Design of convolutional neural networks for automatic detection of Alzheimer's disease

Presented at : Proceedings of Machine Learning Research 116:184-201, 2021

Supervisor :
Prof. C. Krishna Mohan
Dept of CSE
IIT Hyderabad

Mentor :
Udaya Kumar Ambati
IIT Hyderabad

Presenter :
Amit Doda
CS22MDS15012
IIT Hyderabad

Agenda

- Recap
  - Problem Statement
  - Paper Methodology
  - Dataset
  - Proposed Model
  - Results
  - Key Components
- Implementation
  - Data Preprocessing
  - Model Training
  - Results
- Challenges
- Novelty
- Conclusion
- References

Problem Statement

- Alzheimer's disease is a **progressive neurodegenerative disorder** that is difficult to diagnose at an early stage.
- Current diagnostic methods are time-consuming, expensive, and are not sensitive enough to detect the disease in its early stages.
- Early detection and diagnosis are crucial for effective treatment and management of the disease.
- There is a growing interest in developing automated methods for Alzheimer's disease detection using machine learning.
- The goal is to design an optimized CNN model that can **accurately classify** brain **MRI scans** as either Alzheimer's or healthy controls, thus improving early detection and diagnosis of Alzheimer's disease.

Related Work

- Initial works applied simple classifiers such as **support vector machines** on features obtained from **volumetric** measurements of the **hippocampus** (Gerardin et al. (2009)) and other brain areas (Plant et al. (2010)).
- Gupta et al. (2013) used pretraining based on a sparse **autoencoder** to perform classification on the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset.
- Hon and Khan (2017) applied state-of-the-art architectures such as **VGG** (Simonyan and Zisserman (2014)) and **Inception Net** (Szegedy et al. (2015)) on the OASIS dataset (Marcus et al. (2010)).
- Cheng et al. (2017) proposed a more computationally-efficient approach based on large 3D patches processed by individual CNNs, which are then combined by an additional CNN to produce the output.
- Lian et al. (2018) proposed a related hierarchical CNN architecture that automatically identifies significant patches. **Siamese networks** were applied by Khvostikov et al. (2018) to distinguish regions of interest around the hippocampus fusing data from multiple imaging modalities.

Challenges

- Many existing works suffer from **data leakage** due to flawed data splits, **biased transfer learning**, or the absence of an independent test set. In the absence of data leakage, CNNs achieved an accuracy of **72-86%** when distinguishing between **AD** and healthy **controls**

- A significant **drop in test accuracy** (to **52 %** for the three-class classification problem considered in the present work) was reported when there was **no patient overlap** between the training and test sets.

Paper Methodology

In this paper we focus on learning using structural brain MRI (T1-weighted scans) to differentiate between:

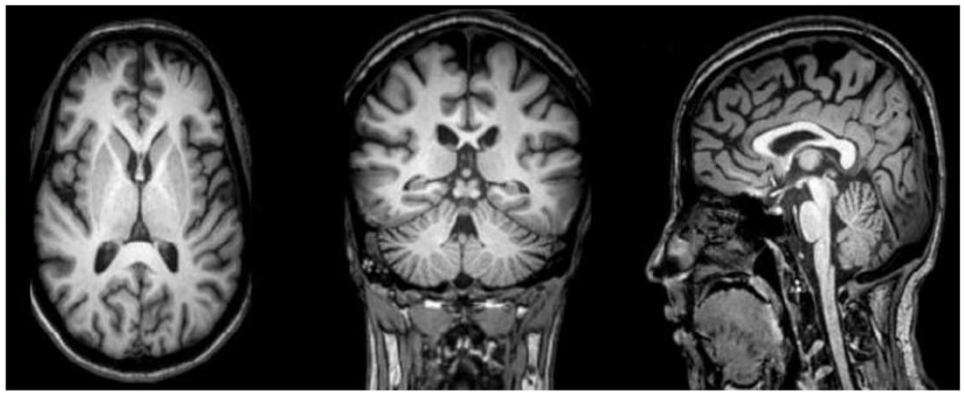
- Cognitively Normal aging (**CN**)
- Alzheimer's Disease (**AD**)
- Mild Cognitive Impairment (**MCI**)

The key components of the architecture are :

- **Instance Normalization**, an alternative to batch normalization introduced originally in the context of style transfer.
- Use of **Small-Sized Kernels** in the first layer to avoid downsampling.
- **Wide Architectures** with large numbers of filters and relatively fewer layers.
- Using **Age** of the patient as an additional feature to the network through an **embedding** inspired by a recent technique from natural language processing

Dataset

- We use T1-weighted structural MRI scans from the **ADNI** (Alzheimer's Disease Neuroimaging Initiative) dataset which have undergone specific image preprocessing steps including **multiplanar reconstruction** (MPR), and corrections of image distortions and non uniformities.
- In total, over **3000** preprocessed scans were used.
- Labels in the ADNI dataset are extracted based on the scores obtained on memory tasks, corrected by education level and other criteria, some of which are subjective.
  - **AD** (mildly demented patients diagnosed with AD)
  - **MCI** (mildly cognitively-impaired patients in prodromal phase of AD)
  - **CN** (elderly control participants)
- MRI scanner can generate three types of orientations of human head as shown
  (a) Axial
  (b) Coronal
  (c) Sagittal

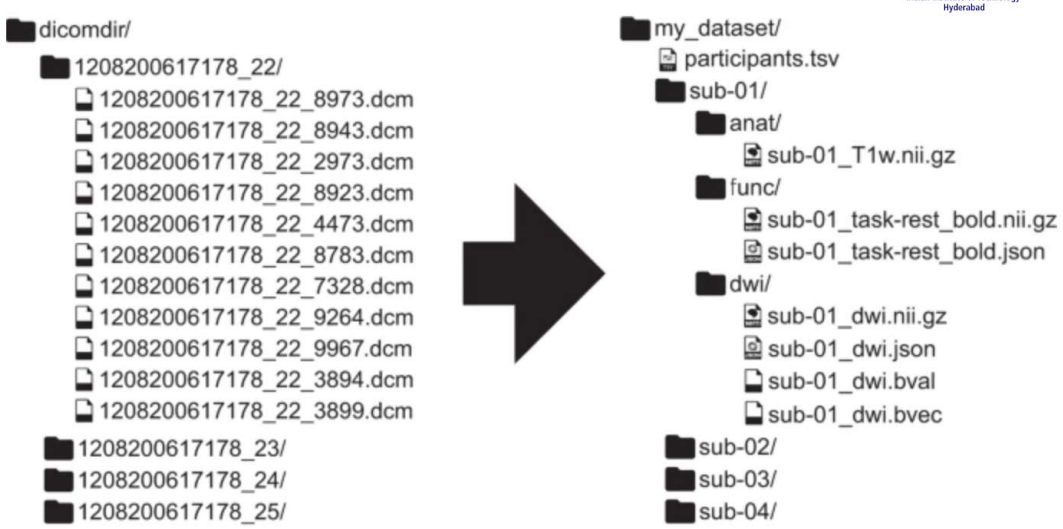


Axial **Coronal** **Sagittal**

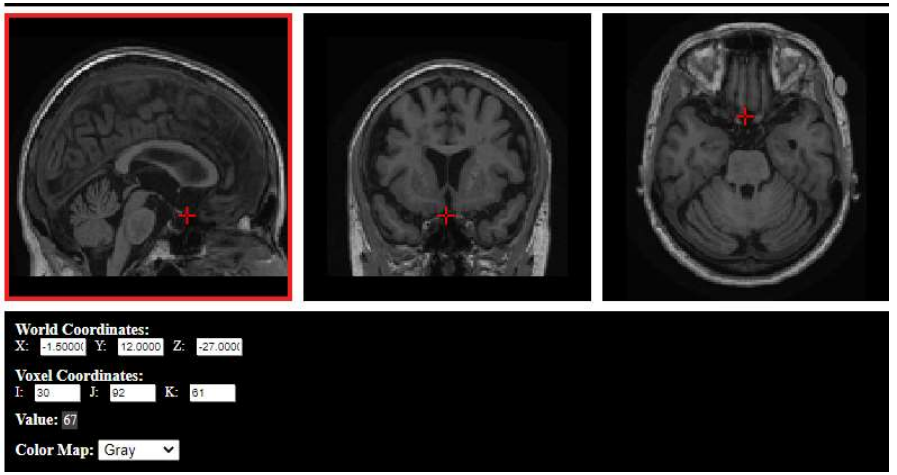
Data Preprocessing

- We first split patients into training, validation and test sets. Then we use **Clinica** software to register the scans to a Dartel template computed exclusively from the training data and normalize them to the Montreal Neurological Institute (MNI) coordinate space.
- The validation and test data are not used to compute any templates in order to **avoid data leakage**.
- The input to the Clinica software is the ADNI scans converted to **BIDS** (brain imaging data structure) format.
- The output is a 3D Image of dimensions **121×145×121** voxels along **sagittal**, **coronal** and **axial** dimensions respectively.

BIDS Structured Dataset



BIDS is a format for standardizing and describing outputs of neuroimaging experiments (left) in a way that is intuitive to understand and easy to use with existing analysis tools (right).



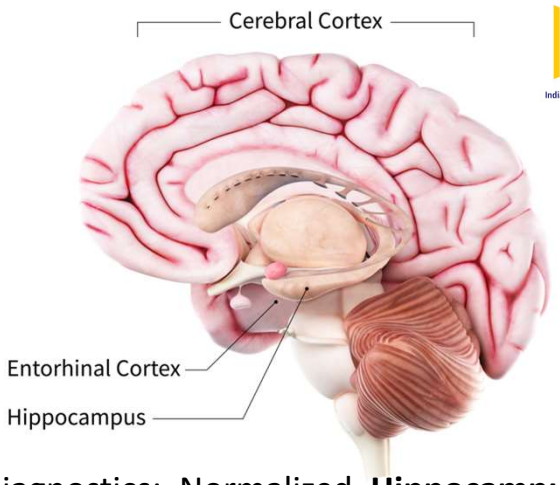
Data Split

- Due to preprocessing and registration errors, the final number of scans in our dataset is **2702**. The subjects in the dataset are split between **training** (70%), **validation** (15%) and **test** (15%) sets.

Split	Class	Num. subjects	Num. Scans	Mean Age (std)
Train	CN	140	567	77.0 (5.4)
	MCI	248	840	75.9 (7.3)
	AD	193	527	76.7 (7.4)
Val	CN	33	126	77.2 (5.6)
	MCI	39	138	73.3 (7.2)
	AD	41	124	76.1 (8.3)
Test	CN	24	105	79.0 (6.1)
	MCI	43	140	76.7 (6.5)
	AD	45	135	76.4 (5.1)

Demographics of our training, validation and test sets after preprocessing.

Proposed Model

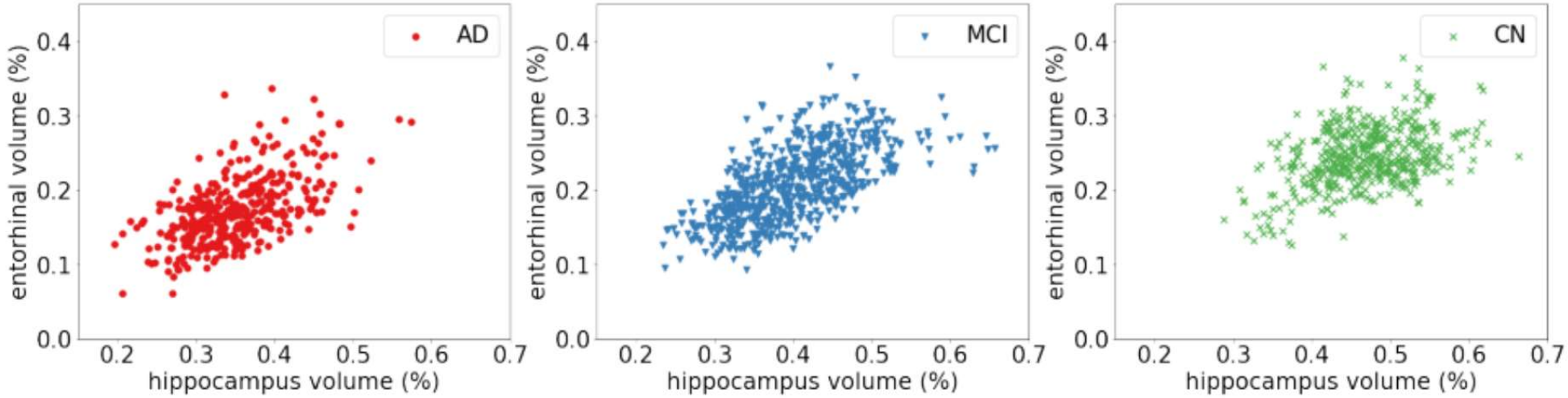


Hippocampus : Hippocampus is a complex S-shaped brain structure embedded deep into temporal lobe. It has a major role in **learning** and **memory**. Studies have shown that it gets affected in a variety of neurological and psychiatric disorders.

Entorhinal : The entorhinal cortex is the gateway for information entering and leaving the hippocampal formation.

Scatterplots of the values of two popular hand-crafted features associated to AD diagnostics: Normalized **Hippocampus volumes** and **Entorhinal volumes**

The features are informative (AD patients tend to have smaller volumes with respect to healthy controls), but they do not enable accurate classification due to the significant overlap between the three classes.



Proposed Model

- The above finding motivates learning discriminative features automatically. Our proposed methodology achieves this using a deep convolutional neural network, inspired by their success in computer vision.

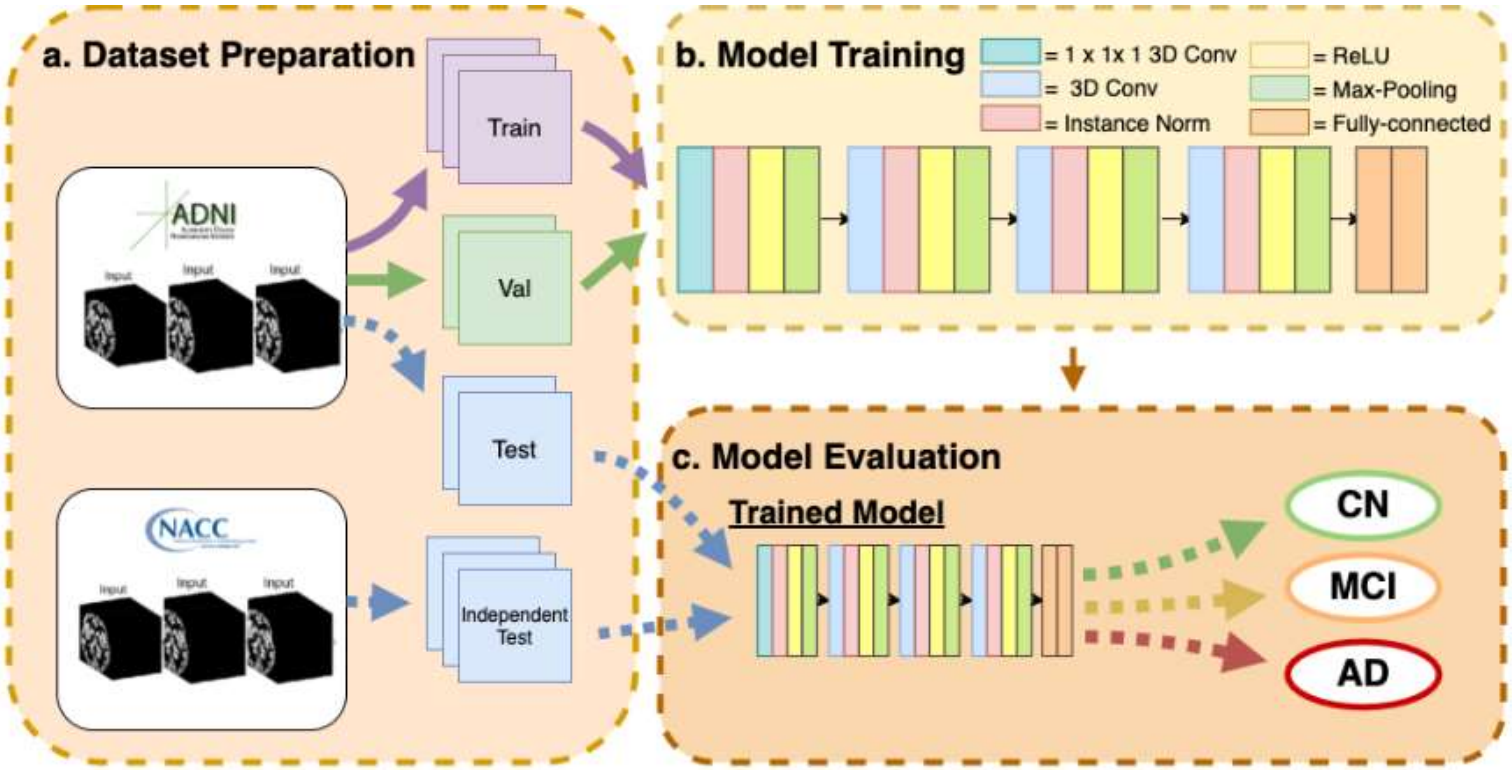
- The proposed architecture is a **3D CNN** model, composed of convolutional, normalization, activation and max-pooling layers.

- We outline several design choices that significantly boost the performance of the network for the task of differentiating between CN, AD, and MCI patients

Block	Layer	Type	Output size
	Inputs		$96 \times 96 \times 96$
1	Conv3D	k1-c4·f-p0-s1-d1	$96 \times 96 \times 96$
	InstanceNorm3D		
	ReLU		
	MaxPool3D	k3-s2	$47 \times 47 \times 47$
2	Conv3D	k3-c32·f-p0-s1-d2	$43 \times 43 \times 43$
	InstanceNorm3D		
	ReLU		
	MaxPool3D	k3-s2	$21 \times 21 \times 21$
3	Conv3D	k5-c64·f-p2-s1-d2	$17 \times 17 \times 17$
	InstanceNorm3D		
	ReLU		
	MaxPool3D	k3-s2	$8 \times 8 \times 8$
4	Conv3D	k3-c64·f-p1-s1-d2	$6 \times 6 \times 6$
	InstanceNorm3D		
	ReLU		
	MaxPool3D	k5-s2	$5 \times 5 \times 5$
FC1		1024	
FC2		3	
Softmax		3	

The backbone architecture. k = kernel size, c = number of channels as a multiple of the **widening factor f**, p = padding size, s = stride and d = dilation.
The **age encoding**, if used, is forward propagated through two linear layers with layer normalization before being added to the output of **FC1**

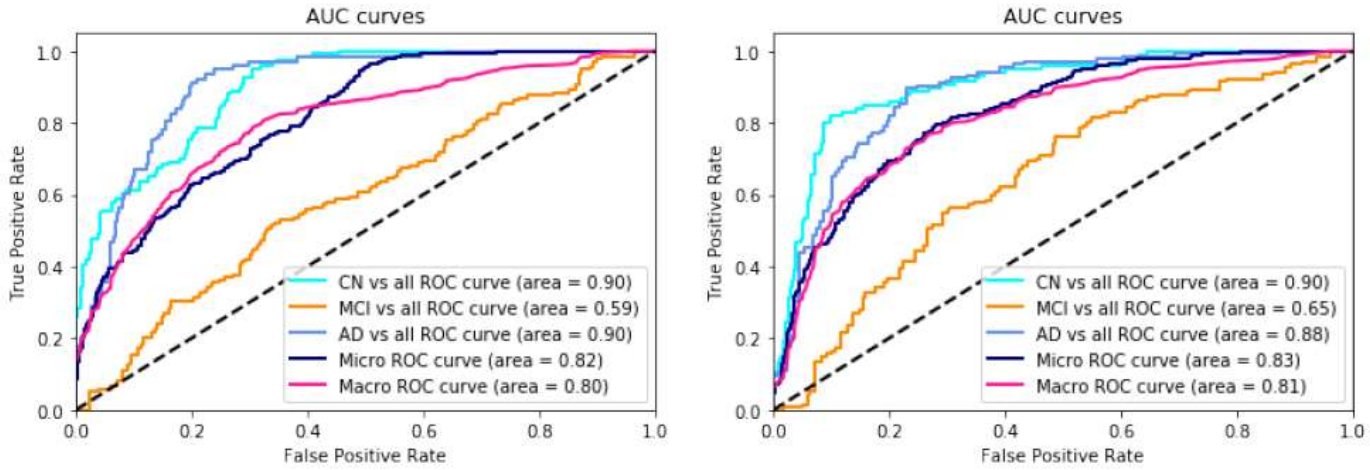
Architecture



Results

Metric

- Our primary metric in this work is standard classification **accuracy** (**Acc**). As the test set is not necessarily balanced, we also use balanced classification accuracy (**Bal-Acc**) which is calculated as the average of the recall of each class.
- We also compute area under the ROC curves (**AUCs**), which are widely used for measuring the predictive accuracy of binary classification problems
- We also calculate micro and macro averages, denoted as Micro-AUC and Macro-AUC respectively.
- Below are the ROC curves obtained on the validation and test set. The model achieves around **90 % AUC** when distinguishing **CN** or **AD** from the other two classes, and **59 – 65 %** when distinguishing **MCI** from the other two classes.



ROC curves on the **validation set** (left) and **test set** (right). Differentiating CN or AD from all other classes results in high AUCs while detecting MCI remains a difficult task.

Results

Comparison with other models

- Our proposed model significantly outperforms previously reported results, as well as the baseline SOTA architectures.
- Incorporating age through the proposed encoding improves performance moderately.

Method	Accuracy	Balanced Acc	Micro-AUC	Macro-AUC
ResNet-18*	50.8%	-	-	-
ResNet-18 pretrained*	56.8%	-	-	-
ResNet-18 3D$^\diamond$	$52.4 \pm 1.8\%$	53.1%	-	-
ResNet-18 3D	$50.1 \pm 1.1\%$	$51.3 \pm 1.0\%$	$71.2 \pm 0.4\%$	$72.4 \pm 0.7\%$
AlexNet 3D	$57.2 \pm 0.5\%$	$56.2 \pm 0.8\%$	$75.1 \pm 0.4\%$	$74.2 \pm 0.5\%$
proposed$^\bullet$	$66.9 \pm 1.2\%$	$67.9 \pm 1.1\%$	$82.0 \pm 0.7\%$	$78.5 \pm 0.7\%$
proposed$^\bullet$ + Age	$\mathbf{68.2 \pm 1.1\%}$	$\mathbf{70.0 \pm 0.8\%}$	$\mathbf{82.0 \pm 0.2\%}$	$\mathbf{80.0 \pm 0.5\%}$

* *Results on 2D ResNets initialized with or without pretrained weights on Imagenet reported by Valliani and Soni (2017).*

$^\diamond$ *3D ResNet with mild modifications, see Fung et al. (2019) for details. The balanced accuracy is computed using the confusion matrix in the paper.*

Comparison of the published models to our best proposed models**. + Age** means that the model incorporates age encodings.

Key Components

Instance Normalization

- Use of Instance Normalization instead of commonly used batch normalization.
- In Instance Normalization, mean and variance are calculated for each individual channel for each individual sample across both spatial dimensions.
- Below, we show that applying **instance** normalization consistently **outperforms batch** normalization for our task of interest.

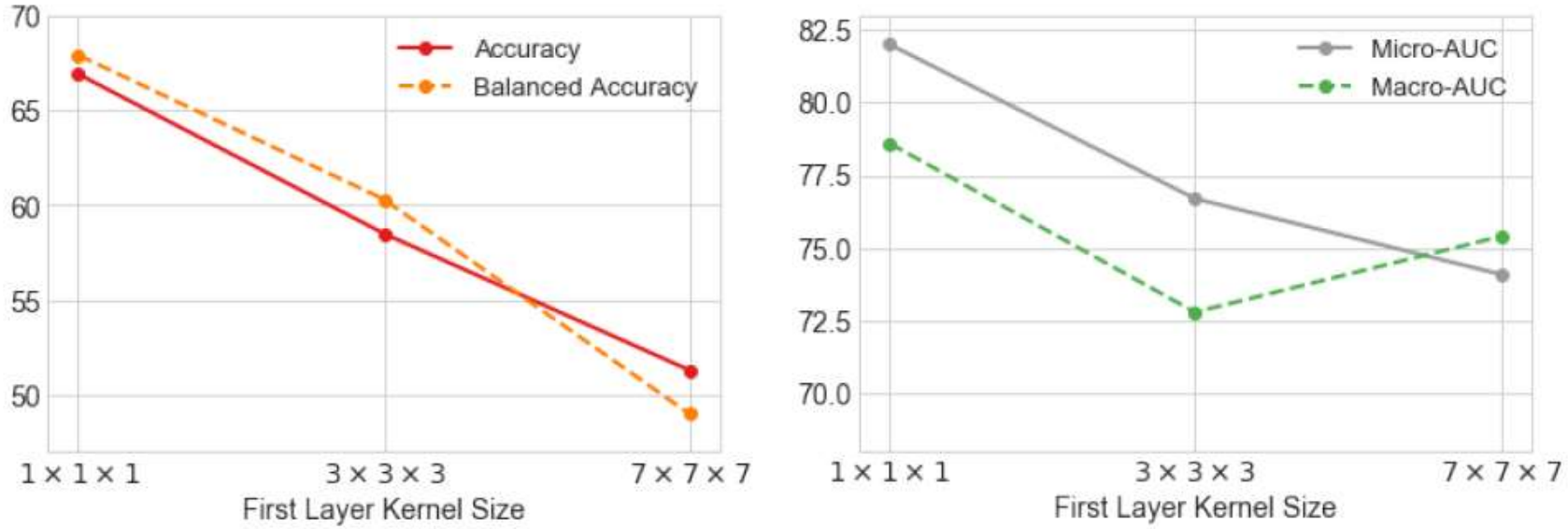
Method	Accuracy	balanced Acc	Micro-AUC	Macro-AUC
$\times 4$ with IN	$\mathbf{63.2 \pm 1.0\%}$	$\mathbf{63.3 \pm 0.9\%}$	$\mathbf{80.5 \pm 0.5\%}$	$\mathbf{77.0 \pm 0.7\%}$
$\times 4$ with BN	$61.8 \pm 1.1\%$	$62.2 \pm 1.1\%$	$77.0 \pm 0.5\%$	$73.0 \pm 0.6\%$
$\times 8$ with IN	$\mathbf{66.9 \pm 1.2\%}$	$\mathbf{67.9 \pm 1.1\%}$	$\mathbf{82.0 \pm 0.7\%}$	$\mathbf{78.5 \pm 0.7\%}$
$\times 8$ with BN	$58.8 \pm 0.9\%$	$60.7 \pm 0.7\%$	$75.9 \pm 0.7\%$	$73.1 \pm 0.8\%$
ResNet-18 with IN	$\mathbf{52.3 \pm 0.8\%}$	$\mathbf{52.7 \pm 1.1\%}$	$\mathbf{74.1 \pm 0.7\%}$	$\mathbf{73.1 \pm 0.9\%}$
ResNet-18 with BN	$50.1 \pm 1.1\%$	$51.3 \pm 1.0\%$	$71.2 \pm 0.4\%$	$72.4 \pm 0.7\%$

Comparison of batch normalization (BN) and Instance normalization (IN) layers on the backbone architecture with widening factor of 4 and 8 and on ResNet-18.

Key Components

Small-sized kernels

- Use of small sized kernel in first layer to **avoid early spatial downsampling**
- The state of art networks such as ResNet and AlexNet use relatively large kernel sizes and strides in their first layer, which dramatically reduce the spatial dimension of their inputs. However, for our task of interest, early downsampling results in significant loss of performance as shown below.

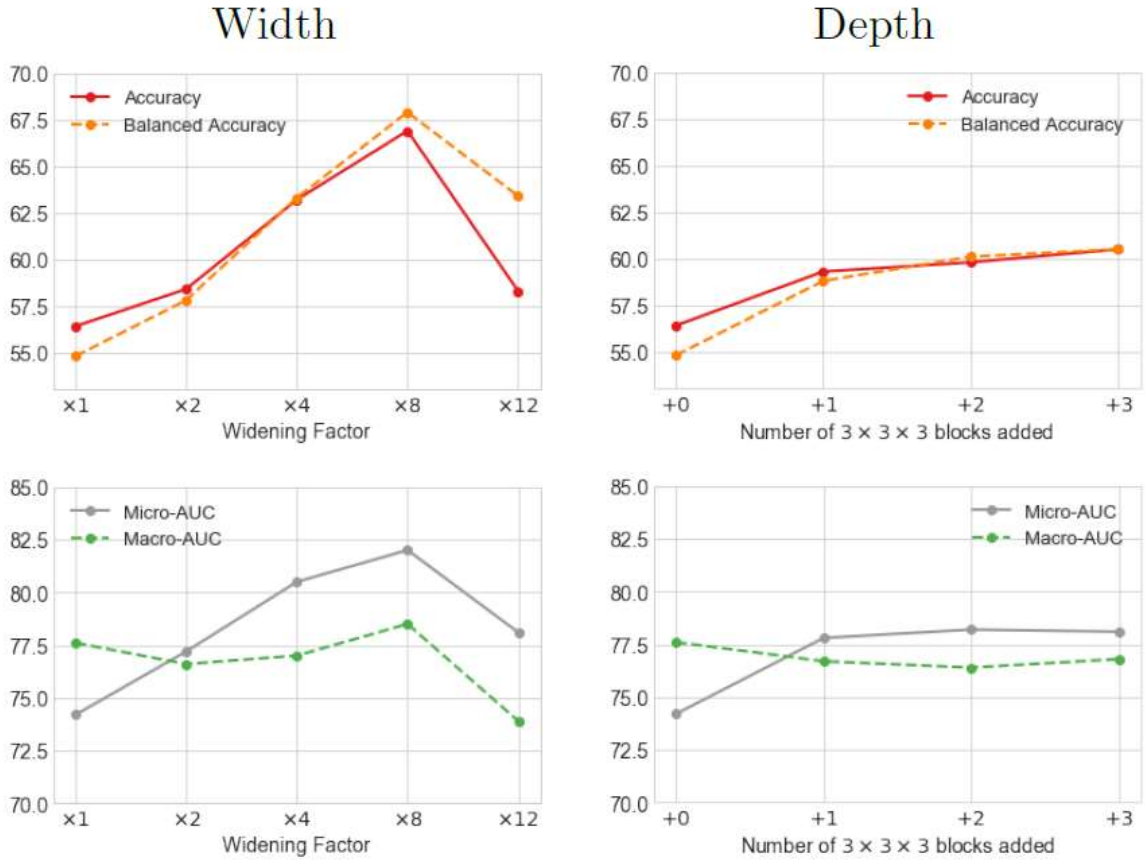


Performances of different first layer kernel sizes. Larger kernel sizes in the first layer result in worse performance.

Key Components

Wider network

- This architecture design favors a wider architecture that is not too deep.
- It is found that increasing the depth of the model only brings marginal gains, whereas **widening** the architecture **improves performance** significantly.
- We also observe that wider networks are often **faster** and **easier** to **train** when compared to deeper networks.



Performance for different widening factors (left) and numbers of additional blocks (right) for backbone architecture.
Wider architectures consistently achieve better performance up until a widening factor of x8. Deeper networks only achieve marginal improvement

Key Components

Age Encodings

- Brains typically **shrink** to some degree in **healthy aging**. This might confuse the model since Alzheimer's disease has a similar effect.
- In order to integrate age information, we encode each age value into a vector and combine the vector with the output of the convolutional layers. we use sinusoidal functions to implement the encoding.

  AE(age,2i) = sin(age/10000^2i/dmodel)

  AE(age,2i+1) = cos(age/10000^2i/dmodel)

  where i = 0, 1, 2, :… dmodel/2 -1 is the dimension and dmodel is the size of the encodings.
- We further transform the age encodings using a few fully connected layers to match the scales and sizes with the visual representation.

Layer	Output size
Linear	512
LayerNorm	512
Linear	1024

Method	Accuracy	Balanced Acc	Micro-AUC	Macro-AUC
No age information	$66.9 \pm 1.2\%$	$67.9 \pm 1.1\%$	$82.0 \pm 0.7\%$	$78.5 \pm 0.7\%$
Proposed age encoding	$\mathbf{68.2 \pm 1.1\%}$	$\mathbf{70.0 \pm 0.8\%}$	$\mathbf{82.0 \pm 0.2\%}$	$\mathbf{80.0 \pm 0.5\%}$
Baseline age encoding	$61.5 \pm 1.4\%$	$62.6 \pm 1.0\%$	$78.6 \pm 1.2\%$	$78.3 \pm 1.1\%$

Comparison of different ways of incorporating the age information using the proposed architecture

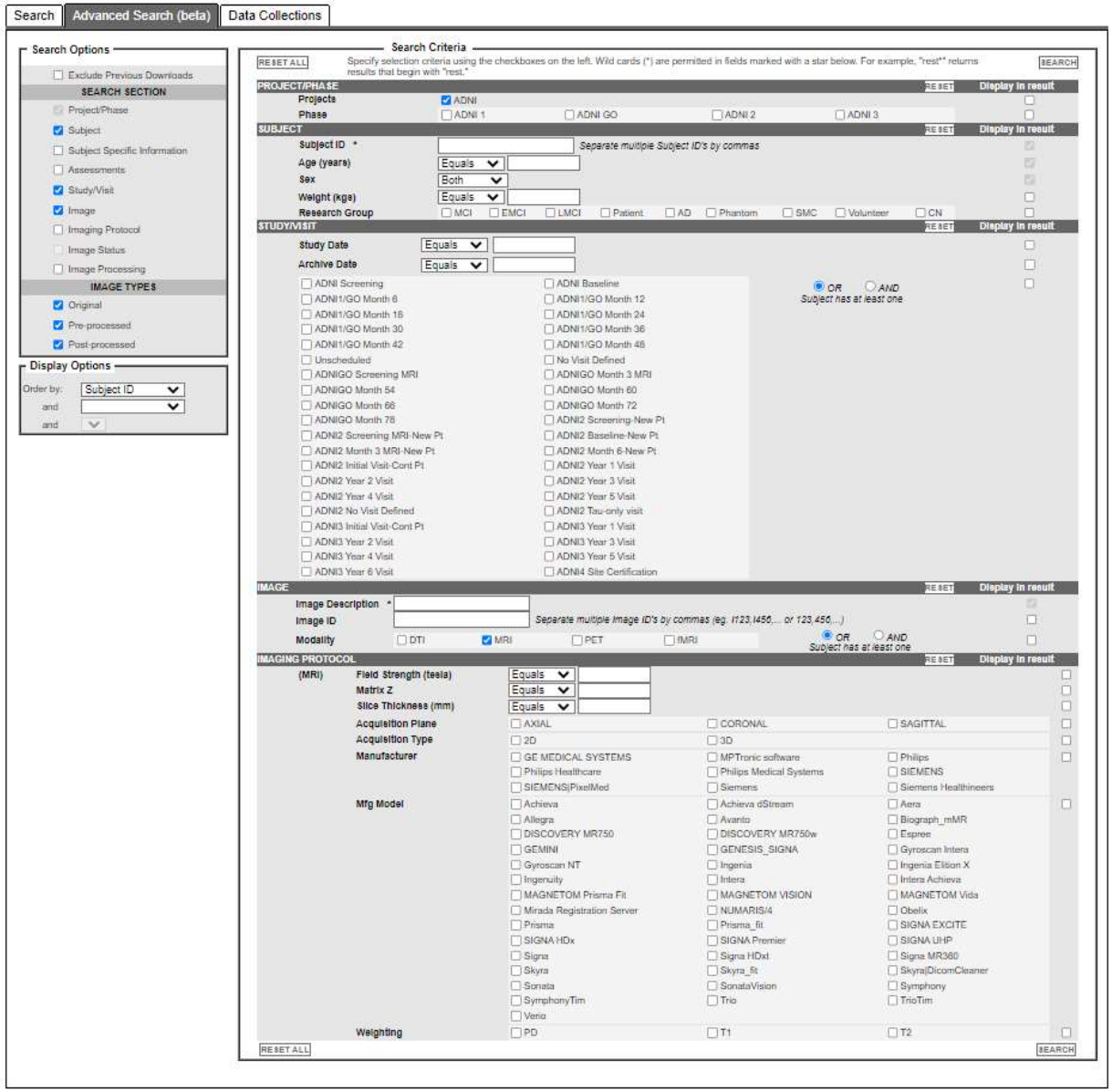
Implementation

- Data Collection
- Code Execution
  - Data Preprocessing
  - Model Training
- Results

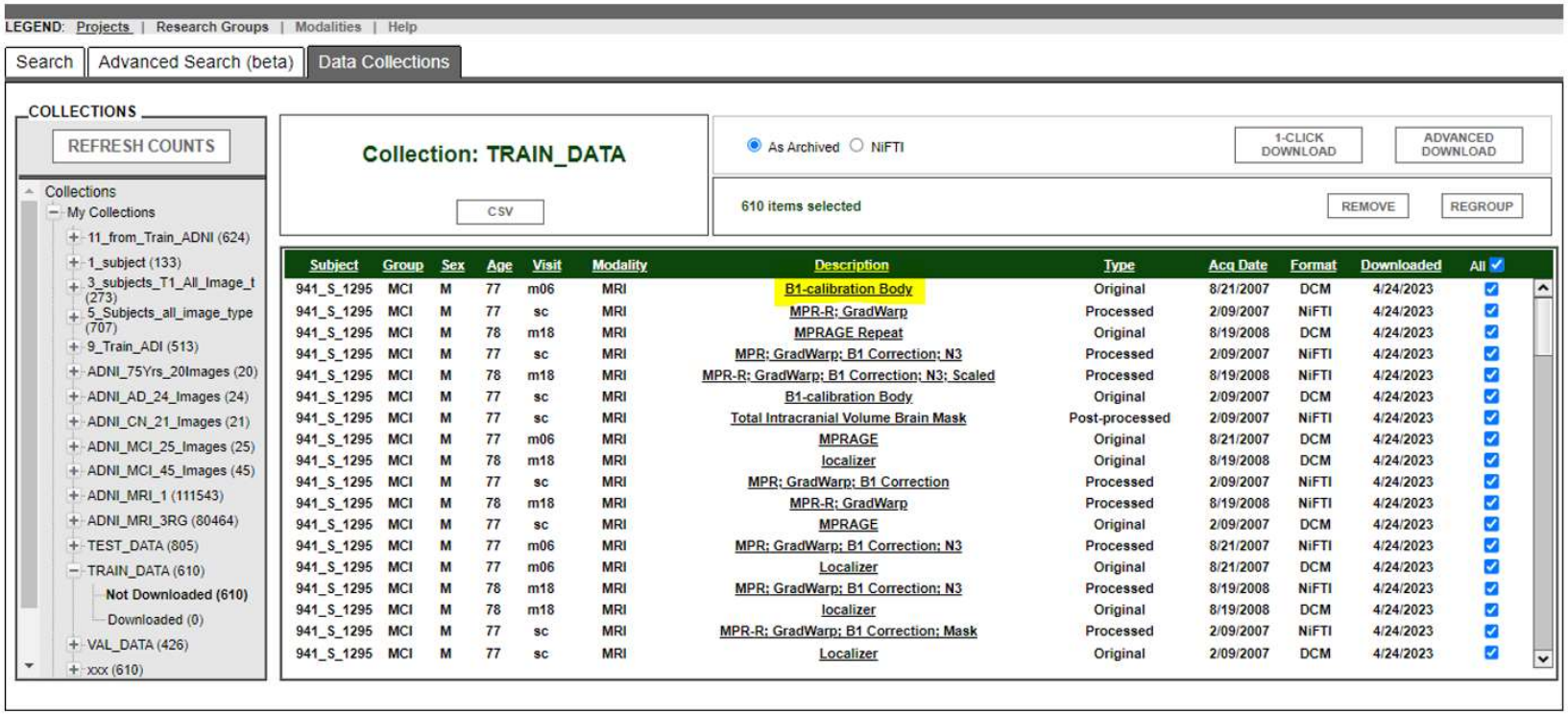
Data Collection : from ADNI

The Image and Data Archive (IDA) is a secure online resource for archiving, exploring and sharing neuroscience data.

Downloaded Alzheimer's Disease Neuroimaging Initiative (ADNI) from the Image and Data Archive (IDA)



Data Collection : Raw Data with Labels



B1-calibration Body — **Subject ID: 941_S_1295**

Modality	: MRI	**Research Group**	: MCI
Image Type	: Original	**Sex**	: Male
Image File Type	: Image Volume	**Visit**	: ADNI1/GO Month 6 (M06)

Acquisition Protocol

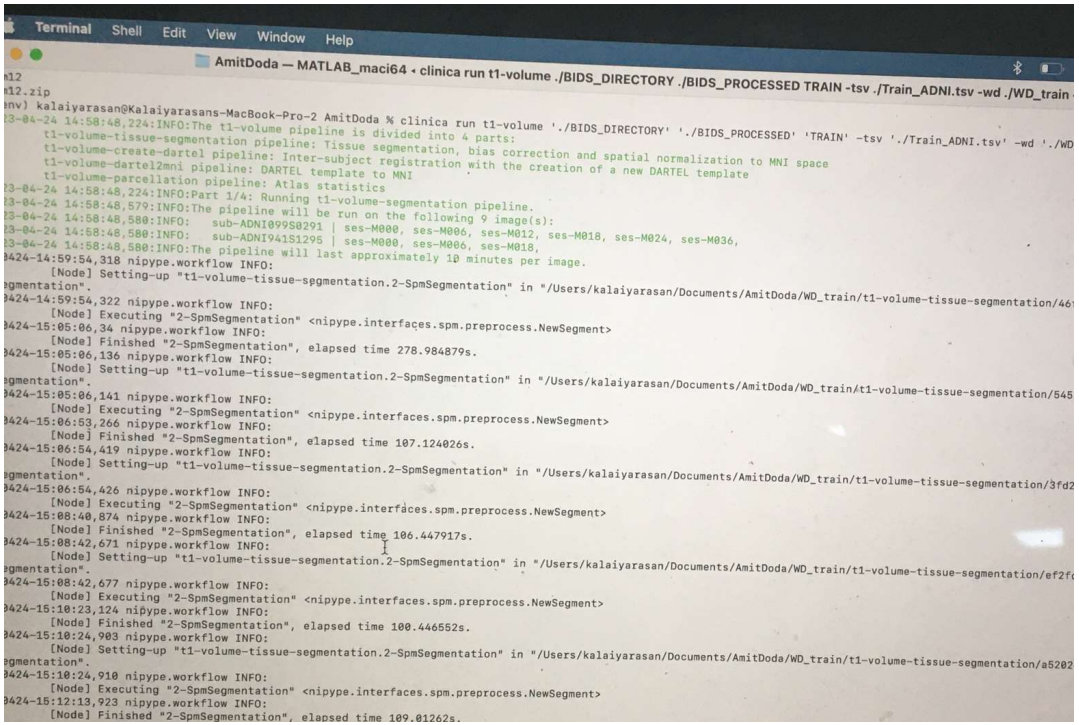
Image Status : AVAILABLE VIEW

Imaging Protocol : Acquisition Plane=SAGITTAL; Acquisition Type=3D; Coil=PA; Field Strength=1.5 tesla; Flip Angle=2.0 degree; Manufacturer=SIEMENS; Matrix X=128.0 pixels; Matrix Y=128.0 pixels; Matrix Z=96.0 ; Mfg Model=Symphony; Pixel Spacing X=2.3 mm; Pixel Spacing Y=2.3 mm; Pulse Sequence=GR; Slice Thickness=2.5 mm; TE=1.3 ms; TI=0.0 ms; TR=3.9 ms; Weighting=T1;

Data Preprocessing : dcm to BIDS t1-volume

```
(clinica-venv) cs22mds15012@instance-2:~$ clinica -v convert adni-to-bids '3_subjects_T1_All_Image_t/ADNI
/home/cs22mds15012/clinica-venv/lib/python3.8/site-packages/clinica/iotools/converters/adni_to_bids/adni_
  adni_merge = pd.read_csv(adni_merge_path)
2023-04-26 18:58:37,478:INFO:Loading a subjects lists provided by the user...
2023-04-26 18:58:37,485:DEBUG:/home/cs22mds15012/BIDS
2023-04-26 18:58:37,486:INFO:Calculating paths of T1 images. Output will be stored in /home/cs22mds15012/
2023-04-26 18:58:39,219:INFO:Paths of T1 images found. Exporting images into BIDS ...
2023-04-26 18:58:39,247:INFO:[T1] Processing subject 003_S_0908in session bl
2023-04-26 18:58:40,562:INFO:[T1] Processing subject 003_S_0908in session m06
2023-04-26 18:58:41,831:INFO:[T1] Processing subject 003_S_0908in session m12
2023-04-26 18:58:43,103:INFO:[T1] Processing subject 003_S_0908in session m18
2023-04-26 18:58:44,363:INFO:[T1] Processing subject 003_S_0908in session m24
2023-04-26 18:58:45,596:INFO:[T1] Processing subject 003_S_0908in session m36
2023-04-26 18:58:46,515:INFO:[T1] Processing subject 003_S_0908in session m48
2023-04-26 18:58:47,757:INFO:[T1] Processing subject 003_S_0908in session m60
2023-04-26 18:58:49,040:INFO:[T1] Processing subject 003_S_0908in session m72
2023-04-26 18:58:50,344:INFO:[T1] Processing subject 003_S_0908in session m84
2023-04-26 18:58:52,223:INFO:[T1] Processing subject 003_S_0908in session m132
2023-04-26 18:58:53,774:INFO:[T1] Processing subject 003_S_0908in session m144
2023-04-26 18:58:55,257:INFO:[T1] Processing subject 003_S_0908in session m156
2023-04-26 18:58:56,656:INFO:[T1] Processing subject 099_S_0291in session bl
2023-04-26 18:58:57,775:INFO:[T1] Processing subject 099_S_0291in session m06
2023-04-26 18:58:59,381:INFO:[T1] Processing subject 099_S_0291in session m12
2023-04-26 18:59:00,625:INFO:[T1] Processing subject 099_S_0291in session m18
2023-04-26 18:59:01,851:INFO:[T1] Processing subject 099_S_0291in session m24
2023-04-26 18:59:02,989:INFO:[T1] Processing subject 099_S_0291in session m36
2023-04-26 18:59:04,716:INFO:[T1] Processing subject 099_S_0291in session m72
2023-04-26 18:59:07,388:INFO:[T1] Processing subject 941_S_1295in session bl
2023-04-26 18:59:09,768:INFO:[T1] Processing subject 941_S_1295in session m06
2023-04-26 18:59:12,210:INFO:[T1] Processing subject 941_S_1295in session m18
2023-04-26 18:59:14,721:DEBUG:T1 conversion done.
2023-04-26 18:59:14,748:INFO:Creating modality agnostic files...
2023-04-26 18:59:14,788:INFO:Creating participants.tsv...
/home/cs22mds15012/clinica-venv/lib/python3.8/site-packages/clinica/iotools/bids_utils.py:114: DtypeWarni
  file_to_read = pd.read_csv(file_to_read_path)
/home/cs22mds15012/clinica-venv/lib/python3.8/site-packages/clinica/iotools/bids_utils.py:114: DtypeWarni
  file_to_read = pd.read_csv(file_to_read_path)
```

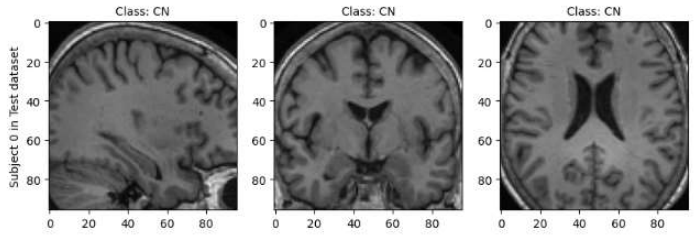
dcm to BIDS



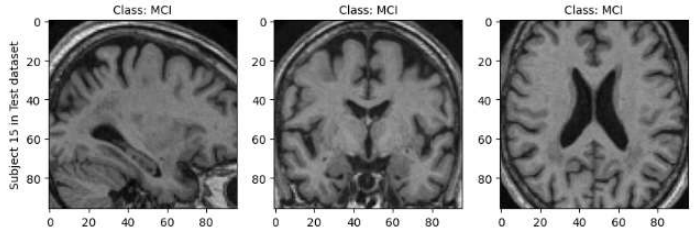
BIDS t1-volume

Code Execution : Loading **pretrained model** and evaluating on small test set

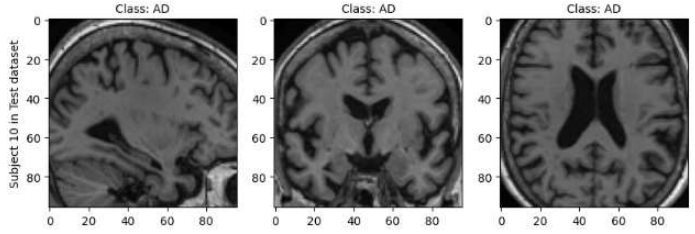
```
visualize_example(Test_dataset,0)
```



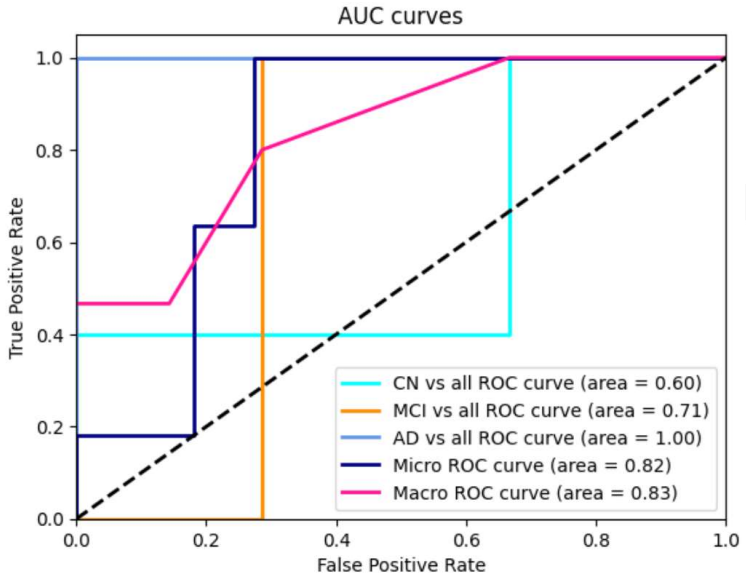

```
[ ] visualize_example(Test_dataset,15)
```




```
[ ] visualize_example(Test_dataset,10)
```



participant_id	session_id	diagnosis	mmse	cdr	cdr_sb	age	examination_date	earliest_time	age_rounde
sub-ADNI033S1016	ses-M00	CN	29.0	0.0	0.0	78.3	2006-11-13	2006-11-13	78.5
sub-ADNI033S1016	ses-M06	CN	27.0	0.0	0.0	78.8	2007-05-30	2006-11-13	79.0
sub-ADNI033S1016	ses-M12	CN	30.0	0.0	0.0	79.3	2007-11-28	2006-11-13	79.5
sub-ADNI033S1016	ses-M24	CN	28.0	0.0	0.0	80.3	2008-12-03	2006-11-13	80.5
sub-ADNI033S1016	ses-M36	CN	29.0	0.0	0.0	81.3	2009-12-02	2006-11-13	81.5
sub-ADNI033S1016	ses-M48	CN	30.0	0.0	0.0	82.2	2010-11-10	2006-11-13	82.5
sub-ADNI130S1290	ses-M00	AD	25.0	0.5	4.0	79.3	2007-02-27	2007-02-27	79.5
sub-ADNI130S1290	ses-M06	AD	21.0	0.5	4.0	79.8	2007-09-10	2007-02-27	80.0
sub-ADNI130S1290	ses-M12	AD	19.0	1.0	5.0	80.3	2008-02-29	2007-02-27	80.5
sub-ADNI126S0606	ses-M00	AD	23.0	0.5	3.5	68.5	2006-08-02	2006-08-02	68.5
sub-ADNI126S0606	ses-M06	AD	25.0	1.0	5.0	69.0	2007-02-05	2006-08-02	69.0
sub-ADNI126S0606	ses-M12	AD	25.0	1.0	5.0	69.4	2007-08-02	2006-08-02	69.5
sub-ADNI126S0606	ses-M24	AD	22.0	1.0	7.0	70.4	2008-07-24	2006-08-02	70.5
sub-ADNI128S1407	ses-M00	MCI	24.0	0.5	2.5	74.6	2007-06-25	2007-06-25	74.5
sub-ADNI128S1407	ses-M06	MCI	26.0	0.5	2.5	75.1	2008-01-28	2007-06-25	75.0
sub-ADNI128S1407	ses-M24	MCI	23.0	0.5	3.5	76.7	2009-08-27	2007-06-25	77.0
sub-ADNI029S0866	ses-M00	CN	28.0	0.0	0.0	80.0	2006-10-05	2006-10-05	80.0
sub-ADNI029S0866	ses-M06	CN	29.0	0.0	0.0	80.4	2007-04-04	2006-10-05	80.5
sub-ADNI029S0866	ses-M12	CN	29.0	0.0	0.0	80.9	2007-10-04	2006-10-05	81.0
sub-ADNI029S0866	ses-M24	CN	30.0	0.0	0.0	82.0	2008-10-09	2006-10-05	82.0
sub-ADNI029S0866	ses-M36	CN	29.0	0.0	0.0	83.0	2009-10-22	2006-10-05	83.0
sub-ADNI094S1397	ses-M00	AD	24.0	1.0	4.5	55.1	2007-05-15	2007-05-15	55.0
sub-ADNI094S1397	ses-M06	AD	22.0	1.0	4.5	55.5	2007-11-13	2007-05-15	55.5
sub-ADNI094S1397	ses-M12	AD	11.0	1.0	9.0	56.1	2008-05-15	2007-05-15	56.0
sub-ADNI133S0792	ses-M18	MCI	30.0	0.5	0.5	74.9	2008-03-17	2006-09-25	75.0
sub-ADNI133S0792	ses-M00	MCI	30.0	0.5	1.0	73.5	2006-09-25	2006-09-25	73.5
sub-ADNI133S0792	ses-M06	MCI	29.0	0.5	0.5	73.9	2007-03-21	2006-09-25	74.0
sub-ADNI133S0792	ses-M12	MCI	29.0	0.5	0.5	74.4	2007-09-25	2006-09-25	74.5
sub-ADNI133S0792	ses-M24	MCI	28.0	0.5	1.0	75.4	2008-09-18	2006-09-25	75.5




```
test_model(all_acc, all_balanced_acc,all_auc)
```

```
Mean Acc  0.6636363636363638
STD Acc  0.0416597790450531
Mean Balanced Acc  0.6666666666666667
std Balanced Acc  1.1102230246251565e-16
Micro mean:  0.8181818181818181
Micro std:  0.021943666193969844
Macro mean:  0.7962648809523809
Macro std:  0.026552204998858332
```

Code Execution

Config file

```
cat config.yaml
```

```
file_name: ./saved_model/1007_pooling_age
data:
  data_root_dir: data/
  dir_to_scans: ./FULL_DATA/subjects
  dir_to_tsv: ./FULL_DATA/TSV_FOR_TRAIN
  batch_size: 4
  val_batch_size: 2
  workers: 2
  percentage_usage: 1.0
exp_name: 1007_pooling_age
visdom:
  port: 8064
  server:
model:
  arch: ours
  input_channel: 1
  nhid: 512
  feature_dim: 1024
  n_label: 3
  expansion: 8
  num_blocks: 0
  type_name: conv3x3x3
  norm_type: Instance
adv_model:
  nhid: 36
  out_dim: 12
mmse_model:
  nhid: 64
training_parameters:
  use_age: False
  pretrain:  #dir to saved model here
  max_iter: 16000
  start_epoch: 0
  epochs: 30
  print_freq: 5
  max_grad_l2_norm:
  report_interval: 100
  snapshot_interval: 1000

optimizer:
  method: SGD
  par:
    lr: 0.01
    weight_decay: 0.000
```

Model Training from scratch on small train sample

```
[ ] ! python main.py
```

```
    Validation [0/7]        Time 0.692 (0.692)      Data 0.432 (0.432)      Loss 5.7886 (5.7886)
    Validation [5/7]        Time 0.272 (0.343)      Data 0.000 (0.073)      Loss 0.0057 (3.2678)
    Epoch [20]: Validation Accuracy 58.333
    saved_model
    Epoch: [21][0/9]        Time 2.235 (2.235)      Data 1.045 (1.045)      Loss 0.1744 (0.1744)
    Epoch: [21][5/9]        Time 1.172 (1.347)      Data 0.000 (0.175)      Loss 0.4384 (0.2719)
    Validation [0/7]        Time 0.715 (0.715)      Data 0.455 (0.455)      Loss 9.9523 (9.9523)
    Validation [5/7]        Time 0.267 (0.347)      Data 0.000 (0.077)      Loss 0.0020 (5.2102)
    Epoch [21]: Validation Accuracy 58.333
    saved_model
    Epoch: [22][0/9]        Time 2.167 (2.167)      Data 0.978 (0.978)      Loss 0.1323 (0.1323)
    Epoch: [22][5/9]        Time 1.175 (1.337)      Data 0.000 (0.165)      Loss 0.7827 (0.6461)
    Validation [0/7]        Time 0.808 (0.808)      Data 0.547 (0.547)      Loss 6.1226 (6.1226)
    Validation [5/7]        Time 0.271 (0.360)      Data 0.000 (0.092)      Loss 0.0897 (3.1200)
    Epoch [22]: Validation Accuracy 66.667
    saved_model
    Epoch: [23][0/9]        Time 2.797 (2.797)      Data 1.584 (1.584)      Loss 0.1585 (0.1585)
    Epoch: [23][5/9]        Time 1.157 (1.439)      Data 0.000 (0.266)      Loss 0.3422 (0.5861)
    Validation [0/7]        Time 0.799 (0.799)      Data 0.546 (0.546)      Loss 7.8977 (7.8977)
    Validation [5/7]        Time 0.270 (0.360)      Data 0.000 (0.091)      Loss 0.0197 (4.1058)
    Epoch [23]: Validation Accuracy 66.667
    saved_model
    Epoch: [24][0/9]        Time 2.844 (2.844)      Data 1.611 (1.611)      Loss 0.0783 (0.0783)
    Epoch: [24][5/9]        Time 1.173 (1.447)      Data 0.000 (0.269)      Loss 0.1149 (0.3799)
    Validation [0/7]        Time 0.769 (0.769)      Data 0.509 (0.509)      Loss 8.7516 (8.7516)
    Validation [5/7]        Time 0.272 (0.375)      Data 0.000 (0.105)      Loss 0.0316 (4.4081)
    Epoch [24]: Validation Accuracy 66.667
    saved_model
    Epoch: [25][0/9]        Time 2.822 (2.822)      Data 1.591 (1.591)      Loss 0.0349 (0.0349)
    Epoch: [25][5/9]        Time 1.174 (1.448)      Data 0.000 (0.266)      Loss 0.9839 (0.3059)
    Validation [0/7]        Time 1.175 (1.175)      Data 0.916 (0.916)      Loss 7.9771 (7.9771)
    Validation [5/7]        Time 0.268 (0.489)      Data 0.000 (0.225)      Loss 0.0050 (4.0618)
    Epoch [25]: Validation Accuracy 66.667
    saved_model
    Epoch: [26][0/9]        Time 2.173 (2.173)      Data 0.974 (0.974)      Loss 0.1579 (0.1579)
    Epoch: [26][5/9]        Time 1.185 (1.342)      Data 0.000 (0.164)      Loss 0.2858 (0.1972)
    Validation [0/7]        Time 1.014 (1.014)      Data 0.762 (0.762)      Loss 11.1706 (11.1706)
    Validation [5/7]        Time 0.273 (0.401)      Data 0.000 (0.128)      Loss 0.0037 (5.6385)
    Epoch [26]: Validation Accuracy 66.667
    saved_model
    Epoch: [27][0/9]        Time 2.218 (2.218)      Data 1.012 (1.012)      Loss 0.0727 (0.0727)
    Epoch: [27][5/9]        Time 1.177 (1.347)      Data 0.000 (0.169)      Loss 0.0721 (0.1090)
    Validation [0/7]        Time 0.802 (0.802)      Data 0.540 (0.540)      Loss 7.9799 (7.9799)
    Validation [5/7]        Time 0.273 (0.360)      Data 0.000 (0.091)      Loss 0.0196 (4.2744)
    Epoch [27]: Validation Accuracy 58.333
    saved_model
    Epoch: [28][0/9]        Time 2.103 (2.103)      Data 0.911 (0.911)      Loss 0.0367 (0.0367)
    Epoch: [28][5/9]        Time 1.176 (1.327)      Data 0.000 (0.153)      Loss 0.0449 (0.8310)
    Validation [0/7]        Time 0.847 (0.847)      Data 0.585 (0.585)      Loss 5.6035 (5.6035)
    Validation [5/7]        Time 0.270 (0.371)      Data 0.000 (0.098)      Loss 0.0109 (2.8687)
    Epoch [28]: Validation Accuracy 58.333
    saved_model
    Epoch: [29][0/9]        Time 2.130 (2.130)      Data 0.933 (0.933)      Loss 0.0455 (0.0455)
    Epoch: [29][5/9]        Time 1.175 (1.339)      Data 0.000 (0.161)      Loss 0.1777 (0.1726)
    Validation [0/7]        Time 0.718 (0.718)      Data 0.461 (0.461)      Loss 4.8243 (4.8243)
    Validation [5/7]        Time 0.266 (0.345)      Data 0.000 (0.078)      Loss 0.1197 (2.3942)
    Epoch [29]: Validation Accuracy 66.667
    saved_model
```

Model Achieved 66.67 % Validation accuracy in 30 epochs

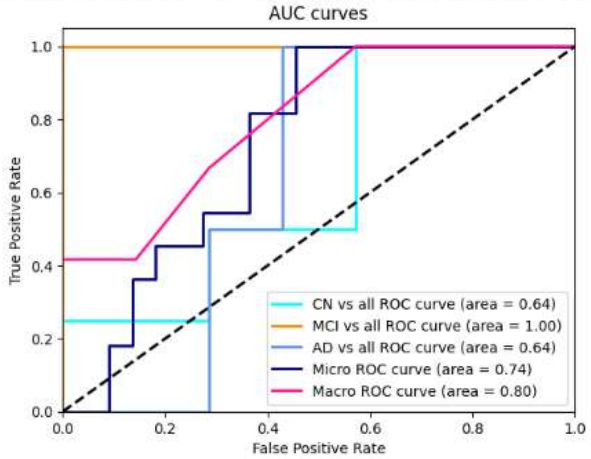
Results on small test set

```
[25] model = build_model(cfg)
     #model_file_name = 'age_expansion_8'
     model_file_name = '1007_pooling_age_train_perc_100.0_expansion_0'
```

Results on Test Data using model trained from scratch

```
[26] all_acc, all_balanced_acc, all_auc = evaluation_models(model_file_name,Test_loader, expansion_list = [8], use_age = True, norm_type= 'Instance')
```

```
Iteration:  0
['model.image_embedding_model.conv.conv0_s1.weight', 'model.image_embedding_model.conv.conv0_s1.bias', 'model.image_embedding_model.conv.conv1_s1.weight', 'model
['image_embedding_model.conv.conv0_s1.weight', 'image_embedding_model.conv.conv0_s1.bias', 'image_embedding_model.conv.conv1_s1.weight', 'image_embedding_model.c
```




```
[27] test_model(all_acc, all_balanced_acc,all_auc)
```

```
Mean Acc  0.5181818181818181
STD Acc  0.09136250564655354
Mean Balanced Acc  0.5238888888888888
std Balanced Acc  0.05550275268448906
Micro mean:  0.7413223140495867
Micro std:  0.025080976703295024
Macro mean:  0.8282440476190475
Macro std:  0.03992617133905823
```

Challenges

- Good amount of medical domain knowledge is needed.
- The ADNI MRI dataset is very bulky.
- Medical imaging requires good amount of preprocessing which is time consuming and compute expensive operation.

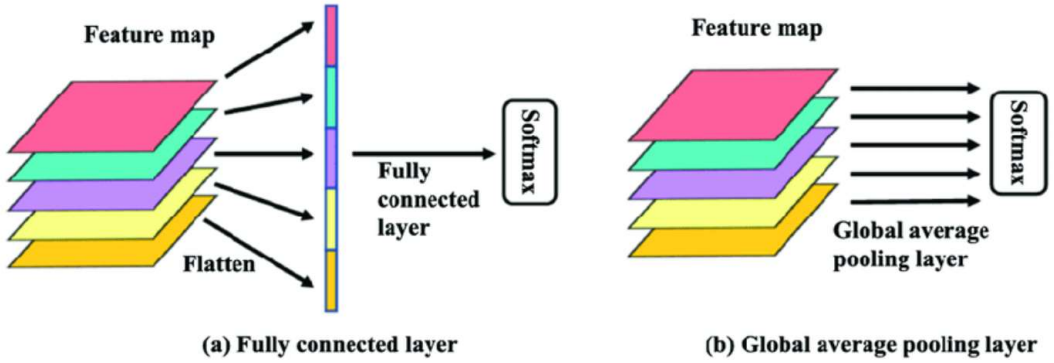
Novelty/ Future Work

Proposal : Use of **Global Average Pooling** (GAP) layer instead of fully connected layers FC1, FC2

Improved Performance: Medical images often contain a large number of small and subtle features which can be effectively captured by GAP, which computes the average value of each feature map.

Efficient Training: The use of GAP can reduce the computational cost of the network, which can help to reduce the training time.

Interpretability: GAP can provide interpretability by understanding the contribution of each feature map to the final classification decision.



Conclusion

- In this paper, a novel 3D CNN architecture is developed to perform three-way classification between patients with Alzheimer's disease, patients with mild cognitive impairment, and healthy controls.

- The architecture combines different elements (instance normalization, small kernels, wider layers, and an encoding of the patient's age) to achieve a significant gain in classification accuracy, demonstrated on completely held-out dataset.

References

1. FMRIB software library. https://fsl.fmrib.ox.ac.uk/fsl. released: 2019-03-11.
2. Statistical parametric mapping. www.fil.ion.ucl.ac.uk/spm/. released: 2014-10-01.
3. ADNI. ADNI website. http://adni.loni.usc.edu/.
4. ADNI. Alzheimer's disease neuroimaging initiative 2 procedures manual, 2008.
5. ARAMIS Lab. Aramis lab. www.aramislab.fr.
6. John Ashburner. A fast diffeomorphic image registration algorithm. Neuroimage, 38(1): 95–113, 2007.
7. Karl B¨ackstr¨om, Mahmood Nazari, Irene Yu-Hua Gu, and Asgeir Store Jakola. An efficient 3D deep convolutional network for Alzheimer's disease diagnosis using MR images. In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pages 149–153. IEEE, 2018.
8. Esther E Bron, Marion Smits, Wiesje M Van Der Flier, Hugo Vrenken, Frederik Barkhof, Philip Scheltens, Janne M Papma, Rebecca ME Steketee, Carolina M´endez Orellana, Rozanna Meijboom, et al. Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the caddementia challenge. NeuroImage, 111:562–579, 2015.
9. Danni Cheng, Manhua Liu, Jianliang Fu, and Yaping Wang. Classification of MR brain images by combination of multi-CNNs for AD diagnosis. In Ninth International Conference on Digital Image Processing (ICDIP 2017), volume 10420, page 1042042. International Society for Optics and Photonics, 2017.
10. Clinica. Clinica software platform. http://www.clinica.run. released: 2019-05-23.
11. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.
12. Kathryn A Ellis, Ashley I Bush, David Darby, Daniela De Fazio, Jonathan Foster, Peter Hudson, Nicola T Lautenschlager, Nat Lenzo, Ralph N Martins, Paul Maruff, et al. The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of Alzheimer's disease. International Psychogeriatrics, 21(4):672–687, 2009.
13. Alan C Evans, D Louis Collins, SR Mills, ED Brown, RL Kelly, and Terry M Peters. 3D statistical neuroanatomical models from 305 MRI volumes. In 1993 IEEE conference record nuclear science symposium and medical imaging conference, pages 1813–1817. IEEE, 1993.
14. Bruce Fischl. Freesurfer. Neuroimage, 62(2):774–781, 2012.
15. GB Frisoni, MP Laakso, A Beltramello, C Geroldi, A Bianchetti, H Soininen, and M Trabucchi. Hippocampal and entorhinal cortex atrophy in frontotemporal dementia and Alzheimer's disease. Neurology, 52(1):91–91, 1999.
16. Yi Ren Fung, Ziqiang Guan, Ritesh Kumar, Joie Yeahuay Wu, and Madalina Fiterau. Alzheimer's disease brain MRI classification: Challenges and insights. arXiv preprint arXiv:1906.04231, 2019.
17. Emilie Gerardin, Ga¨el Ch´etelat, Marie Chupin, R´emi Cuingnet, B´eatrice Desgranges, HoSung Kim, Marc Niethammer, Bruno Dubois, St´ephane Leh´ericy, Line Garnero, et al. Multidimensional classification of hippocampal shape features discriminates Alzheimer's disease and mild cognitive impairment from normal aging. Neuroimage, 47(4):1476–1486, 2009.
18. Ashish Gupta, Murat Ayhan, and Anthony Maida. Natural image bases to represent neuroimaging data. In International conference on machine learning, pages 987–994, 2013.
19. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
20. Marcia Hon and Naimul Mefraz Khan. Towards Alzheimer's disease classification through transfer learning. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 1166–1169. IEEE, 2017.

21. Ehsan Hosseini-Asl, Georgy Gimel'farb, and Ayman El-Baz. Alzheimer's disease diagnostics by a deeply supervised adaptable 3D convolutional network. arXiv preprint arXiv:1607.00556, 2016.

22. Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision, pages 1501–1510, 2017.

23. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

24. William Jagust. Imaging the evolution and pathophysiology of Alzheimer disease. Nature Reviews Neuroscience, 19(11):687, 2018.

25. Alexander Khvostikov, Karim Aderghal, Jenny Benois-Pineau, Andrey Krylov, and Gwenaelle Catheline. 3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies. arXiv preprint arXiv:1801.05968, 2018.

26. Stephanos Leandrou, I Mamais, Styliani Petroudi, Panicos A Kyriacou, Constantino Carlos Reyes-Aldasoro, and Constantinos S Pattichis. Hippocampal and entorhinal cortex volume changes in Alzheimer's disease patients and mild cognitive impairment subjects. In 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), pages 235–238. IEEE, 2018.

27. Chunfeng Lian, Mingxia Liu, Jun Zhang, and Dinggang Shen. Hierarchical fully convolutional network for joint atrophy localization and Alzheimer's disease diagnosis using structural MRI. IEEE transactions on pattern analysis and machine intelligence, 2018.

28. Daniel S Marcus, Anthony F Fotenos, John G Csernansky, John C Morris, and Randy L Buckner. Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults. Journal of cognitive neuroscience, 22(12):2677–2684, 2010.

29. Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford R Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. Ways toward an early diagnosis in Alzheimer's disease: the Alzheimer's disease neuroimaging initiative (ADNI). Alzheimer's & Dementia, 1(1):55–66, 2005.

30. National Center for Health Statistics. Deaths and mortality, May 2017. URL https: //www.cdc.gov/nchs/fastats/deaths.htm. [Online; posted 07-May-2017].

31. Ruth Peters. Ageing and the brain. Postgraduate medical journal, 82(964):84–88, 2006.

32. Claudia Plant, Stefan J Teipel, Annahita Oswald, Christian B¨ohm, Thomas Meindl, Janaina Mourao-Miranda, Arun W Bokde, Harald Hampel, and Michael Ewers. Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease. Neuroimage, 50(1):162–174, 2010.

33. Shannon L Risacher and Andrew J Saykin. Neuroimaging and other biomarkers for Alzheimer's disease: the changing landscape of early detection. Annual review of clinical psychology, 9:621–648, 2013.

34. K Servick. Another major drug candidate targeting the brain plaques of Alzheimer's disease has failed. what's left. Science, 10, 2019.

35. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

36. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034, 2013.

37. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

38. Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022, 2016.

39. Aly Valliani and Ameet Soni. Deep residual nets for improved Alzheimer's diagnosis. In BCB, page 615, 2017.

40. Gary W Van Hoesen, Bradley T Hyman, and Antonio R Damasio. Entorhinal cortex pathology in Alzheimer's disease. Hippocampus, 1(1):1–8, 1991.

41. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

42. Junhao Wen, Elina Thibeau-Sutre, Jorge Samper-Gonzalez, Alexandre Routier, Simona Bottani, Stanley Durrleman, Ninon Burgos, and Olivier Colliot. Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation. arXiv preprint arXiv:1904.07773, 2019.

43. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.

Thank You