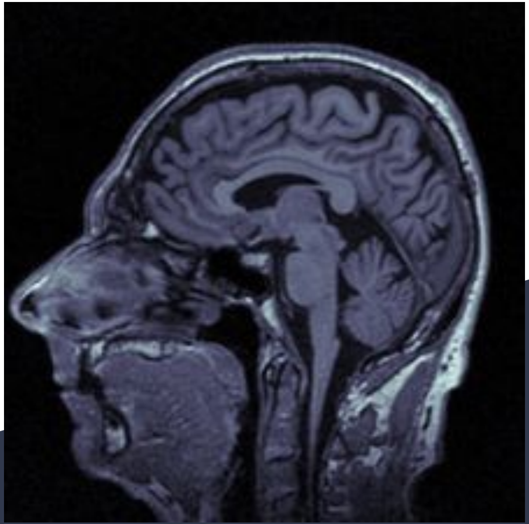
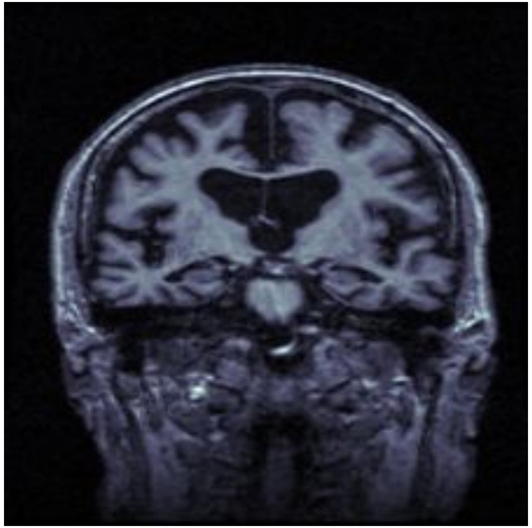
Augmenting and Ensembling CNN Models to Increase the Accuracy of Alzheimer's Disease MRI Classification



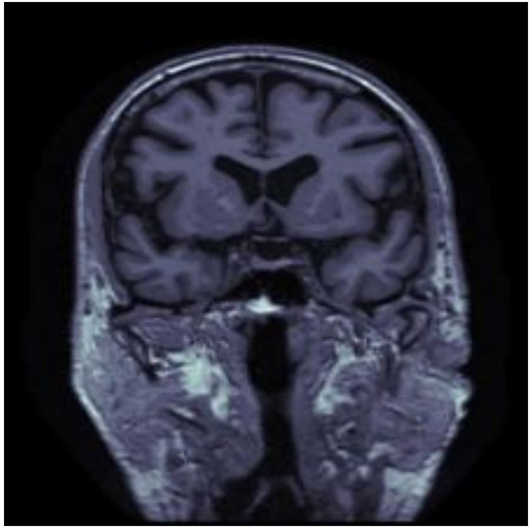
Capstone
Spring 2022
Kristin Levine

The Task:

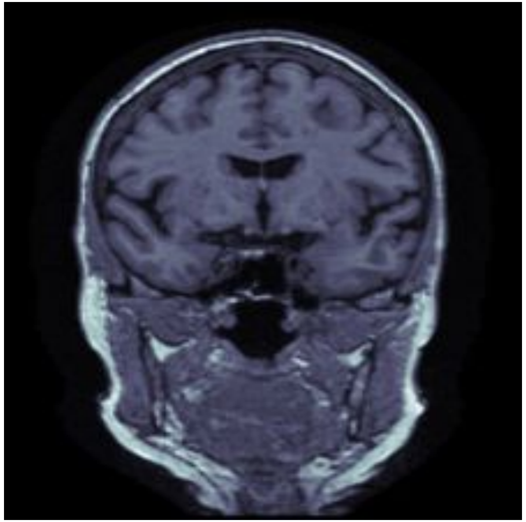
Use CNNs model to classify brain MRIs by their clinical diagnosis



Alzheimer’s Disease (AD)



Cognitively Normal (CN)



Mild Cognitive Impairment (MCI)

Why this problem:

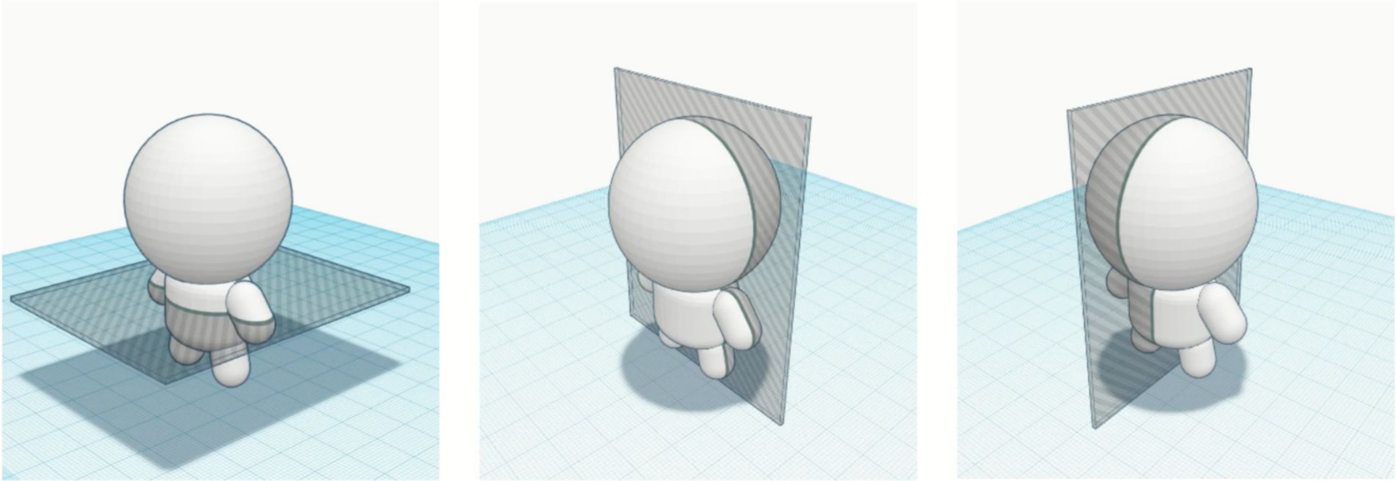
- Prevalence of Alzheimer’s Disease in the US
  - Big Problem – approximately 6.5 millions Americans
  - No treatment
- Humans aren’t that great at this task
  - Even trained neurologists get it wrong a lot of the time
  - One study – Radiologists accuracy of 57.5 and 70%
- Good medical access and care for everyone
  - In DC, we have lots of doctors and specialists
  - In rural areas, lack of radiologists or neurologists to accurately read the MRI

About MRI images

- Magnetic resonance imaging (MRI) non-invasive produces 3D images of parts of the body
- Dementia research – mainly used to take measurements of certain regions of the brain
- Differences between machines/people move
- CNN to literally just LOOK at the images

MRI – 3 dimensions

Planes and Orientations

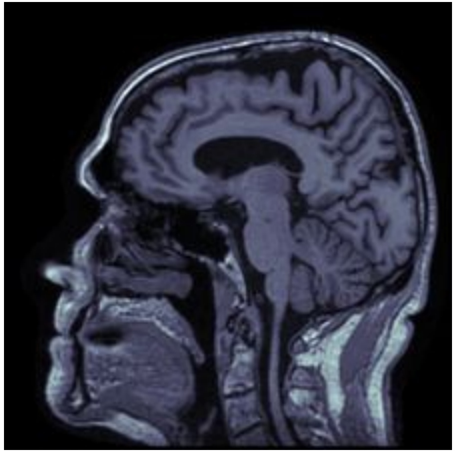


Axial or Transversal Plane | Coronal or Frontal plane | Sagittal plane

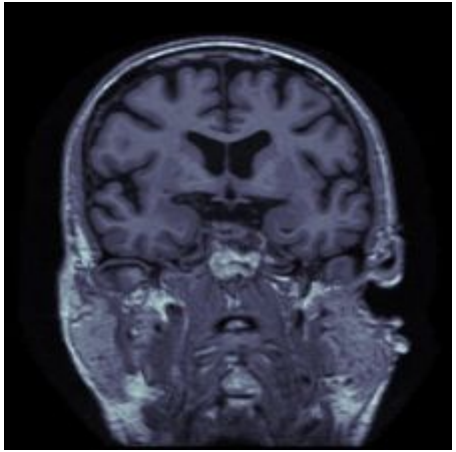
From Udemy: Deep Learning with PyTorch for Medical Image Analysis class

MRI – 3 dimensions

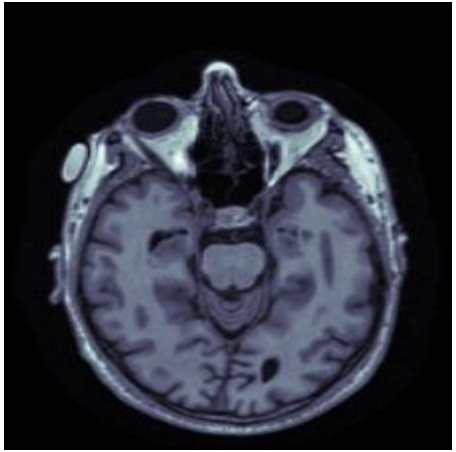
Slices in different directions



Sagittal View



Coronal View



Axial View

Previous studies/concerns:

- Many previous studies binary
  - Alzheimer’s Disease vs Cognitively Normal
  - May to too late to treat once have full-blown AD
  - Being able to identify mild cognitive impairment stage very useful

- Lack of data
  - Not that many correctly labeled images

- My goals
  - Multiclass classification
  - Explore ways of augmenting data

Questions:

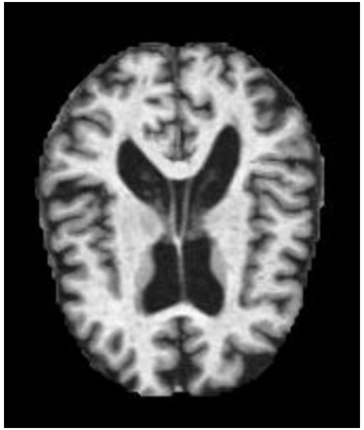
- Is there a particular pre-trained CNN model that works best?

- Does augmentation of images help improve models? If so, what types help the most?

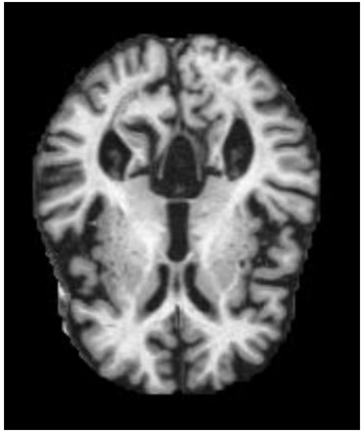
- How much does ensembling improve models?

Four Categories

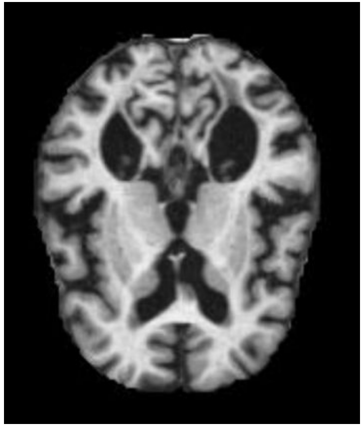
Use CNN model to classify brain MRIs by their clinical diagnosis



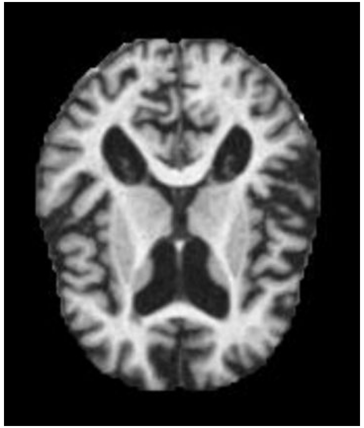
Non-impaired



Very mild impairment



Mild impairment



Moderate impairment

Data Sources: Kaggle

- [Alzheimer's Dataset (4 class of Images)]

Train Set	# Images
Non-impaired	2560
Very mild impairment	1792
Mild impairment	717
Moderate impairment	52

Test Set	# Images
Non-impaired	640
Very mild impairment	448
Mild impairment	179
Moderate impairment	12

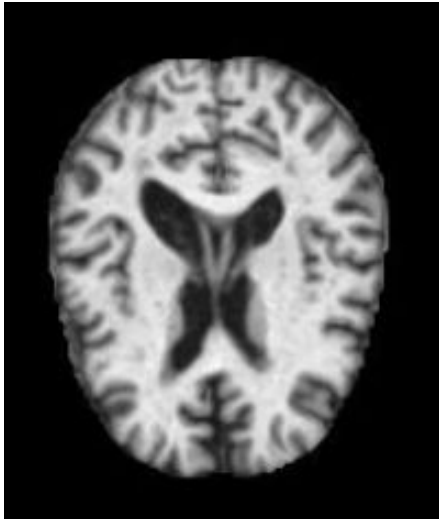
Creating Validation Set

- Used 70/30 split

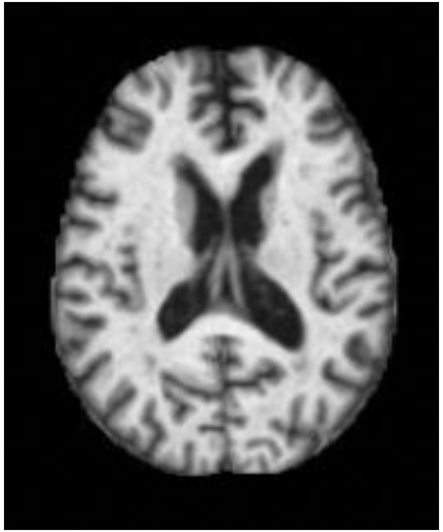
Train Set	# Images
Non-impaired	1792
Very mild impairment	1254
Mild impairment	502
Moderate impairment	36

Validation Set	# Images
Non-impaired	768
Very mild impairment	538
Mild impairment	215
Moderate impairment	16

Augmenting Training Set – Flipping



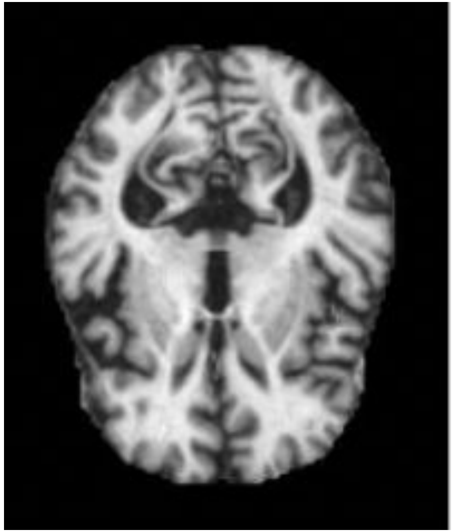
Original image



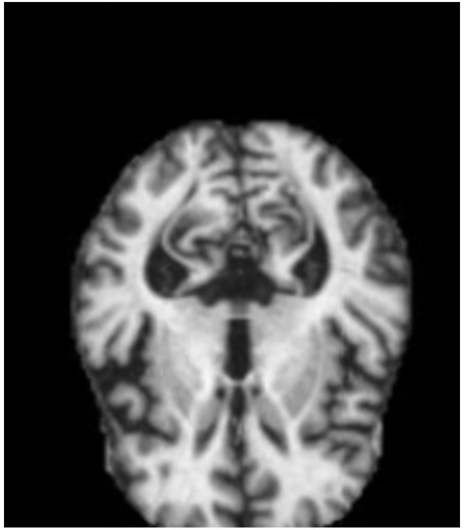
Flipped image

Flip Train	# Images
Non-impaired	2008
Very mild impairment	2008
Mild impairment	2008
Moderate impairment	144

Augmenting Training Set – Shifting



Original image



Shifted image

Shifted Train	# Images
Non-impaired	2008
Very mild impairment	2008
Mild impairment	2008
Moderate impairment	2016

CNN Models used

- ResNet50
  - First CNN model to introduce idea skip connection (2015)
  - Input shape (224, 224, 3)

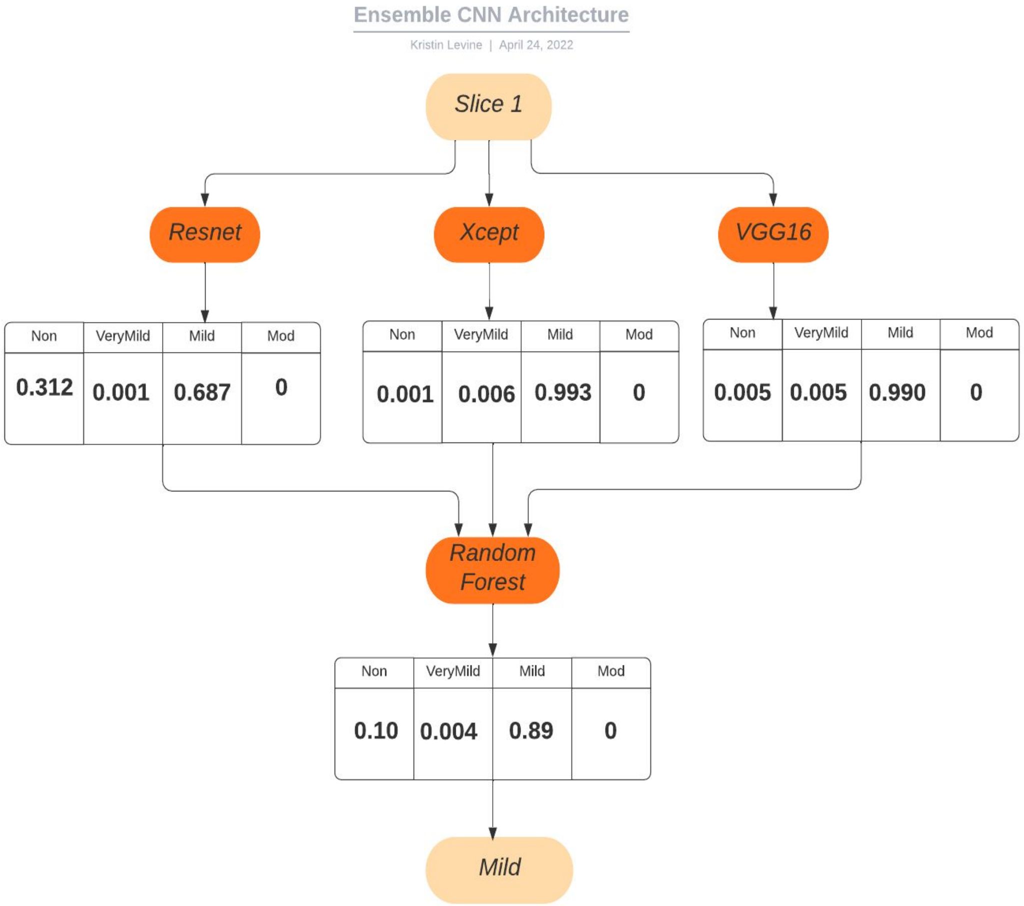
- VGG16
  - Smaller filter, deeper network (2014)
  - Input shape (224, 224, 3)

- Xception
  - Improves upon Inception model (2017)
  - Input shape (299, 299, 3)

Modeling

- 3 pretrained CNNs (ResNet, VGG16, Xception)
- 3 training sets (No Augment, Flipped, Shifted)
- Created 9 models
- Picked the best of each type of CNN
- Combined into the final prediction

Ensemble CNN Architecture



Individual Model Results

Model	Augmentation	Train Acc	Val Acc	Test Acc
ResNet	None	1.00	0.9740	0.7303
VGG16	None	0.8597	0.8699	0.6529
Xception	None	0.9980	0.9453	0.6701
ResNet	Flip	0.9992	0.9818	0.7224
VGG16	Flip	0.9162	0.8530	0.6482
Xception	Flip	0.9990	0.9655	0.7177
ResNet	Shift	0.9981	0.9876	0.7482
VGG16	Shift	0.9778	0.8783	0.6904
Xception	Shift	0.9994	0.9681	0.7209

Ensembled Model

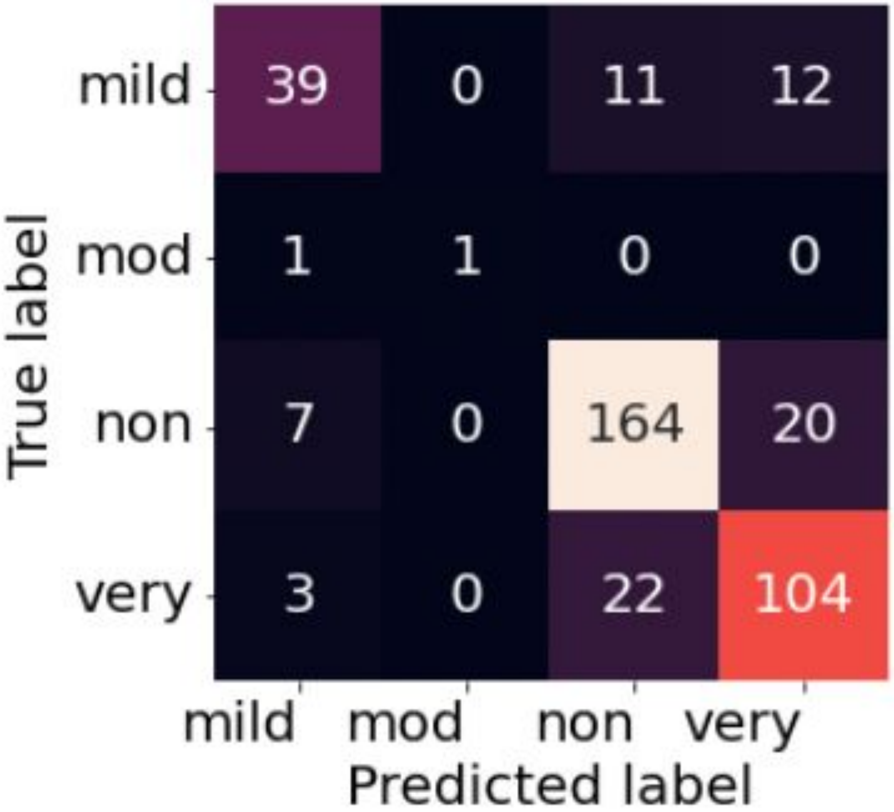
Results Using All Features:

Classification Report:

	precision	recall	f1-score	support
mild	0.78	0.63	0.70	62
mod	1.00	0.50	0.67	2
non	0.83	0.86	0.85	191
very	0.76	0.81	0.78	129
accuracy			0.80	384
macro avg	0.84	0.70	0.75	384
weighted avg	0.80	0.80	0.80	384

Accuracy : 80.20833333333334

Ensembled Model



Improvement – shifting images

Model	Test Acc (No Aug)	Test Acc (Shifted)	Improvement
ResNet	0.7303	0.7482	0.0179
VGG16	0.6529	0.6904	0.0375
Xception	0.6701	0.7209	0.0508

Improvement – ensembling

Model	Test Acc (Shifted)	Test Acc (Ensembled)	Improvement
ResNet	0.7482	0.8021	0.0539
VGG16	0.6904	0.8021	0.1117
Xception	0.7209	0.8021	0.0812

Part 1 Conclusions:

- Augmenting data can improve test set accuracy 2-5%

- Ensembling can improve test set accuracy 5-11%

- Pre-trained models performed similarly

Attempts to create own dataset

- Very little info about this dataset on Kaggle
- No info how patients diagnosed
- No genetic or clinical info
- Wanted to see if I could create own dataset give me more control over these variables

Issues trying to address:

- Data leakage
  - Lack of data – multiple slices from each person
  - Images from the same person may be in train AND test
  - Test set may contain images model has already seen
  - Control for multiple visits as well

- Slice-wise predictions vs. patient-wise predictions

Data Sources: ADNI

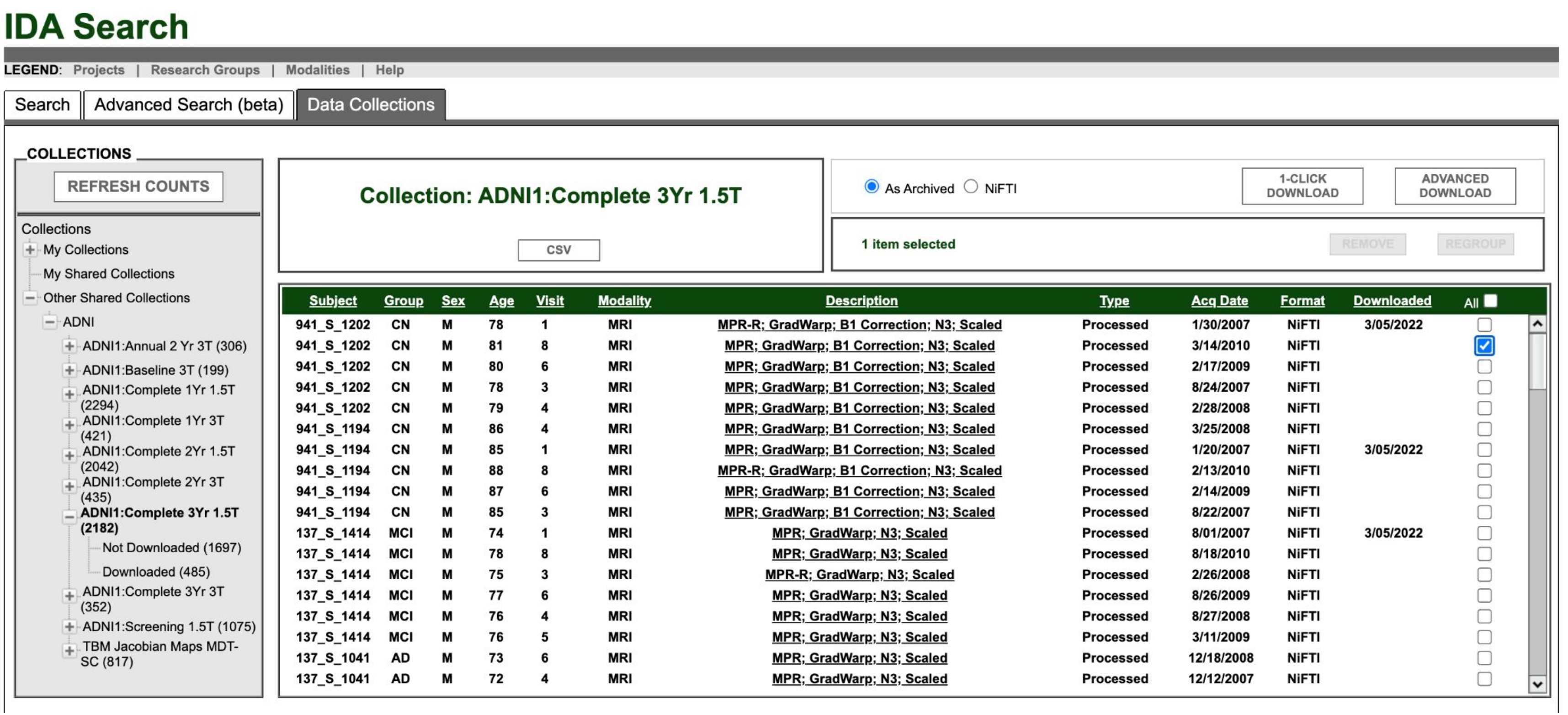
- Alzheimer's Disease Neuroimaging Initiative
  - multicenter longitudinal study
  - clinical, imaging, genetic, and biochemical biomarkers for AD

- Started in 2004 as a private-public partnership involving 20 companies and NIH

- Most of datasets publicly available online – derived from ADNI

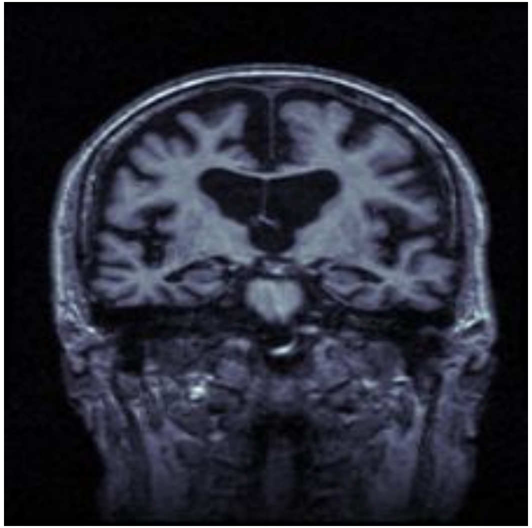
- https://adni.loni.usc.edu/

Data Sources: ADNI

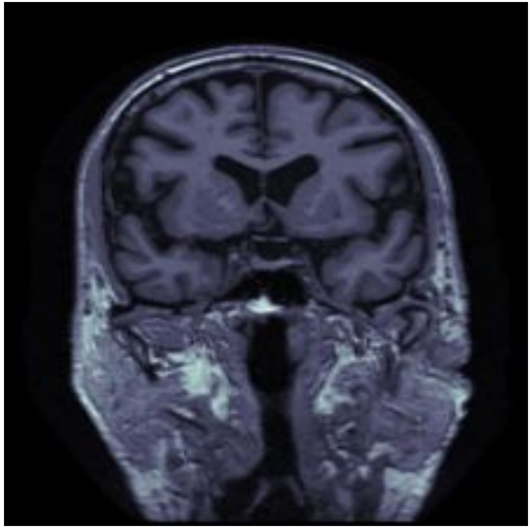


Subject	Group	Sex	Age	Visit	Modality	Description	Type	Acq Date	Format	Downloaded	All
941_S_1202	CN	M	78	1	MRI	MPR-R; GradWarp; B1 Correction; N3; Scaled	Processed	1/30/2007	NiFTI	3/05/2022	☐
941_S_1202	CN	M	81	8	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	3/14/2010	NiFTI		☑
941_S_1202	CN	M	80	6	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	2/17/2009	NiFTI		☐
941_S_1202	CN	M	78	3	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	8/24/2007	NiFTI		☐
941_S_1202	CN	M	79	4	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	2/28/2008	NiFTI		☐
941_S_1194	CN	M	86	4	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	3/25/2008	NiFTI		☐
941_S_1194	CN	M	85	1	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	1/20/2007	NiFTI	3/05/2022	☐
941_S_1194	CN	M	88	8	MRI	MPR-R; GradWarp; B1 Correction; N3; Scaled	Processed	2/13/2010	NiFTI		☐
941_S_1194	CN	M	87	6	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	2/14/2009	NiFTI		☐
941_S_1194	CN	M	85	3	MRI	MPR; GradWarp; B1 Correction; N3; Scaled	Processed	8/22/2007	NiFTI		☐
137_S_1414	MCI	M	74	1	MRI	MPR; GradWarp; N3; Scaled	Processed	8/01/2007	NiFTI	3/05/2022	☐
137_S_1414	MCI	M	78	8	MRI	MPR; GradWarp; N3; Scaled	Processed	8/18/2010	NiFTI		☐
137_S_1414	MCI	M	75	3	MRI	MPR-R; GradWarp; N3; Scaled	Processed	2/26/2008	NiFTI		☐
137_S_1414	MCI	M	77	6	MRI	MPR; GradWarp; N3; Scaled	Processed	8/26/2009	NiFTI		☐
137_S_1414	MCI	M	76	4	MRI	MPR; GradWarp; N3; Scaled	Processed	8/27/2008	NiFTI		☐
137_S_1414	MCI	M	76	5	MRI	MPR; GradWarp; N3; Scaled	Processed	3/11/2009	NiFTI		☐
137_S_1041	AD	M	73	6	MRI	MPR; GradWarp; N3; Scaled	Processed	12/18/2008	NiFTI		☐
137_S_1041	AD	M	72	4	MRI	MPR; GradWarp; N3; Scaled	Processed	12/12/2007	NiFTI		☐

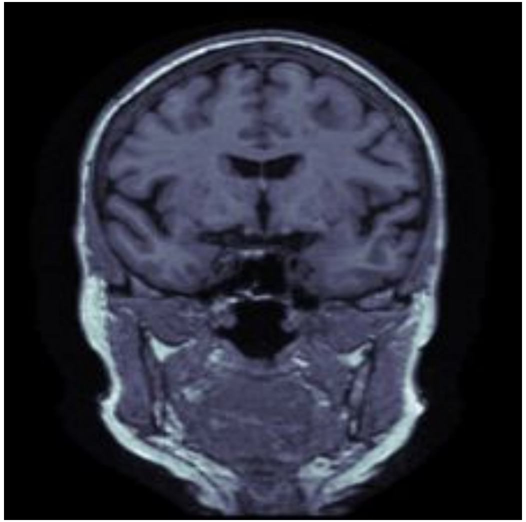
ADNI Data – 3 categories



Alzheimer's Disease (AD)



Cognitively Normal (CN)



Mild Cognitive Impairment (MCI)

Files types/how to load

- Two main types: DICOM and NiFTI
- Package called nibabel

```
import nibabel as nib
```

```
nifti = nib.load("137_S_0796.nii")
print(nifti.shape)
```

```
(256, 256, 180)
```

```
nii = nib.load(nii_path).get_fdata()[:,:,90] #sagital
nii = nib.load(nii_path).get_fdata()[:,128,:] #coronal
nii = nib.load(nii_path).get_fdata()[128,:,:] #axial
```

Data Used

- 1 image per person
- Split images BEFORE taking slices
- Report final answer as per PERSON, not per slice

	Train	Test
AD	79	20
CN	108	26
MCI	118	29

ADNI
2182

$\xrightarrow{\text{Visit 1}}$

Visit 1
485 MRIs

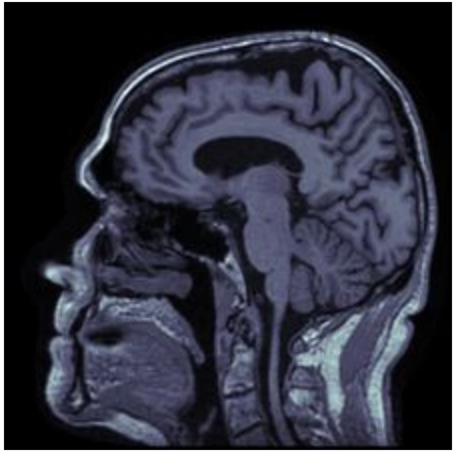
$\xrightarrow{\text{Duplicates removed}}$

382
MRIs

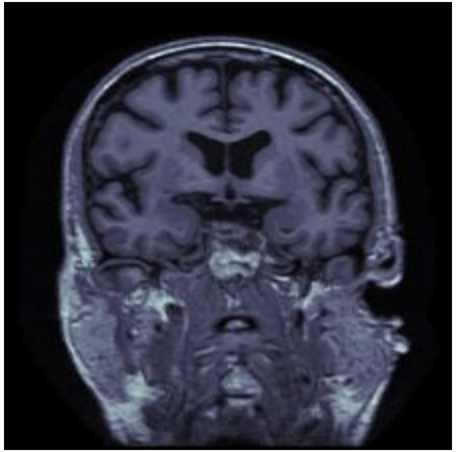
- 63 slices per MRI (21 in each orientation)

Differences from Kaggle Dataset

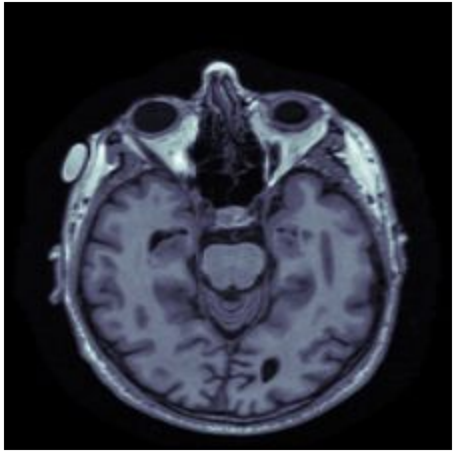
- Slices in different orientations (21 each, 63 total)
- Used raw images without “skull stripping” – Freesurfer



Sagittal View



Coronal View



Axial View

Didn’t really work: Overfitting

Model	Train Accuracy	Validation Accuracy	Test Accuracy
ResNet	0.9999	0.9982	0.4018
Xception	0.9999	0.9977	0.3900
VGG16	0.9741	0.9667	0.3993
Basic Keras	0.9987	0.9874	0.3353

Patient-wise accuracy

	AD	CN	MCI	Prediction	target
3402	9.829723e-15	0.755327	0.244673	1	2
3403	9.320969e-12	0.416101	0.583899	2	2
3404	1.108769e-11	0.484034	0.515966	2	2
3405	3.003239e-13	0.630967	0.369033	1	2
3406	3.435860e-17	0.971632	0.028368	1	2
...	...	...	...	...	...
3460	1.762819e-01	0.111977	0.711742	2	2
3461	1.018578e-01	0.123774	0.774368	2	2
3462	1.139304e-02	0.151275	0.837332	2	2
3463	1.746368e-02	0.146033	0.836503	2	2
3464	2.256335e-03	0.165082	0.832662	2	2

63 rows × 5 columns

Average of all 63 scores – to make final prediction:

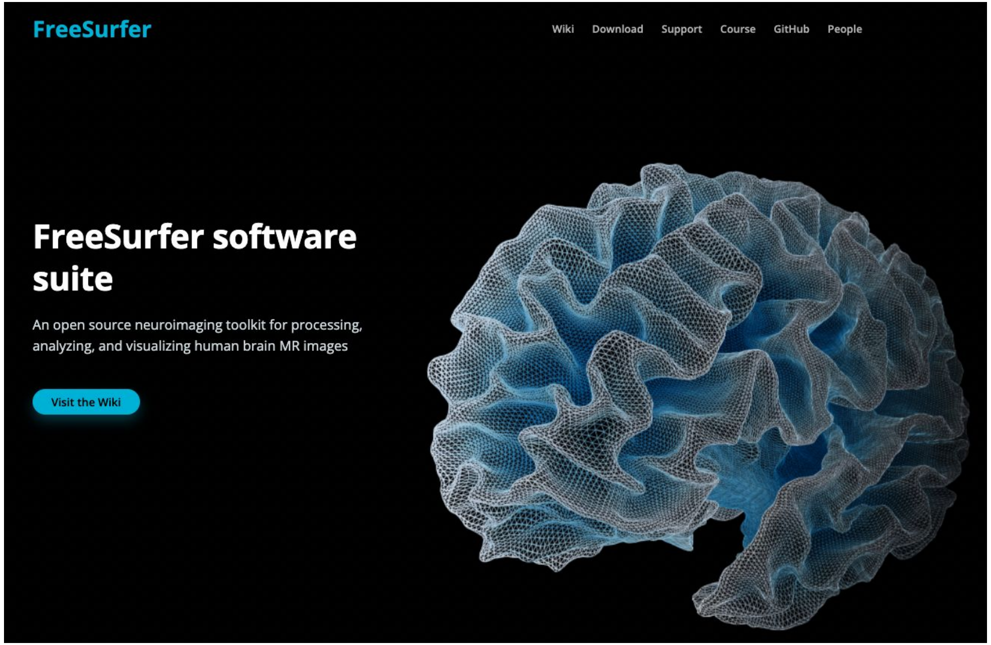
	Subject	AD	CD	MCI	Prediction	Target
0	55	0.06	0.4	0.54	MCI	MCI

Worked for this subject; still only gave me 40% accuracy

Skull stripping – necessary

- Brain disorder unlikely found in bones of skull
- Remove that info, create a better model

ADNI – Freesurfer Images



- Got similar results to raw images

Investigate why?

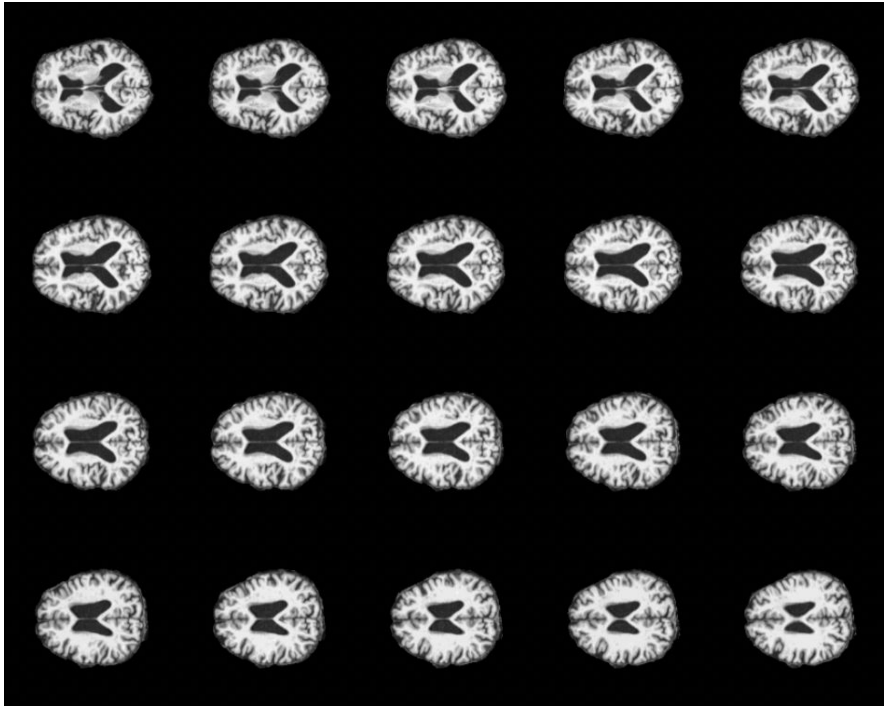
- Studied the images from Kaggle dataset – that I HAD been able to train, decided to take images visually similar and from same orientation

- Manually explored 10 MRI files from each class

- Discovered relevant slice varied from person to person

```
image = nib.load(nii_path).get_fdata()[:,:,160:180, 0]
```

Range of relevant slices changes



```
image =
nib.load(nii_path).get_fdata
()[:,:,160:180, 0]
```

However, for another patient, the range of slices was 135:155.

Only 21 slices – miss relevant info

What I learned:

- Appreciation for the complexity of MRI images
- Importance of domain knowledge for this task
- Challenges of supervised learning – people can move from one group to another

Thank you!

- Thank you to everyone from data science program – it's been a wonderful experience – learned so much