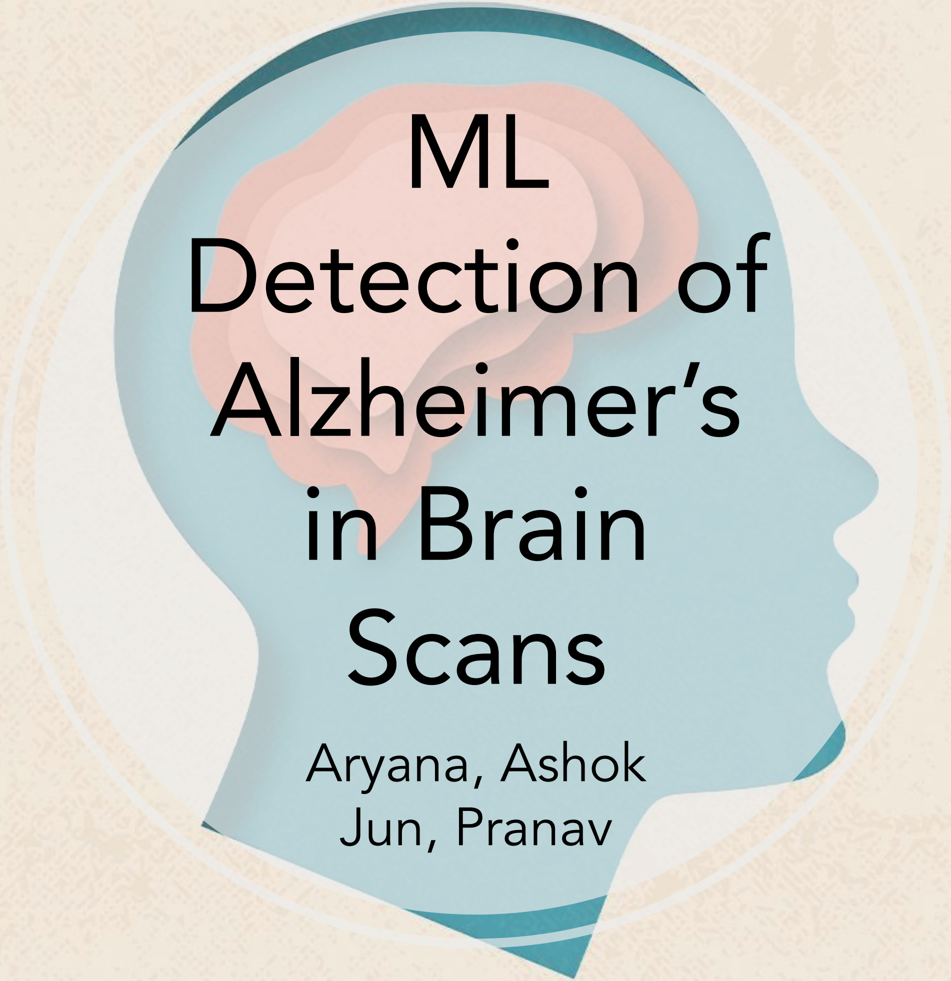


ML Detection of Alzheimer's in Brain Scans

Aryana, Ashok
Jun, Pranav

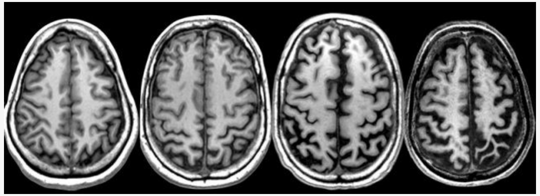
AGENDA

One	Background & Approach
Two	Datasets
Three	Preprocessing
Four	Data Modeling
Five	Conclusion

Background & Approach

Background

- Alzheimer’s...
    - 1/9 elderly, 6 million Americans
    - impaired memory
    - neurodegenerative
- ML-powered detection...
    - prevention, planning, research



Approach

- 21 different models that prioritize **accuracy**, *90% threshold*
    - correctly classify as many cases as possible
    - paying attention to FNs via confusion matrices

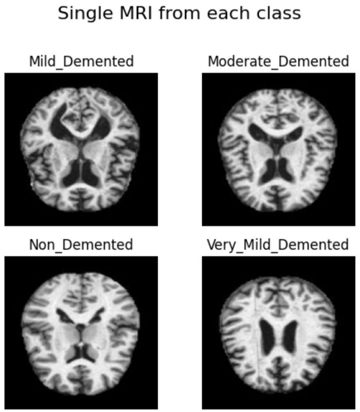
2 Datasets Overview

Dataset 1

- **size** = 6.4k
- **image dimensions** = 128x128
- **preprocessed** = True
- **number of unique brains** = unknown
    - every image treated independently

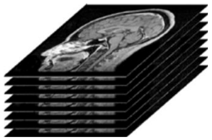
Dataset 2

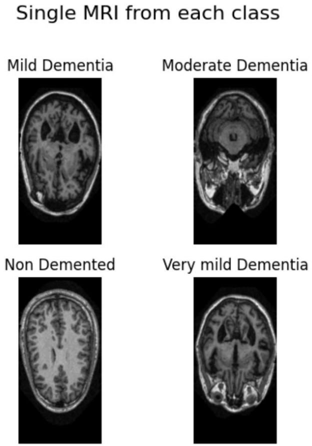
- **size** = 86k
- **image dimensions** = 244x488
- **preprocessed** = False
- **number of unique brains** = 1.5k
    - every 61 images stacked → 3D brain



Shared Characteristics

- **axial** = True
- **classes** = non-demented, very mildly demented, mildly demented, moderately demented

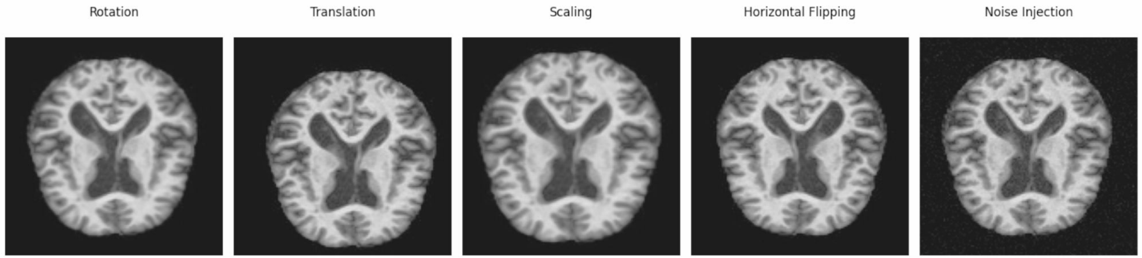




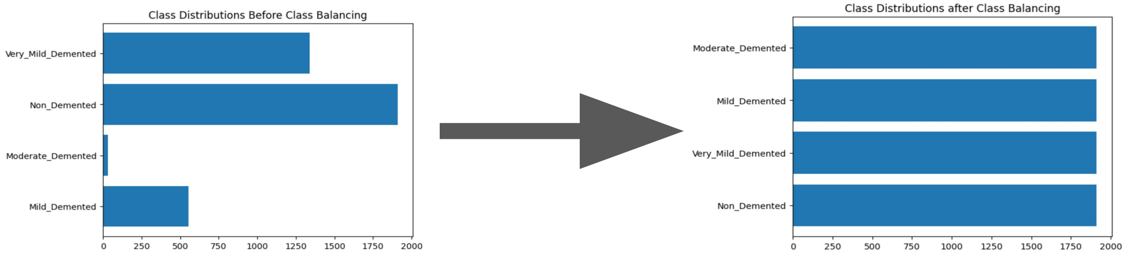
Dataset 1 Preprocessing

60/20/20 Split

Augmentation



Balancing



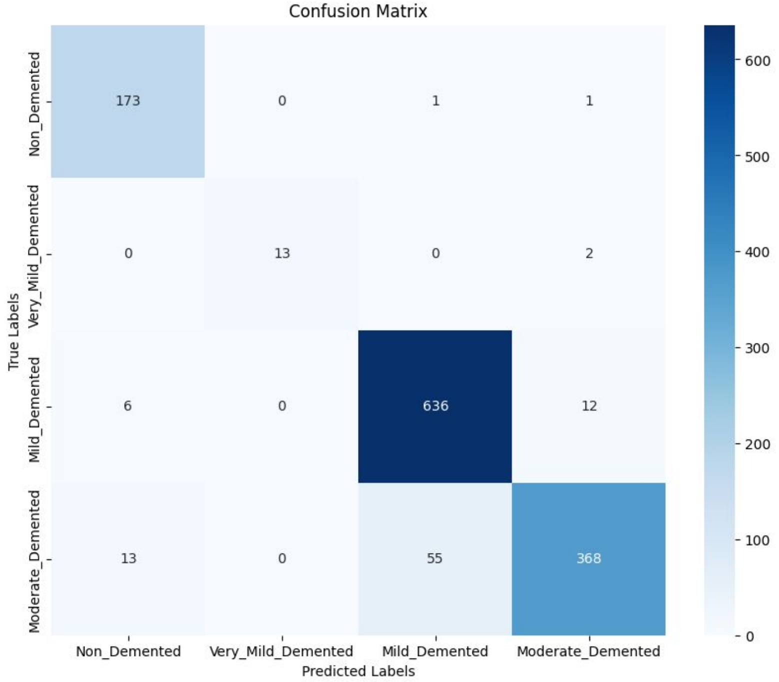
Modeling Dataset 1

Model	Baseline results	Fine-tuned results
Logistic Regression	Train Accuracy: 0.4566 Val. Accuracy: 0.4906 Test Accuracy: 0.5063	Train Accuracy: 0.8515 Val. Accuracy: 0.8555 Test Accuracy: 0.8609
Decision Tree	Train Accuracy: 1.0000 Val. Accuracy: 0.6898 Test Accuracy: 0.6695	
Random Forest	Train Accuracy: 0.9946 Val. Accuracy: 0.7852 Test Accuracy: 0.7633	Train Accuracy: 1.0000 Val. Accuracy: 0.8680 Test Accuracy: 0.8633
CNN	Train Accuracy: 0.8360 Val. Accuracy: 0.7141 Test Accuracy: 0.7203	Train Accuracy: 0.8371 | 0.9297 Val. Accuracy: 0.7461 | 0.9258 Test Accuracy: 0.7492 | 0.9297
Transfer Learning	Train Accuracy: 0.6484 | 0.9850 Val. Accuracy: 0.6313 | 0.8539 Test Accuracy: 0.6469 | 0.8562	Train Accuracy: 0.9953 Val. Accuracy: 0.9008 Test Accuracy: 0.9031

Dataset 1 - CNN Models

- Addition of a third 2D convolutional layer with 64 filters **slightly improved test accuracy from baseline (72% to 74%)** in the first improvement
- Changing third 2D convolutional layer to 128 filters and increasing dense layer units to 256 **improved test accuracy to 93%**
- Final model confusion matrix shown on right:

Advanced CNN Model #2



Dataset 1 - Transfer Learning Models

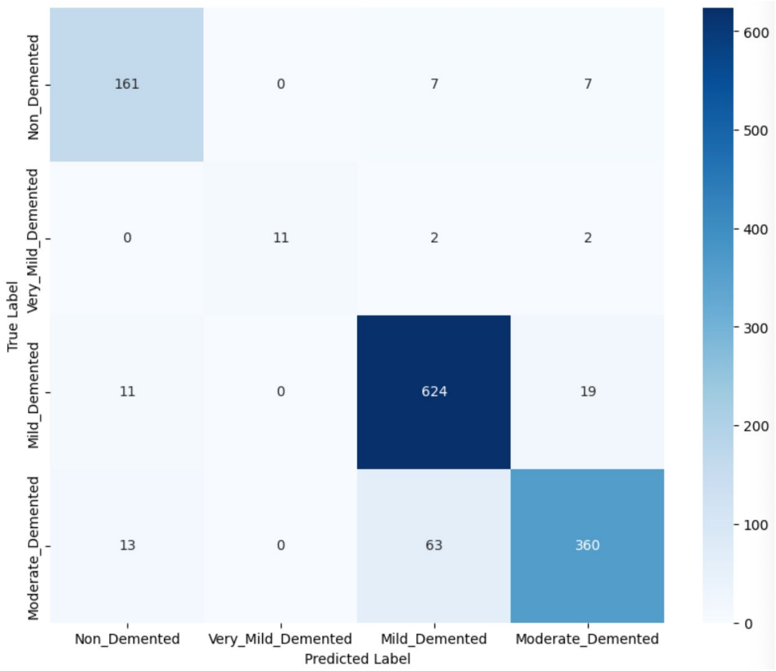
EfficientNetB0 Baseline Model

- Weights pre-trained on ImageNet
- Fine-tuned last 17 layers (237 total layers)
- Accuracies were around 64% across split data

VGG16 Models

- Both: fine-tuned last 8 layers (16 total layers)
- Improved: 1/2 dense layer units
- Test accuracy improved by 5% to 90%, with less overfitting

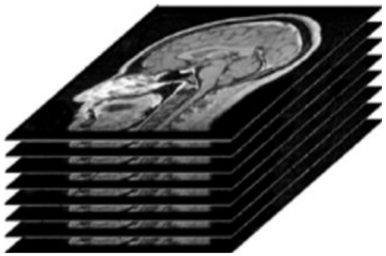
VGG16 Improved Model Results



Dataset 2 Preprocessing

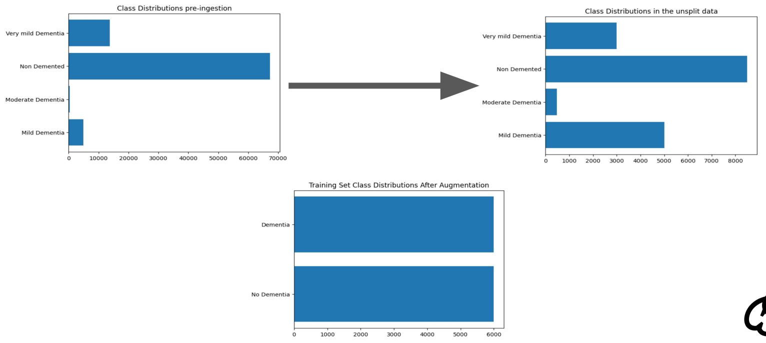
3D Pipeline

- **stacked** 61 axial scans during ingestion
- **transformed** data
- class **merged** → binary, multiclass
- 80/20 **split**
- **augmented** (some) training volumes



2D Pipeline

- **undersampled** during ingestion
- **transformed** data
- class **merged** → binary
- 60/20/20 **split**
- **augmented** training images



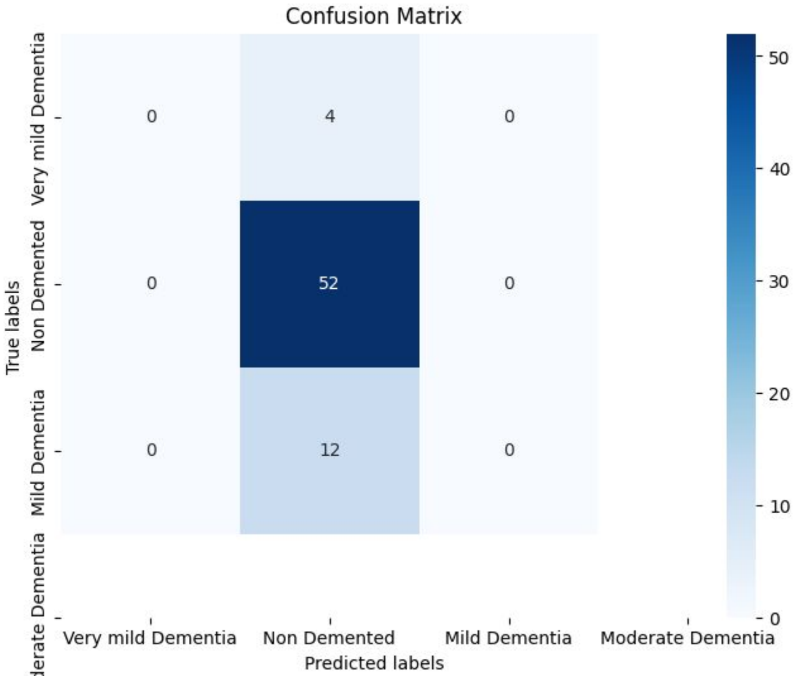
Modeling Dataset 2 - 3D Pipeline

Model	Results	Takeaways
CNN	Test Accuracy: 0.7794 Test Loss: 0.7982	- Skewed predictions based on confusion matrix - Missing Class
Binary CNN	Test Accuracy: 0.7647 Test Loss: 0.5505	- Similar performance to multi-class, likely due to class imbalance
CNN with Slice Augmentation	Test Accuracy: 0.4450 Validation Accuracy: 0.4699 Test Loss: 2.3780 Validation Loss: 1.3218	- Performs worse than volume-augmentation - Lower results than no augmentation are likely due to lower epochs
CNN with Volume Augmentation	Test Accuracy: 0.5532 Validation Accuracy: 0.6933 Test Loss: 1.7072 Validation Loss: 1.0582	- Performs better than slice-by-slice - Spatial relations are important - Room for more training

Dataset 2, 3D Pipeline - Simple CNN Model

Simple CNN Model

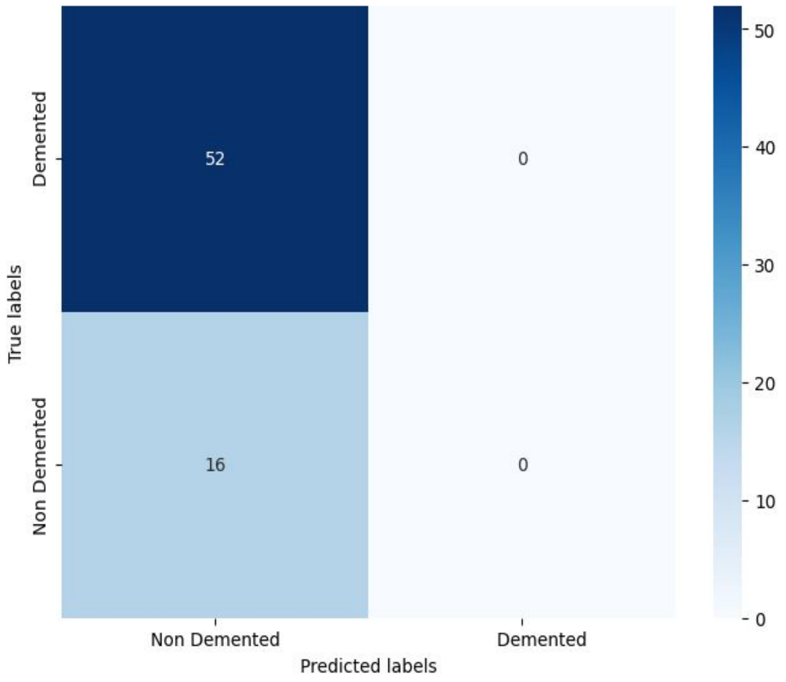
- Data-points heavily reduced
- Large class imbalance
- Memory issues
  - Generators
  - .dat files
  - Compression
- Despite this...
  - Test Accuracy: 77.94%
  - Test Loss: 0.7982



Dataset 2, 3D Pipeline - Binary CNN Model

Binary CNN Model

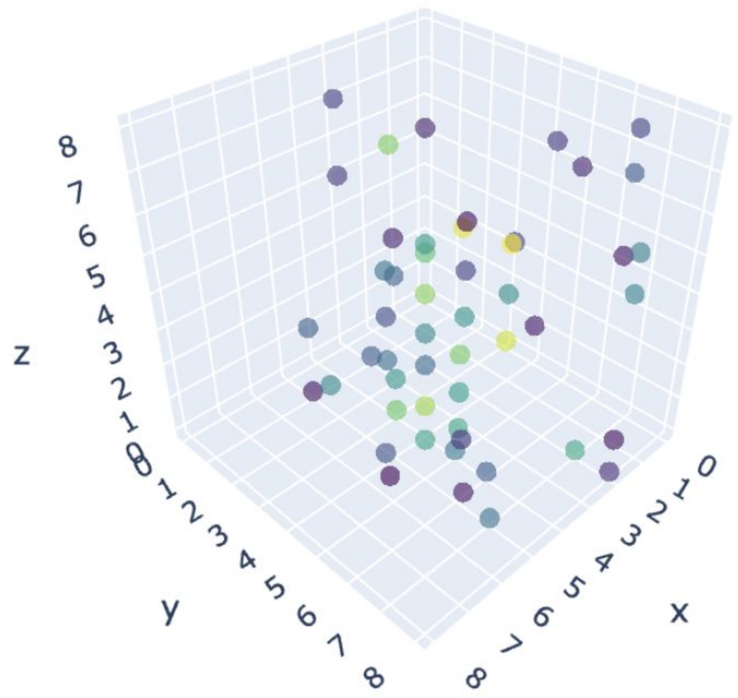
- Same preprocessing steps and model architecture
- Similar performance
  - Lack of sensitivity within predictions, likely due to class imbalance



Dataset 2, 3D Pipeline - CNN, Augmentation 1

Slice Augmentation CNN

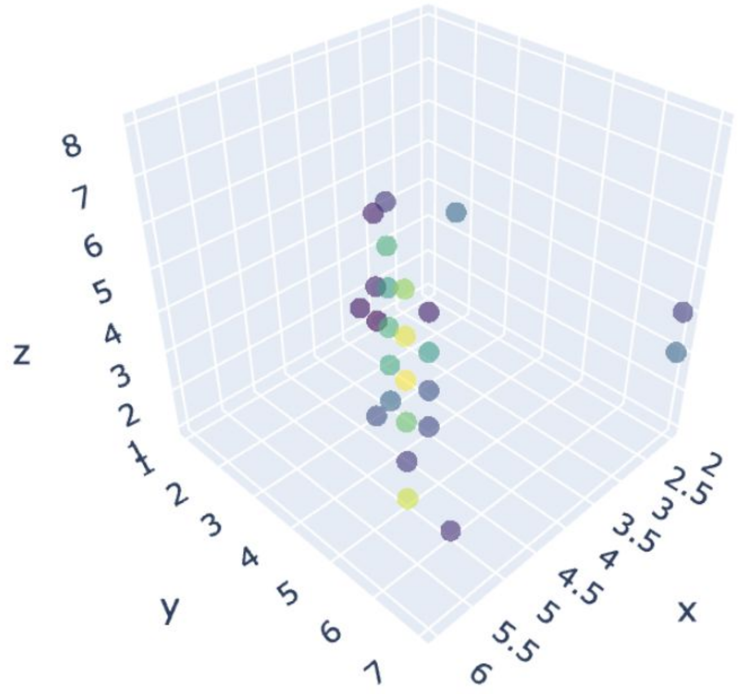
- Augmenting Slice-by Slice
- Initial Theory:
    - Model resiliency
        - Resistant to noisy and missing data
    - Lose value of spatial relationships
- Visualization: too much scatter



Dataset 2, 3D Pipeline - CNN, Augmentation 2

Volume Augmentation CNN

- Augmenting by Volume
- Initial Theory:
  - Introducing unrealistic artifacts within the volume
  - Maintain spatial relationships



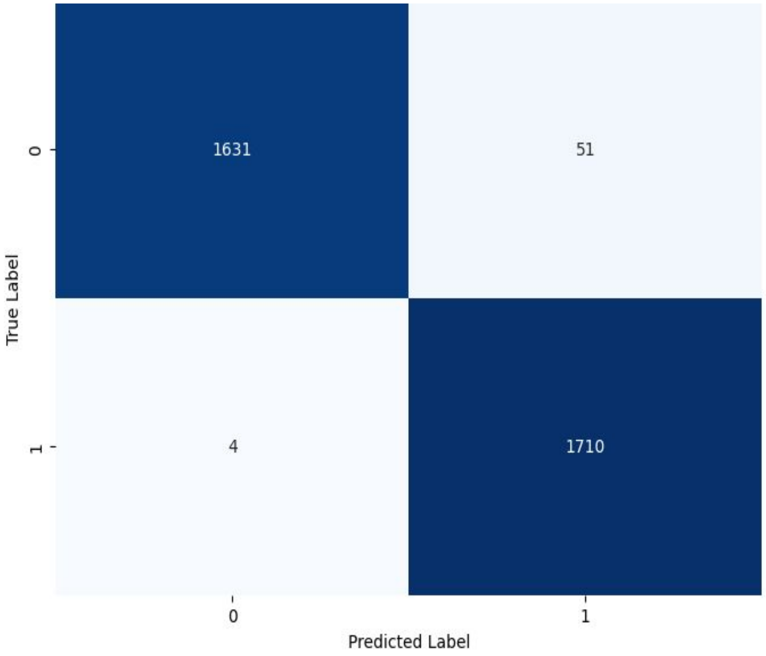
Modeling Dataset 2 - 2D Pipeline

Model	Baseline results	Hyperparameter tuning
Logistic Regression	Train Accuracy: 0.6823 Val. Accuracy: 0.8157 Test Accuracy: 0.8193	Train Accuracy: 0.9312 Val. Accuracy: 0.9049 Test Accuracy: 0.9112
CNN	Train Accuracy: 0.9286 Val. Accuracy: 0.9423 Test Accuracy: 0.9464	Train Accuracy: 0.9903 Val. Accuracy: 0.9838 Test Accuracy: 0.9862
Transfer Learning	Train Accuracy: 0.8205 Val. Accuracy: 0.8716 Test Accuracy: 0.8790	Train Accuracy: 0.9882 Val. Accuracy: 0.9691 Test Accuracy: 0.9714

Dataset 2, 2D Pipeline - CNN Models

- Doubling the number of filters in the final 2D convolutional layer and the dense layer units from the baseline **increased test accuracy from 95% to 99%**
- Minimal disparity between train and test accuracies **suggests robust generalizability**
- Hyperparameter tuned **CNN was the highest performing model**

CNN Tuned Model Results



Dataset 2, 2D Pipeline - Transfer Learning Models

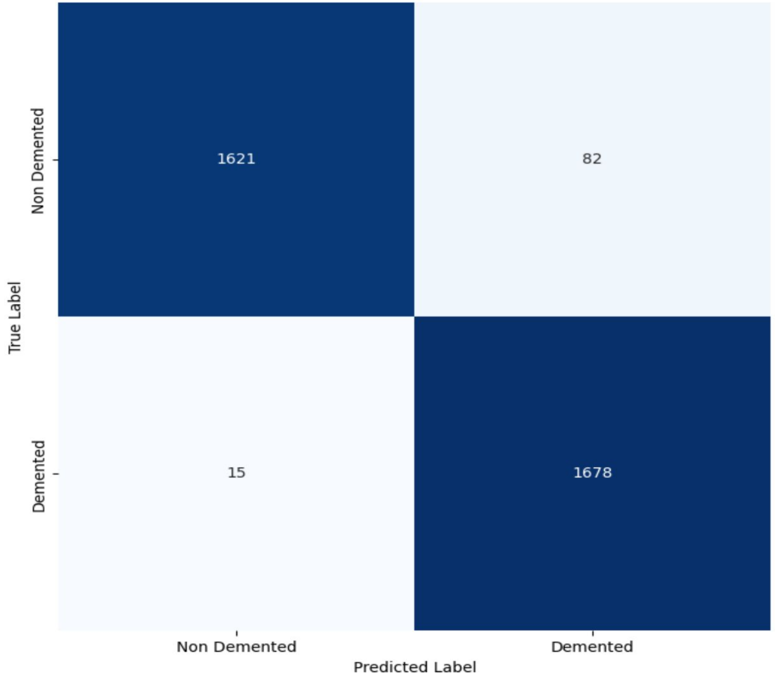
EfficientNetB0 Model

- Weights pre-trained on ImageNet
- Fine-tuned last 17 layers (237 total layers)
- Test accuracy was 88%, with no overfit

VGG16 Model

- Fine-tuned last 8 layers (16 total layers)
- Average accuracy was 97%, with no overfit

VGG16 Tuned Model Results



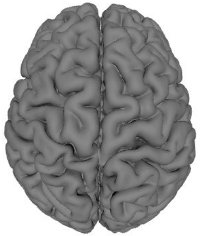
Conclusion

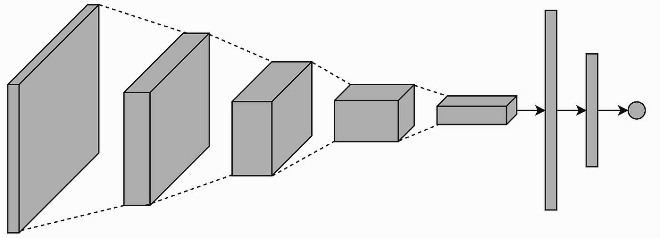
Highlights

- **CNNs** performed best (test accuracy, FNR, generalization)
  - Dataset 1 – Model 8 : **92.97%**
  - Dataset 2, 2D – Model 4: **98.62%**
- **Transfer** models were a close second
  - Dataset 1 – Model 11: **90.86%**
  - Dataset 2, 2D – Model 6: **97.14%**

Takeaways

- **Breadth** and **depth** in modeling methodology
  - **Breadth**: diverse models
  - **Depth**: Deep learning focus
- **Future** work: CNNs, Transfer, 3D brain volumes
- **Fairness:** diagnostic automation





THANK YOU

Questions?

Dataset 1 - Traditional ML Models

Logistic Regression

- Baseline to improved multiclass logistic regression showed a **36% test accuracy increase (50% to 86%)**
- Biggest impact came from dropping **learning late from 0.01 to 0.0001** and increasing training **epochs from 3 to 20**
- **Little overfitting** across both models

Decision Tree

- Explored decision trees solely to gain insights for potential **applications in random forest** models
- **Highly overfit to training data** with a perfect training accuracy but **66% validation and testing accuracies**
- Generally recognized for **underperforming on image data**

Random Forest

- Baseline to advanced demonstrated an ~**11% increase in test accuracy (from 76% to 88%)**
- Using **grid search**, we achieved our optimal model parameters
- Both baseline and advanced models **indicate some overfitting** with training accuracies of ~100%

Dataset 2, 2D Pipeline - LR Model

Model Overview

- Increasing the number of epochs (1 to 20) and changing the Adam optimizer to Stochastic Gradient Descent **increased test accuracy 81% to 91%**
- Large increase from simple hyperparameter tuning hints at the **potential of models with complex layered architectures** in learning hierarchical pixel relationships