Structure-of-the-Resnet-18-Model_fig1_366608244,
https://arxiv.org/pdf/1512.03385

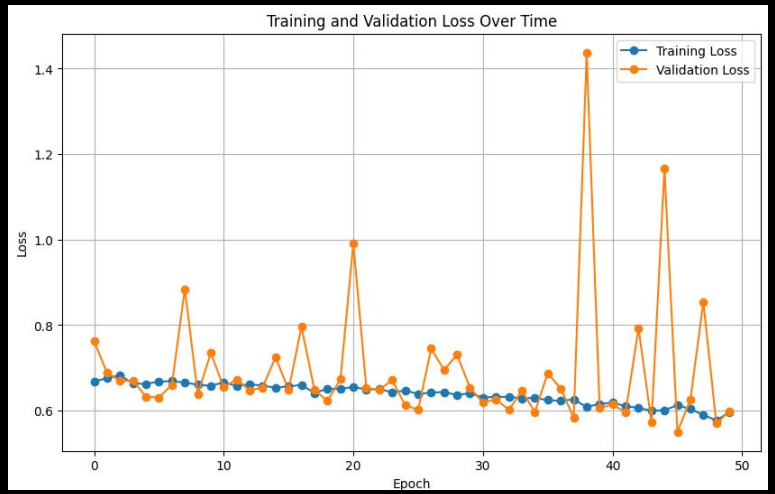
Model Outline (2)

```python
class SkipConnectionConvBlock(nn.Module):
  def __init__(self, in_channel, out_channel, kernel, padding, num_conv_blocks):
    super().__init__()
    self.in_channels = in_channel
    self.out_channels = out_channel
    self.kernel = kernel
    self.padding = padding
    self.num_conv_blocks = num_conv_blocks

    self.conv_blocks = nn.ModuleList()
    self.batch_norms = nn.ModuleList()
    if self.num_conv_blocks % 2 == 1:
      raise Exception('number of conv blocks should be even')

    self.conv_blocks.append(
        torch.nn.Conv3d(
            self.in_channels,
            self.out_channels,
            kernel_size=self.kernel,
            padding=self.padding,
            device=device
        )
    )
    self.batch_norms.append(nn.BatchNorm3d(self.out_channels, device=device))
    skip_input_channel = self.in_channels
    for i in range(1, self.num_conv_blocks):
        if i % 2 == 0:
            input_channel = self.out_channels + skip_input_channel
            skip_input_channel = self.out_channels
        else:
            input_channel = self.out_channels
        self.conv_blocks.append(
            torch.nn.Conv3d(
                input_channel,
                self.out_channels,
                kernel_size=self.kernel,
                padding=self.padding,
                device=device
            )
        )
        self.batch_norms.append(nn.BatchNorm3d(self.out_channels, device=device))
```

- one skip implemented
  - since model is shallow, not much benefit from adding this
- batch size: 4
  - maximum that the model would allow before memory issues
    - ~40GB
  - learning rate decreased to account for this

https://www.researchgate.net/figure/Structure-of-the-Resnet-18-Model_fig1_366608244
https://arxiv.org/pdf/1512.03385

Output and Results

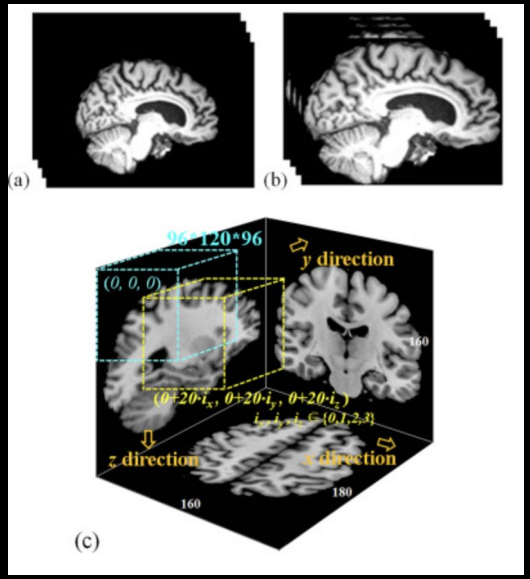


	precision	recall	f1-score	support
0	0.82	0.44	0.58	95
1	0.66	0.92	0.77	111
accuracy			0.70	206
macro avg	0.74	0.68	0.67	206
weighted avg	0.73	0.70	0.68	206

- binary output (0 / 1) representing OASIS dataset's definition of dementia
- Training before N4 Bias correction and using old mask template gave average test accuracy of 54%
  - pre-processing and image cleanup crucial
- spikes in validation loss over training, but more room to improve
  - validation loss still appears steadily decreasing

Challenges and Improvements

3D volume and shape characteristics of hippocampus, parahippocampal gyrus, and entorhinal cortex



https://www.sciencedirect.com/science/article/pii/S0730725X

- Registration and working with template masks difficult
- Memory / RAM issues when working with large 3D image blocks

- Original idea: isolate hippocampal brain tissue (or entorhinal cortex)
  - performed well in similar models with peak accuracies of 89%
  - apply skull stripping technique to hippocampus and other key brain regions

- larger image sets
  - counteracts overfitting among other benefit

- multi-class classification
  - range of severity
  - target various subsets of dementia

- more epochs
  - model still appears to have room to improve

https://nist.mni.mcgill.ca/icbm-152-nonlinear-atlases-2009/
https://nipy.org/nibabel/
https://neuroimaging-data-science.org/content/006-image/003-registration.html
https://www.kaggle.com/code/rasoulisaeid/oasis1-skull-stripping
https://alzres.biomedcentral.com/articles/10.1186/s13195-021-00837-0
https://www.researchgate.net/publication/366856470_Hippocampus_Segmentation-Based_Alzheimer's_Disease_Diagnosis_and_Classification_of_MRI_Images

What did we learn?

- preprocessing methods and techniques
  - intensive, and important for test accuracy
- ResNet structure, and related features
  - impact of learning rate
  - impact of architecture depth
- relevant medical libraries for Pytorch
- many relevant and robust NN models in medical field
- exciting potential for early detection of diseases, and room for improvement!