Evaluating Alzheimer's Detection Through Use of Convolutional Neural Networks, Transfer Learning and Binary Classification.

Dillon Fleharty

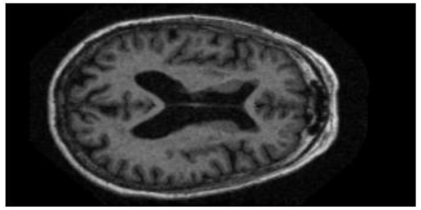
Classifying Alzheimer's Disease

- Introduction
- Exploratory Data Analysis
- Classifying Multiple Classes with Convolutional Neural Network
- Classifying Multiple Classes with Pre-Trained Model
- Binary Classification with Convolutional Neural Network
- Results

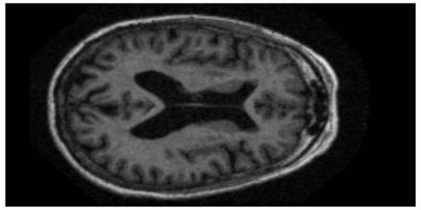
Introduction

- Alzheimer's disease (AD) effects more than 55 million individuals world-wide.
- No current cure, early detection is critical
- This research aims to enhance early diagnostics by using Convolutional Neural Networks (CNNs) analyzing MRI scans
- Dataset used is from Open Access Series of Imaging (OASIS-1)
  - 80,000 cross-sectional MRI scans from 416 subjects aged 18-96, including 100 elderly subjects diagnosed with Alzheimer's
- Three CNNs:
  - Model 1: Classifies subjects into four categories: Non-Demented, Very Mild, Mild, and Moderate
  - Model 2: Employs transfer learning techniques using a pre-trained 'EfficientNetB0' base, supplemented with a custom layer
  - Model 3: Using binary classification for either Non-Demented or Demented

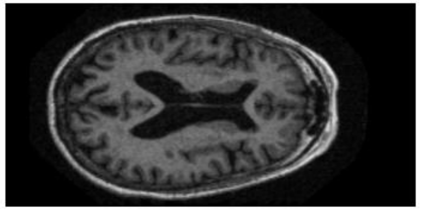
Non 1



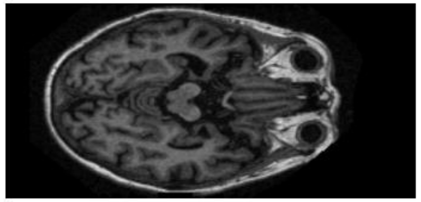
Non 2



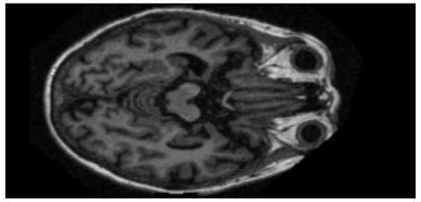
Non 3



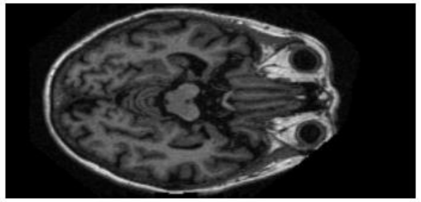
Very 1



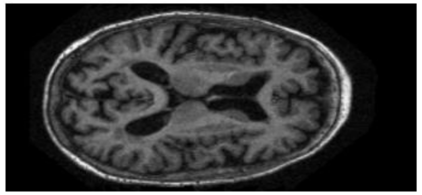
Very 2



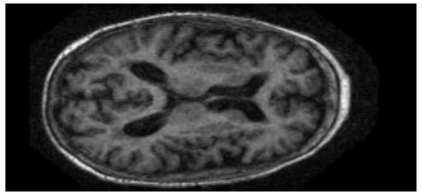
Very 3



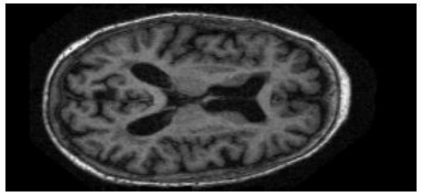
Mild 1



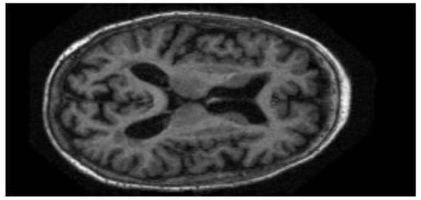
Mild 2



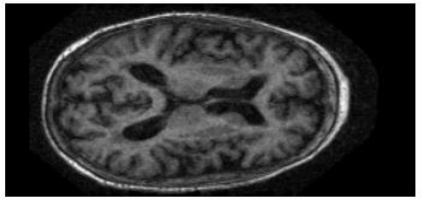
Mild 3



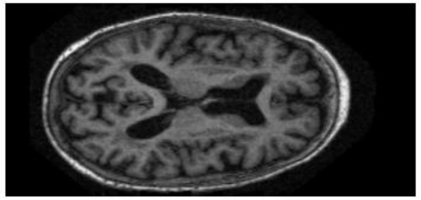
Mild 1



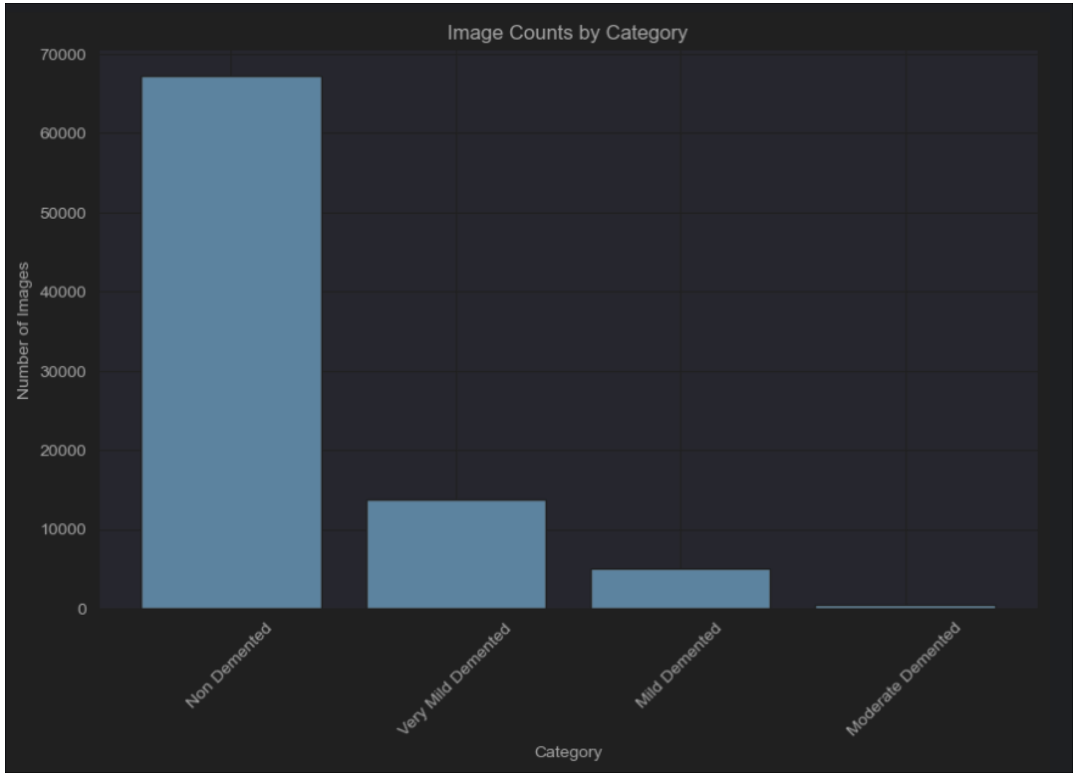
Mild 2



Mild 3

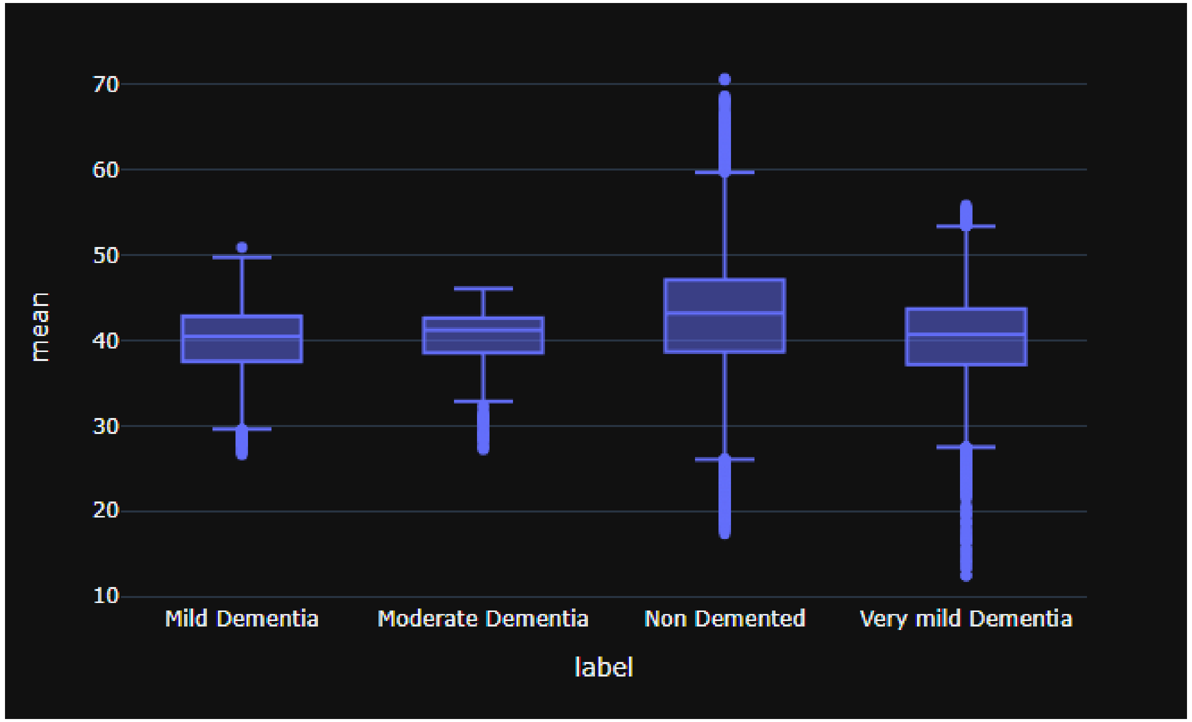


Dataset and Statistics

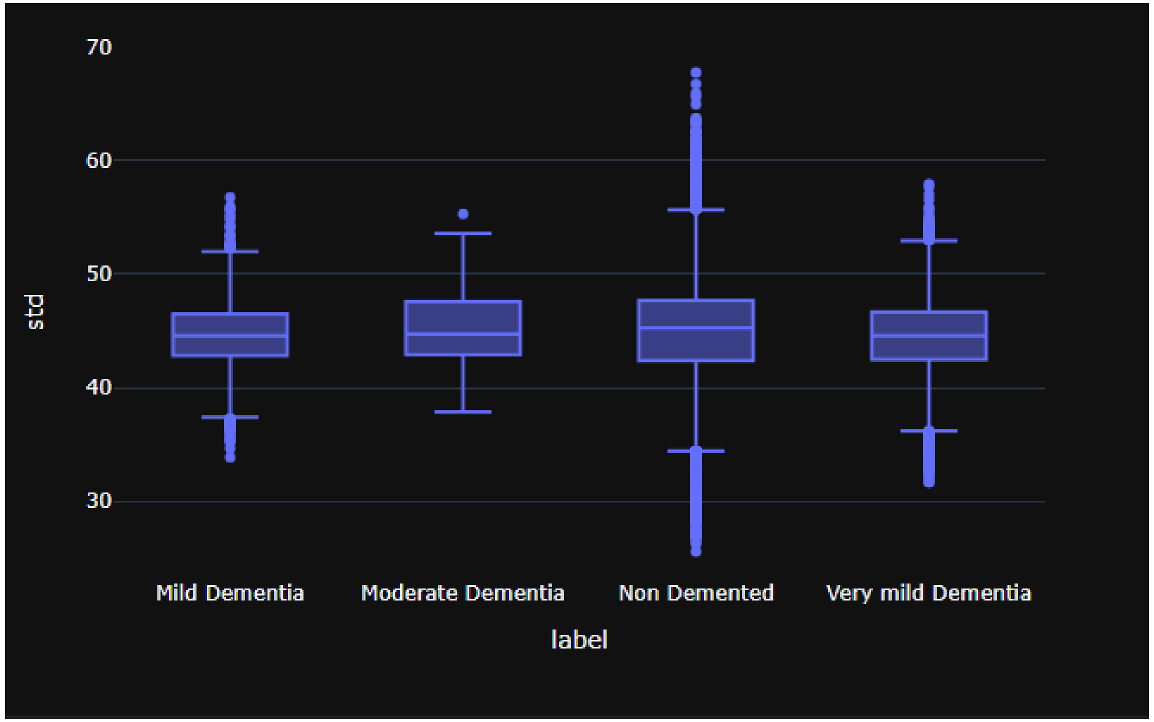


- **Non-Demented**: 67,222
- **Very Mild**: 13,725
- **Mild**: 5002
- **Moderate**: 488

Image Stats

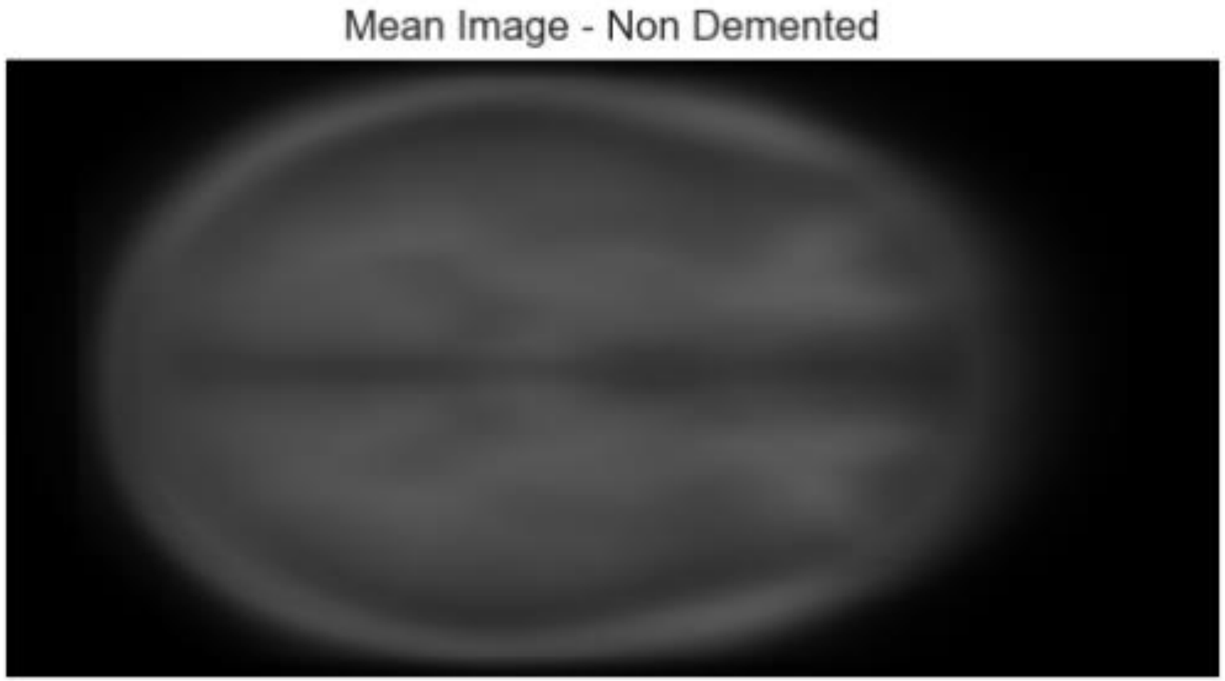


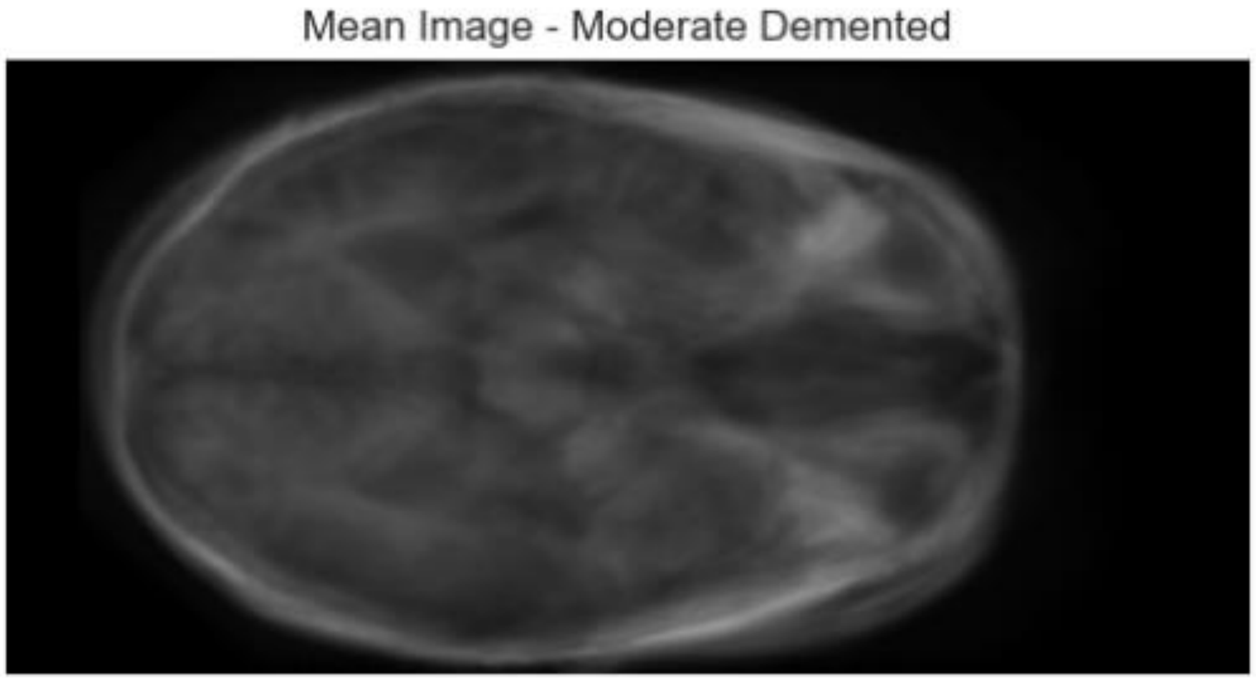
70
60
50
40
30
20
10
mean
Mild Dementia
Moderate Dementia
Non Demented
Very mild Dementia
label



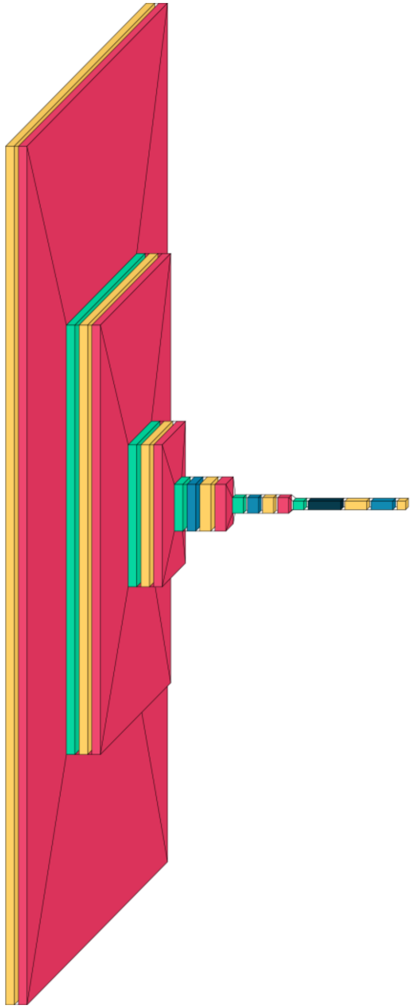
70
60
50
40
30
std
Mild Dementia
Moderate Dementia
Non Demented
Very mild Dementia
label

Display of Mean Pixel Values for Non-Demented / Demented





Model 1 Architecture



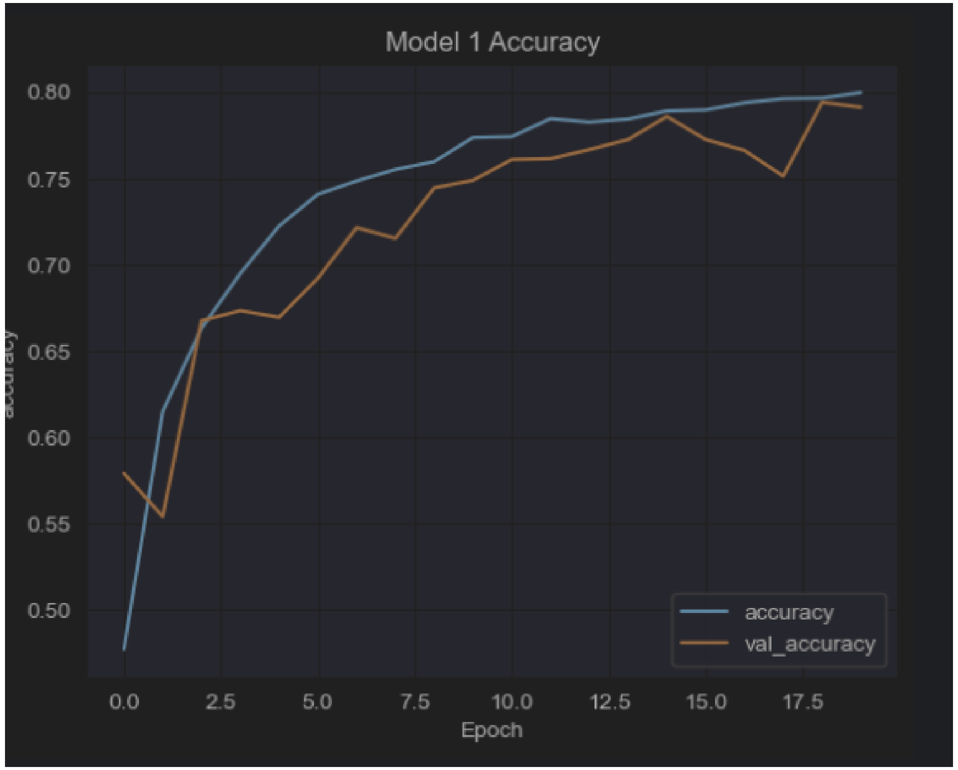
Architecture			
Input Layer	**Input Shape: 248x496x3**	**Regularization**	**Notes**
Conv2D	32 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 2x2		
Conv2D	64 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same, Strides: 1x1
BatchNormalization			
MaxPooling2D	Pool Size: 3x3		
Conv2D	128 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 3x3		
Dropout	Rate: 0.5		
Conv2D	256 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 2x2		
Dropout	Rate: 0.5		
Conv2D	256 filters, Kernel Size: 5x2, Activation: ReLU	L2(0.001)	Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 2x2		
Flatten			
Dense	512 units, Activation: ReLU	L2(0.001)	
Dropout	Rate: 0.5		
Output Layer	4 units, Activation: Softmax		

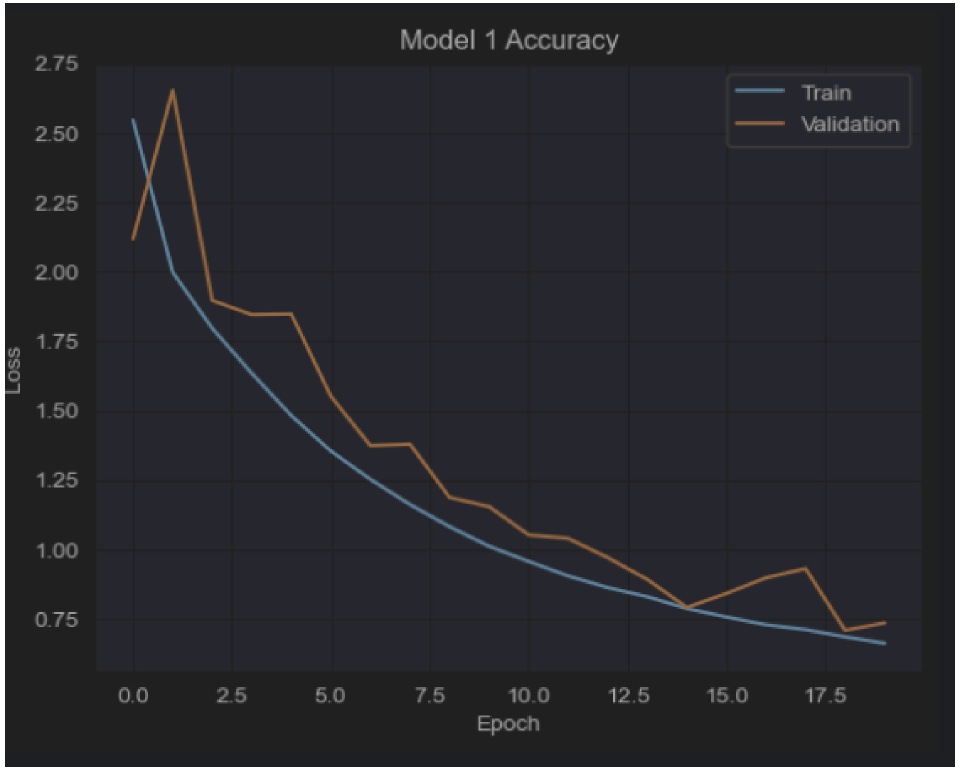
Model 1

- **Model Architecture**: Sequential CNN
- **Training Images:**
  - Non-Demented: 4000
  - Very Mild: 2000
  - Mild: 4001
  - Moderate: 390
- **Test Images:**
  - Non-Demented: 1000
  - Very Mild: 248
  - Mild: 496
  - Moderate: 3
- **Optimization Techniques**:
  - Adam optimizer with .0001 learning rate
  - Learning rate scheduler with linear decay
  - L2 Regularization
  - Kernal size variations
- **Strategy**:
  - Data augmentation on MRI scans to prevent overfitting,
  - one-hot encoding for label categorization
  - 20% validation split

Model 1 Training and Test Results

- **Test Accuracy: 79.42%**
- **Precision: 0.6569**
- **Recall: 0.7942**
- **F1 Score: 0.7163**

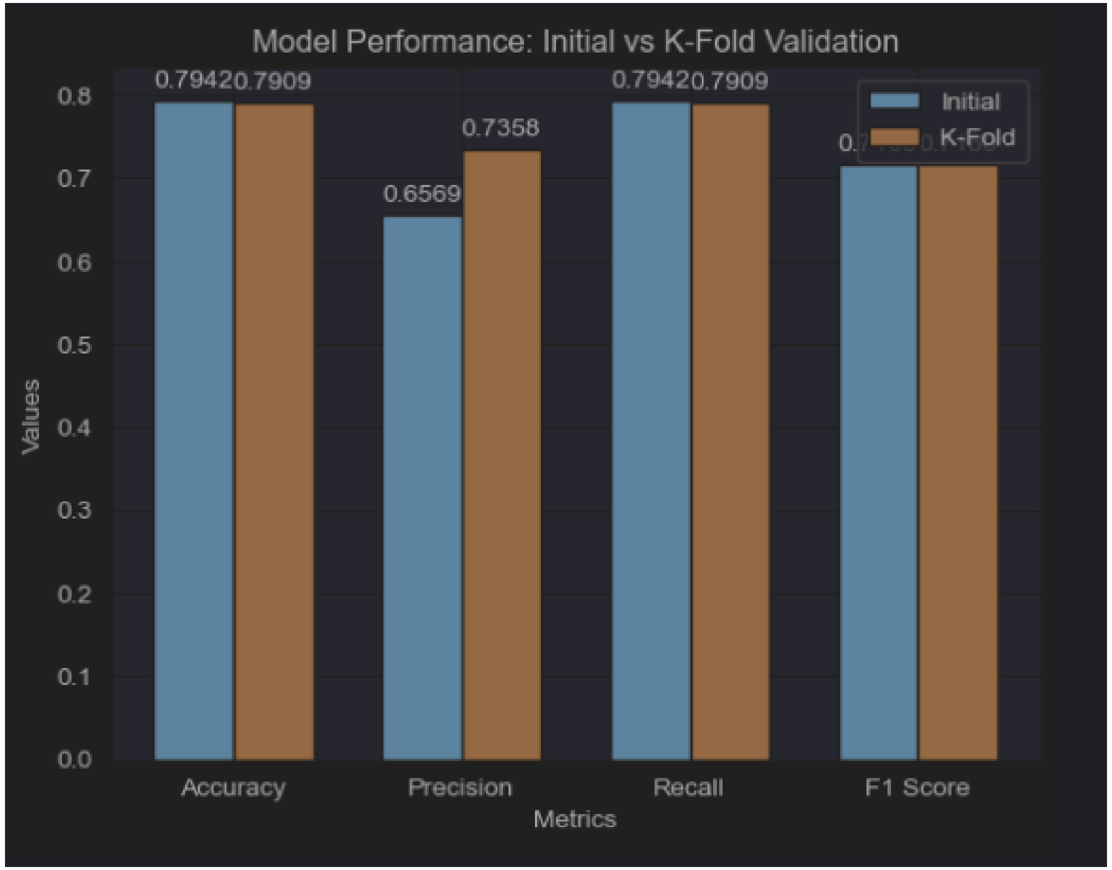




Model 1 K-Fold Cross Validation Results

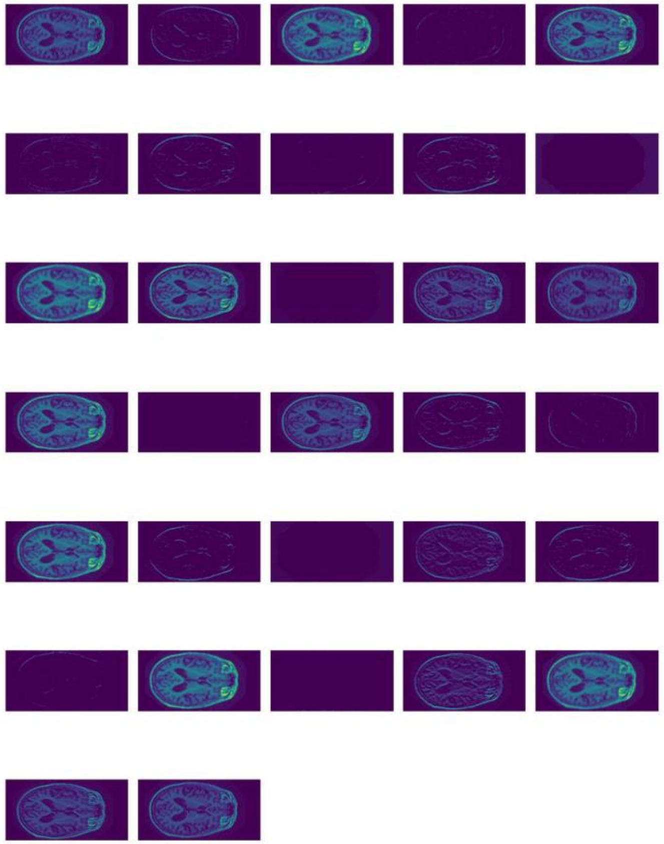
K-fold validation is a statistical technique that divides a dataset into multiple subsets, using each in turn for testing a model trained on the remaining data to ensure a robust and generalized evaluation of its performance.

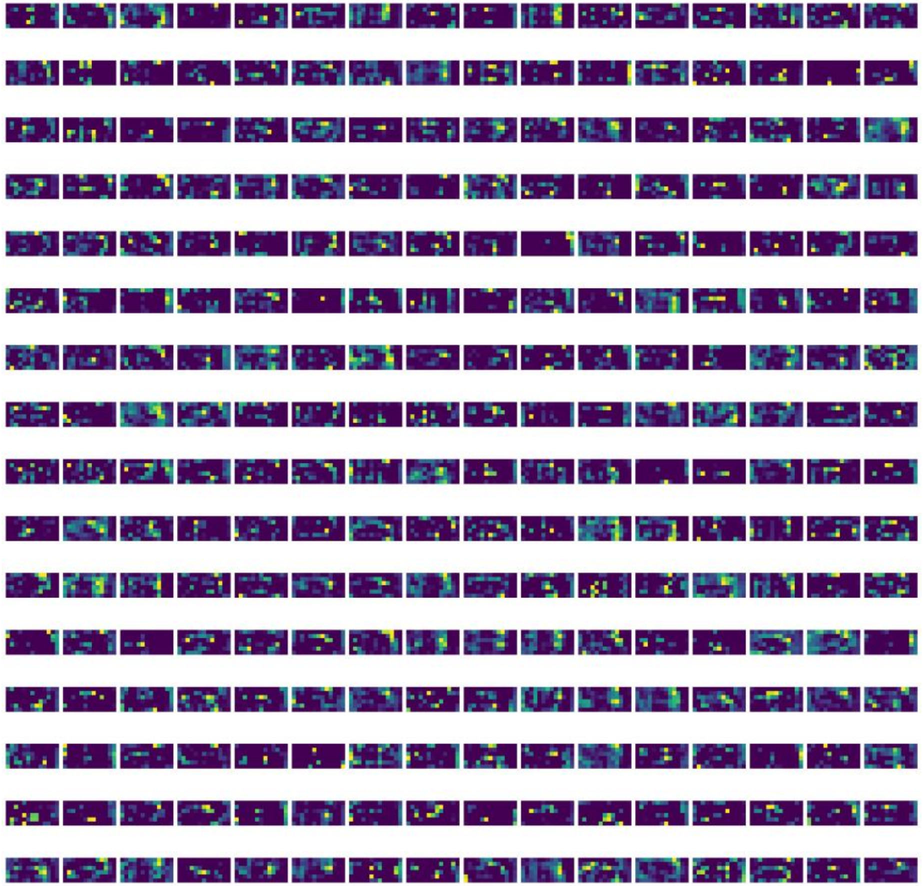
- **Average Test Accuracy: 0.7660**
- **Average Precision: 0.7238**
- **Average Recall: 0.7660**
- **Average F1 Score: 0.7028**



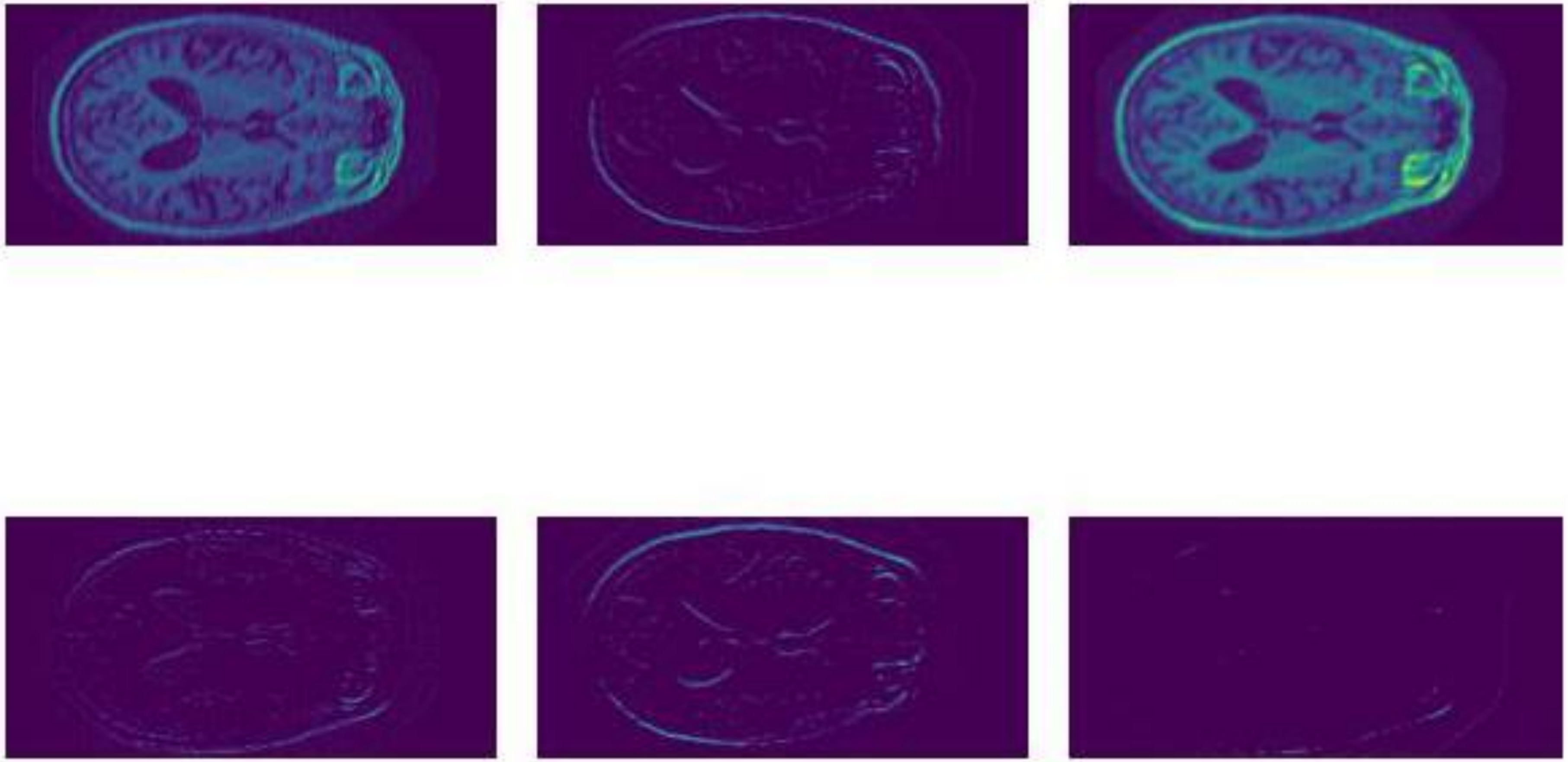
Model 1 Feature Maps

- **Highlight Key Visual Cues** to reveal textures, edges, pattens the network detect at different layers.
- **Understanding Model Focus** by examining the feature maps it helps to understand the models decision-making process and to ensure it aligns as expected
- **Progressive Complexity** in feature maps increases with each layer, moving from simple edges in early layers to more abstract features in deeper layers, indicating the hierarchical nature of feature extraction in CNNs.

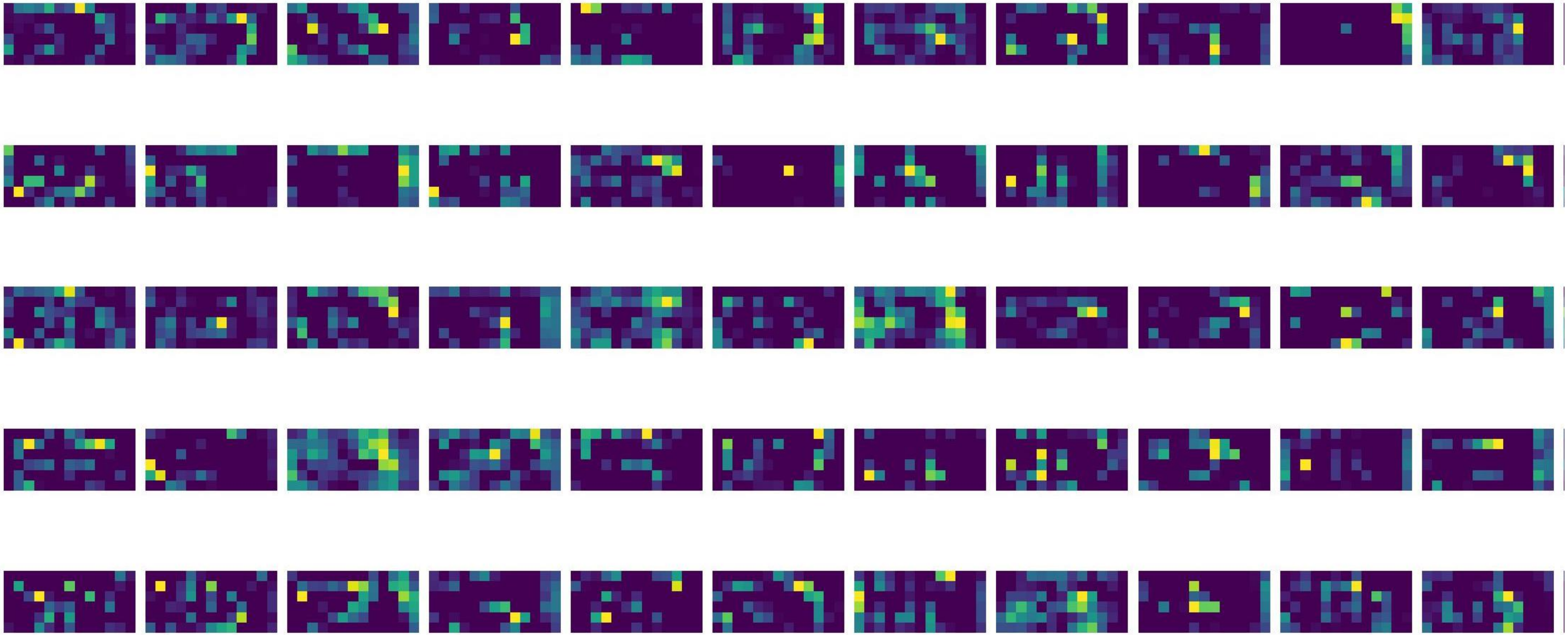




Model 1 Feature Maps

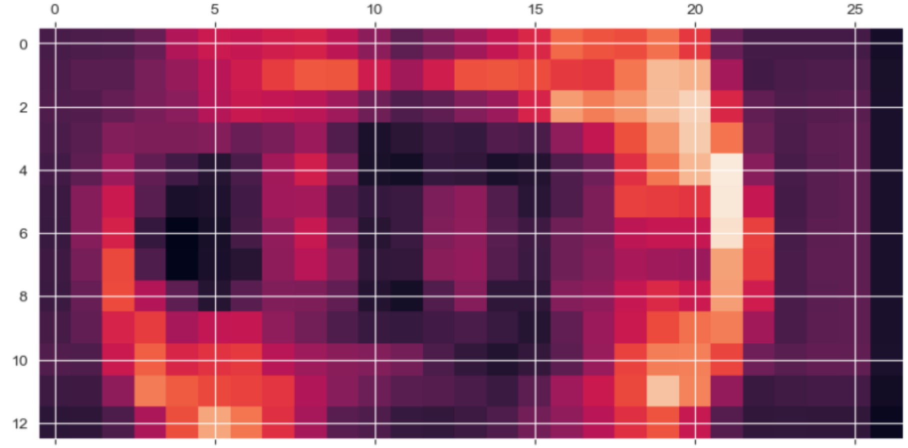


Model 1 Feature Maps

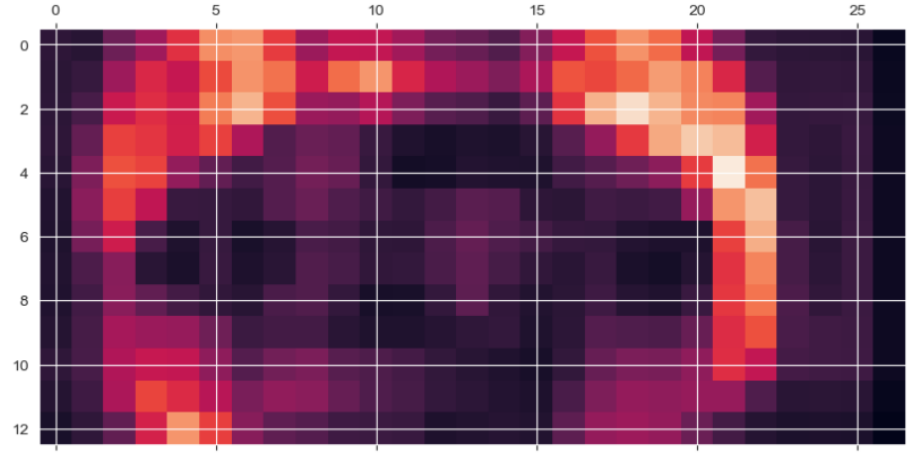


Model 1 Heat Map

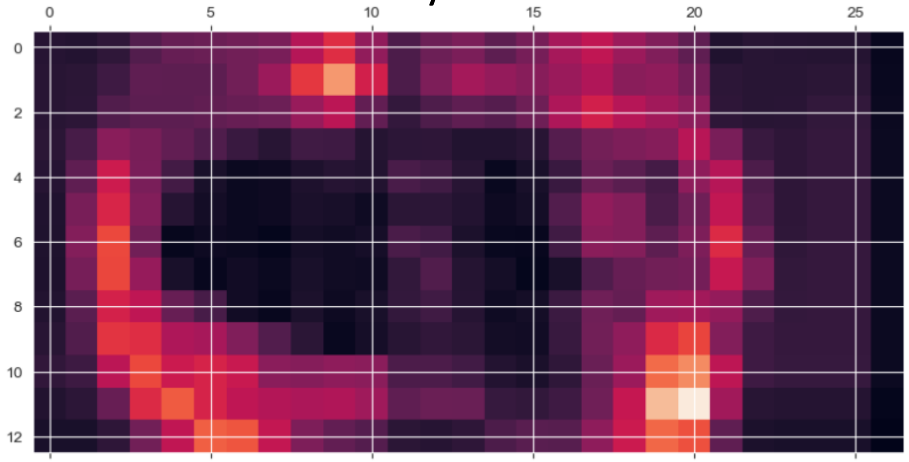
Non-Demented



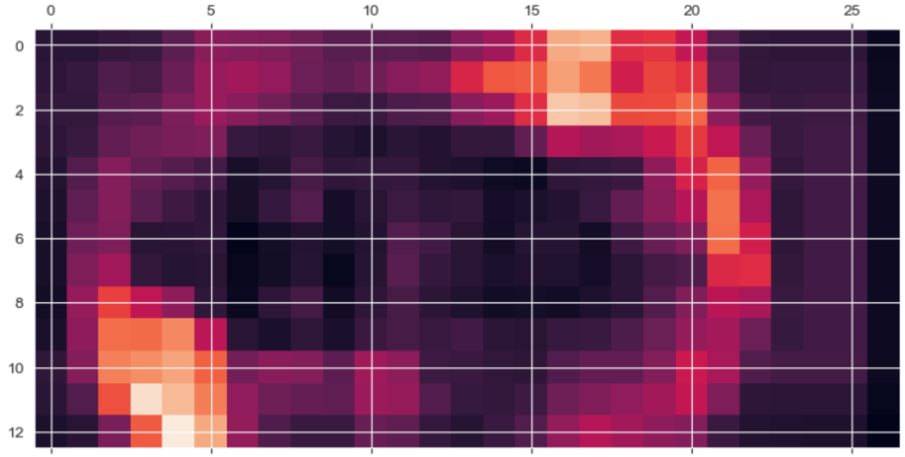
Mild



Very Mild

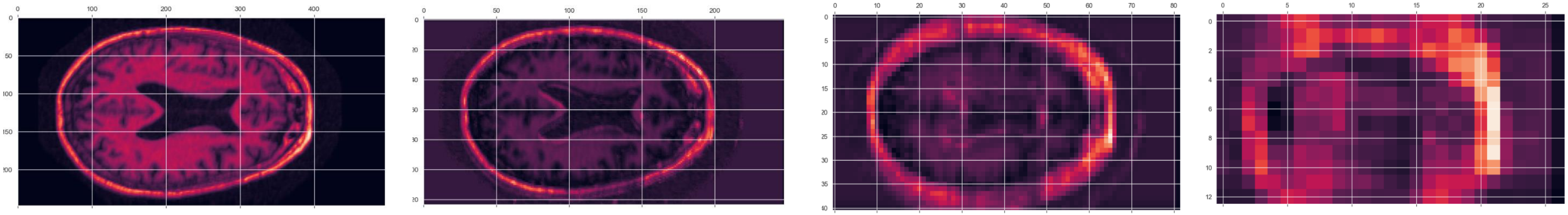


Moderate

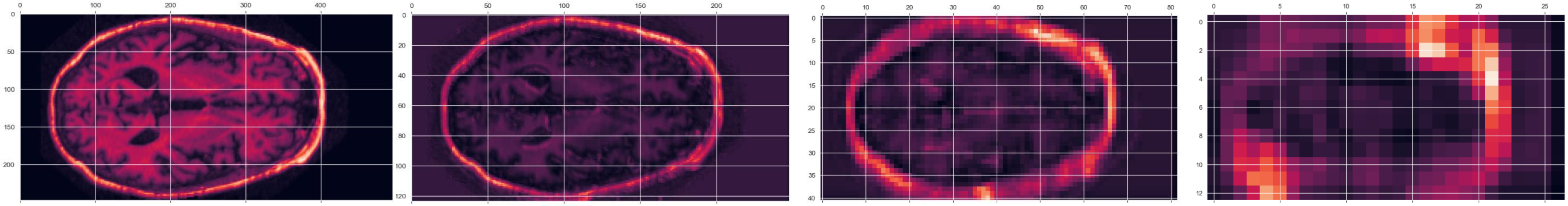


Model 1 Heatmap Layers 1 to 4

Non-Demented



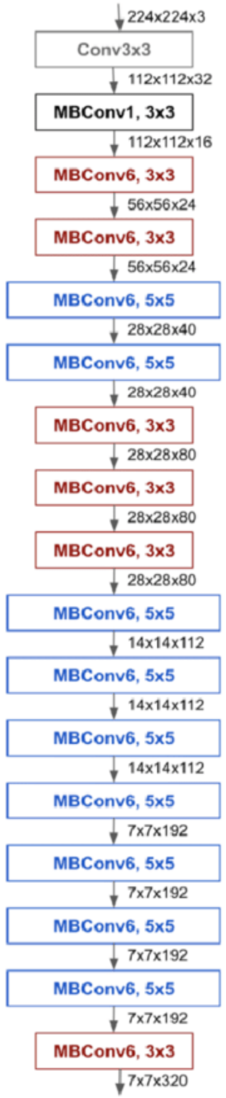
Moderate Demented



Model 2 EfficientNetB0 Architecture

- **Developed by:** Minxing Tan and Quoc V. Le at Google Research
- **Scalable Architecture**: EfficientNetB0 is a convolutional neural network that uses a compound scaling method to uniformly scale the depth, width, and resolution of the network.
- **Mobile Inverted Bottleneck Convolution (MBConv)**: It employs MBConv blocks, an efficient variant of convolutional layers that use depthwise separable convolutions to reduce the number of parameters and computational cost.
- **ImageNet Dataset**: Original model was trained on the ImageNet dataset (+14 million images with 1000 different classes)
- **Dataset Subset**: 1.2 million images.

The Mobile Inverted Bottleneck Convolution (MBConv) utilizes inverted residuals and linear bottlenecks, combining depthwise separable convolutions with shortcut connections

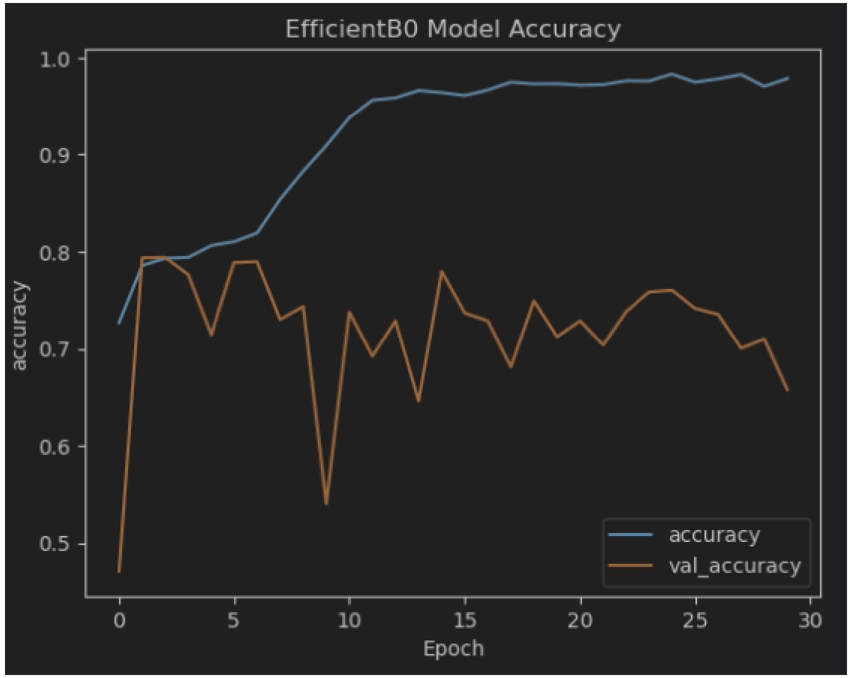


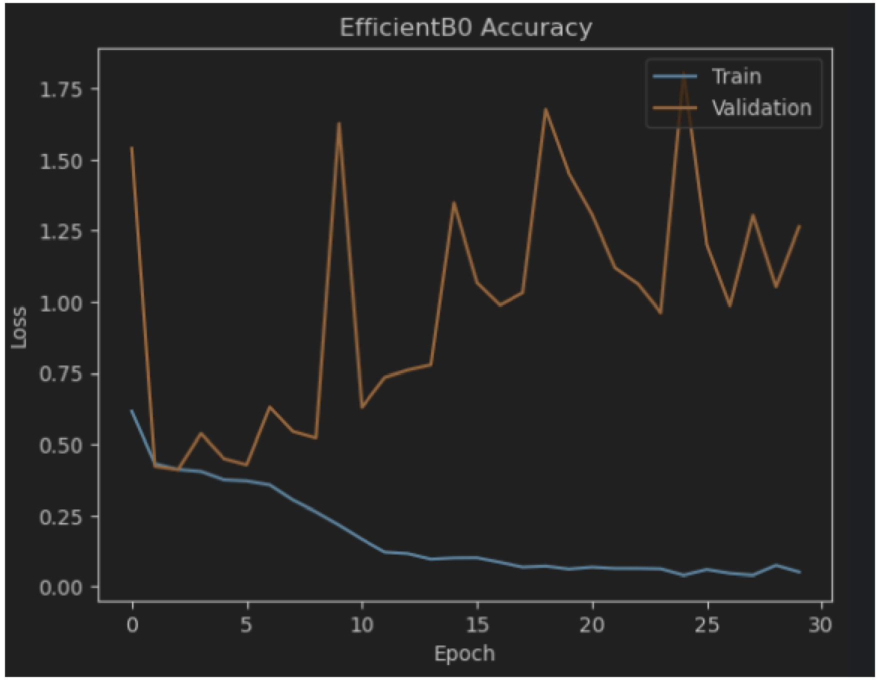
Model 2 EfficientNetB0 Parameters

- **Model Architecture**: Sequential CNN
- **Training Images:**
  - Non-Demented: 4000
  - Very Mild: 2000
  - Mild: 4001
  - Moderate: 390
- **Test Images:**
  - Non-Demented: 2599
  - Very Mild: 248
  - Mild: 496
  - Moderate: 3
- **Optimization Techniques**:
  - Adam optimizer
  - Loss Function: Categorical Cross Entropy
- Processing
  - one-hot encoding for label categorization
  - 20% validation split

Model 2 EfficientNetB0 Training Results

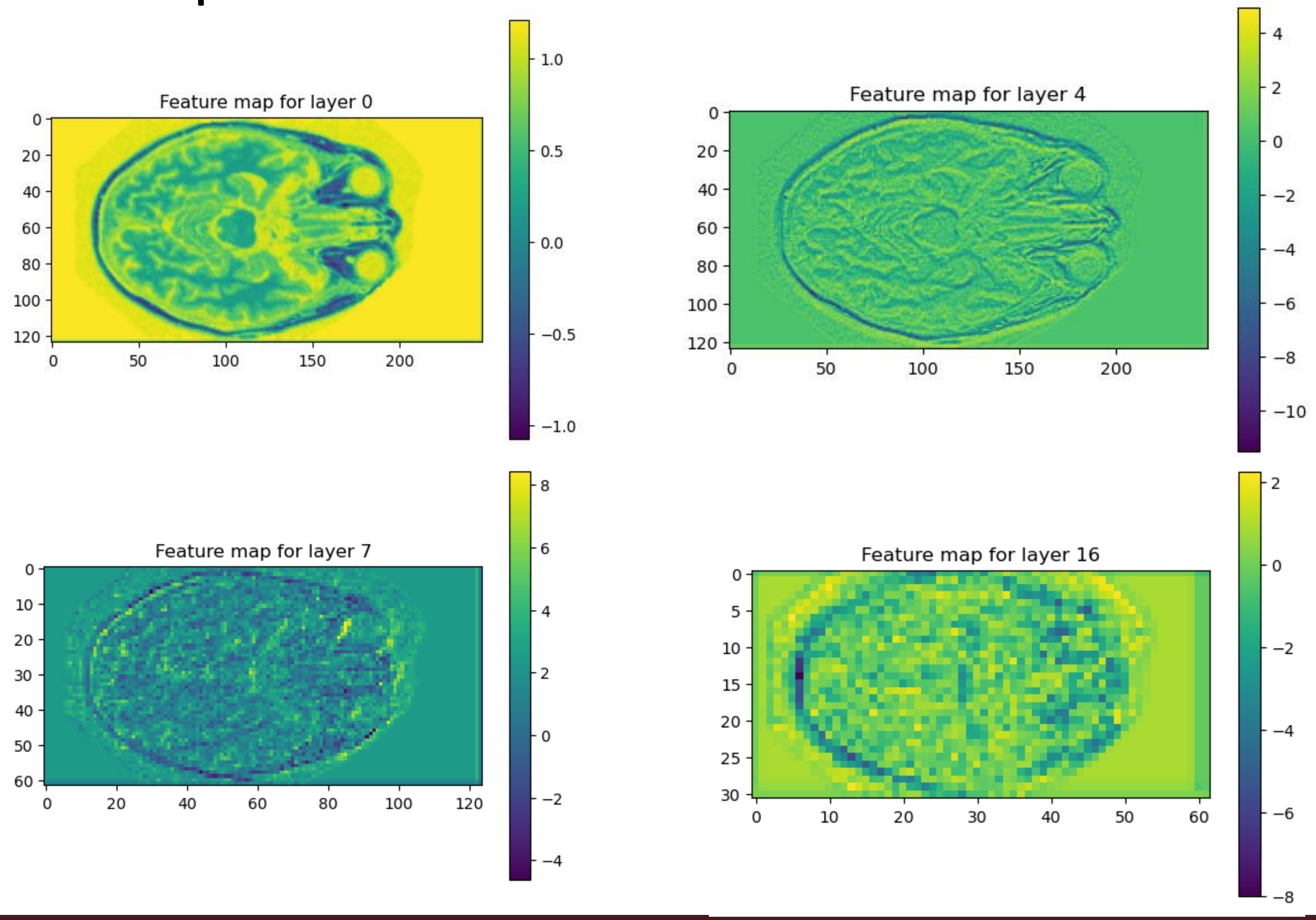
- **Test Accuracy: 68.99**
- **Precision: 0.7174**
- **Recall: 0.6899**
- **F1 Score: 0.7021**



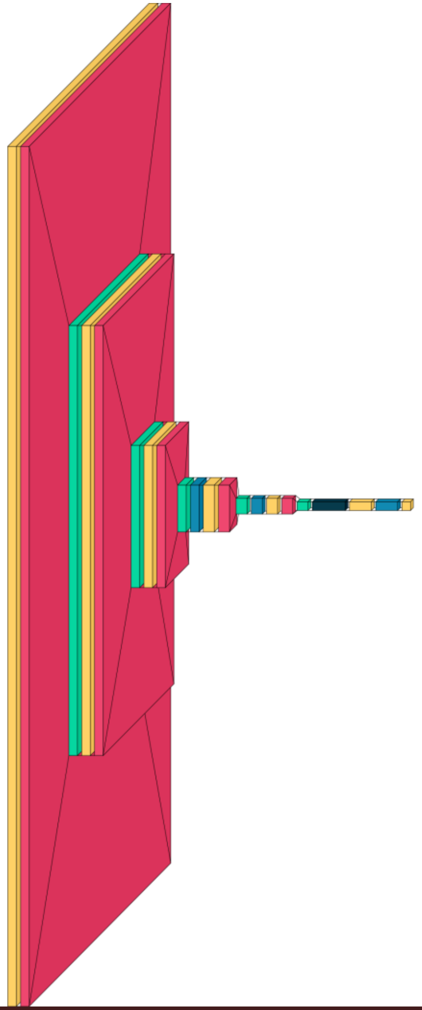


Model 1 Feature Maps

- **Highlight Key Visual Cues** to reveal textures, edges, pattens the network detect at different layers.
- **Understanding Model Focus** by examining the feature maps it helps to understand the models decision-making process and to ensure it aligns as expected
- **Progressive Complexity** in feature maps increases with each layer, moving from simple edges in early layers to more abstract features in deeper layers, indicating the hierarchical nature of feature extraction in CNNs.



Model 3 Binary Classification Architecture



Architecture			
Input Layer	**Input Shape: 248x496x3**	**Regularization**	**Notes**
Conv2D	32 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 2x2		
Conv2D	64 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same, Strides: 1x1
BatchNormalization			
MaxPooling2D	Pool Size: 3x3		
Conv2D	128 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 3x3		
Dropout	Rate: 0.5		
Conv2D	256 filters, Kernel Size: 5x2, Activation: ReLU		Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 2x2		
Dropout	Rate: 0.5		
Conv2D	256 filters, Kernel Size: 5x2, Activation: ReLU	L2(0.001)	Padding: same
BatchNormalization			
MaxPooling2D	Pool Size: 2x2		
Flatten			
Dense	512 units, Activation: ReLU	L2(0.001)	
Dropout	Rate: 0.5		
Output Layer	4 units, Activation: Softmax		

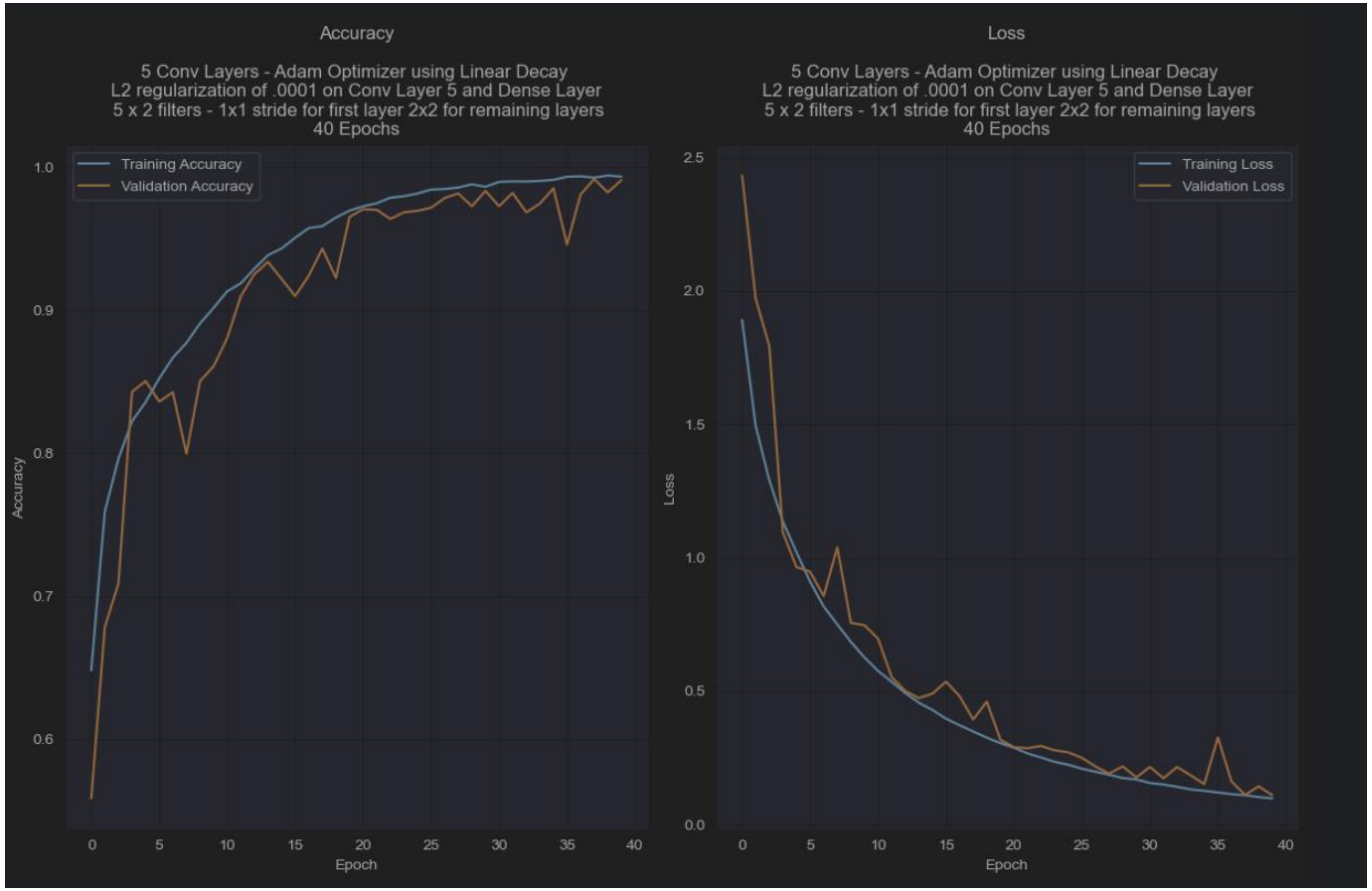
Model 3 Binary Classification Parameters

- **Model Architecture**: Sequential CNN
- **Processing**
  - one-hot encoding for label categorization
  - 20% validation split
- **Training Images:**
  - Non-Demented: 7000
  - Demented: 7000
- **Test Images:**
  - Non-Demented: 1400
  - Demented: 1400

- **Optimization Techniques**:
  - Adam optimizer with .0001 learning rate
  - Custom learning rate scheduler with linear decay
  - L2 Regularization
  - Kernal size 5 x 2
  - Loss: binary_crossentropy

Model 3 Binary Classification Training and Test Results

- **Test Accuracy: .9600**
- **Precision: .9321**
- **Recall: .9484**
- **F1 Score: .9642**



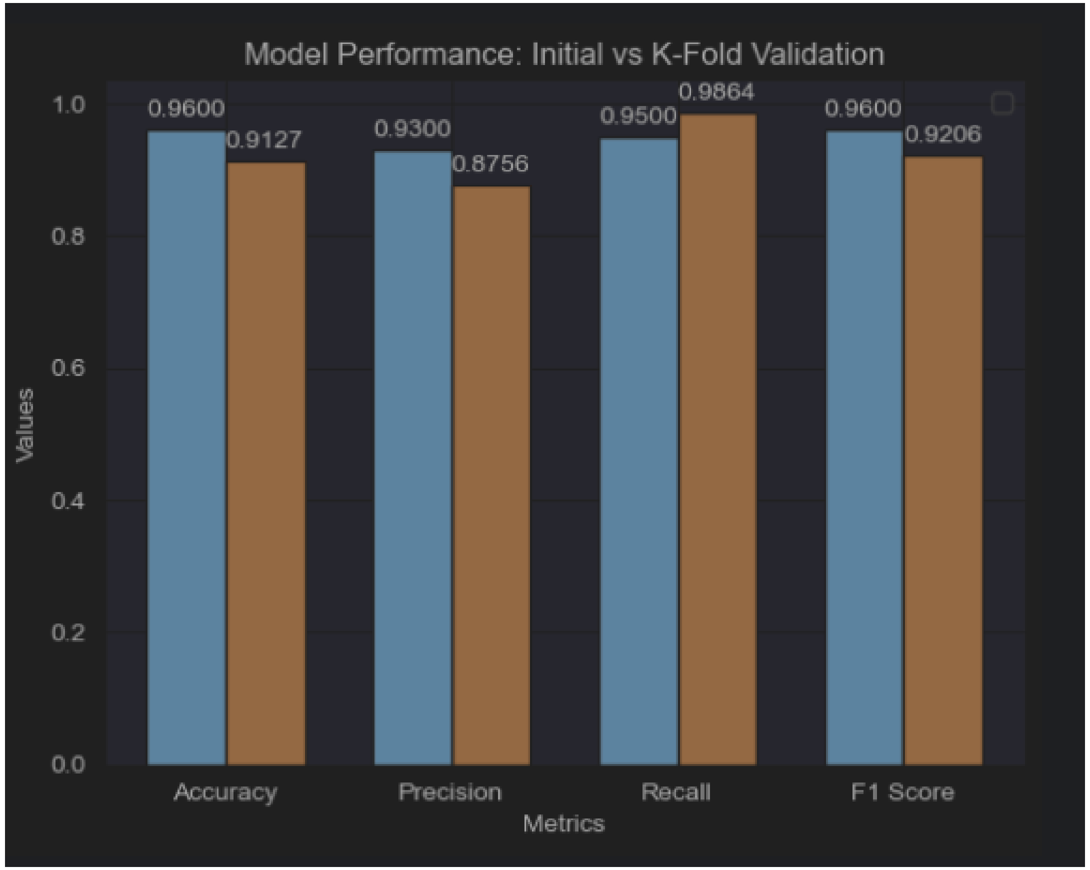
DataLab12.github.io

TEXAS STATE UNIVERSITY

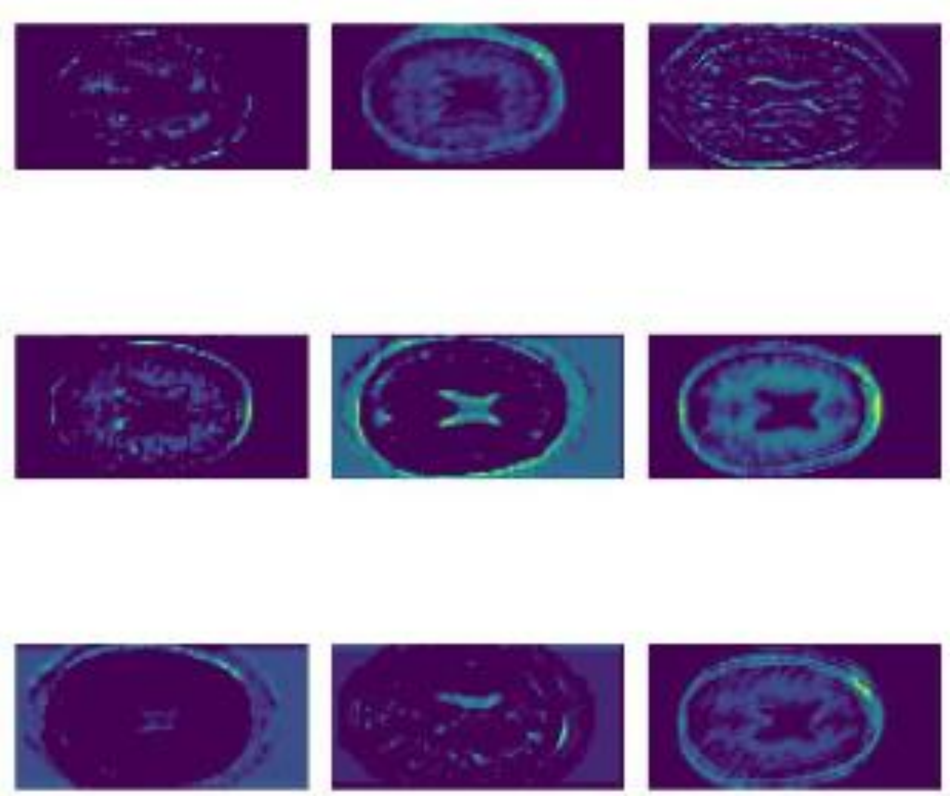
Model 3 K-Fold Cross Validation Results

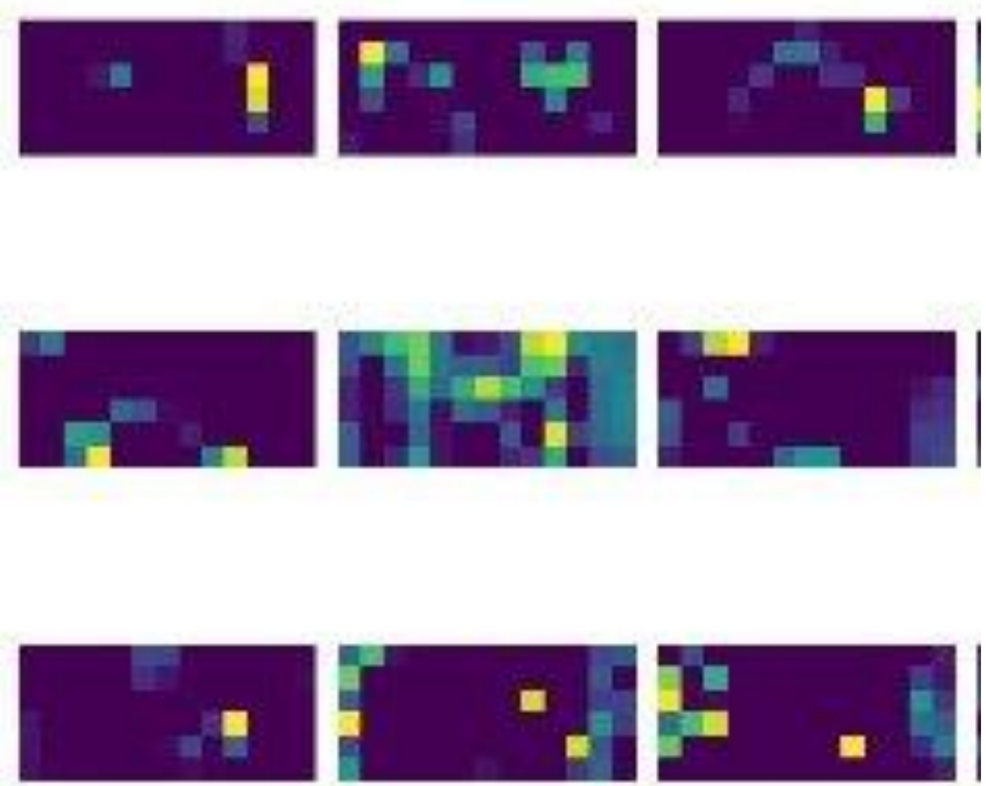
K-fold validation is a statistical technique that divides a dataset into multiple subsets, using each in turn for testing a model trained on the remaining data to ensure generalized evaluation of its performance.

- **Average Test Accuracy: .9127**
- **Average Precision: .8756**
- **Average Recall: .9864**
- **Average F1 Score: 0.9206**



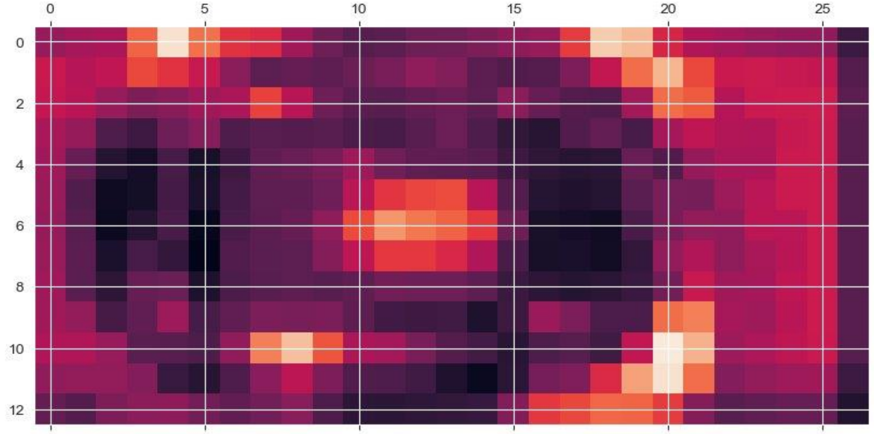
Model 1 Feature Maps Layer 1 and 3 Non-Demented



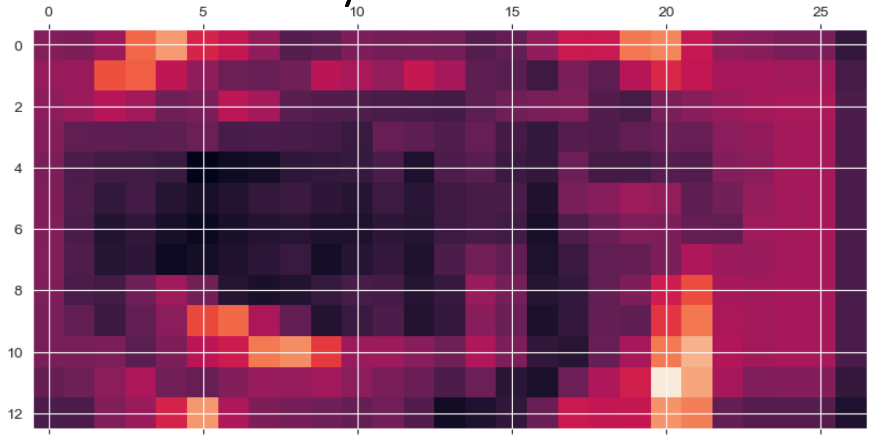


Model 1 Heat Map Layer 3

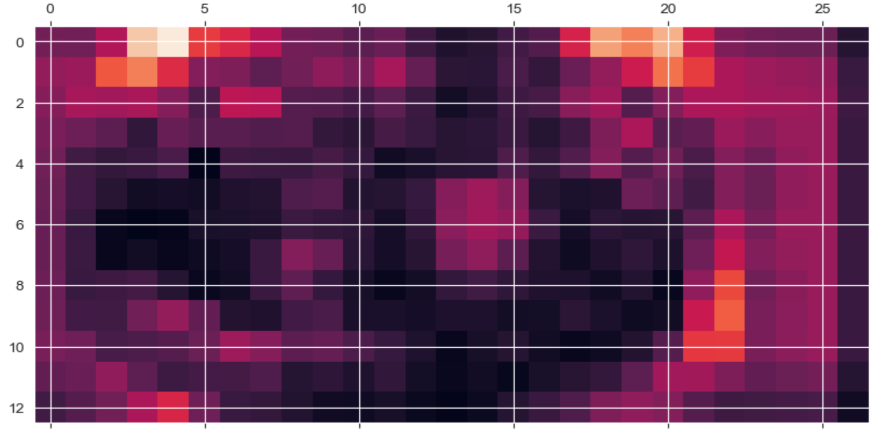
Non-Demented



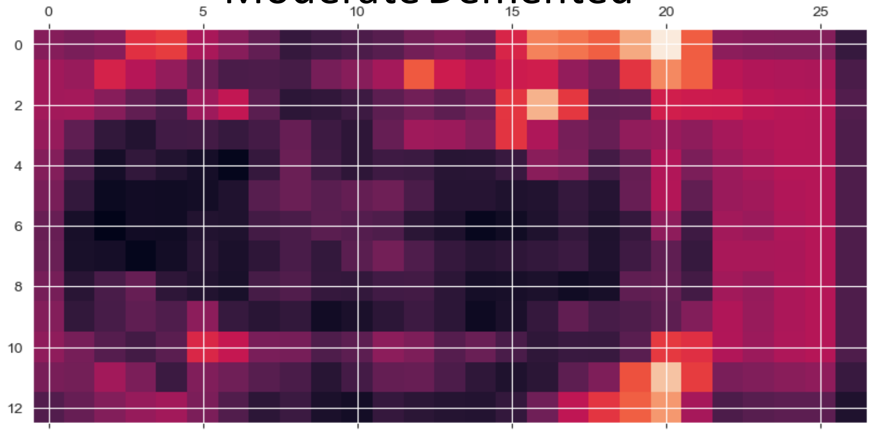
Very Mild-Demented



Mild-Demented



Moderate Demented



Results

Categorical	Default	L1	No Reg	Kernal 2x2
Test Acc	0.7934	0.7607	0.7772	0.7364
Precision	0.6661	0.6226	0.7343	0.5873
Recall	0.7934	0.7607	0.7772	0.7226
F1	0.7171	0.6832	0.6994	0.6462

Binary Model	Default	L1	No Reg	Kernal 2x2
Test Acc	0.9600	.8050	.8890.	.8590
Precision	0.9321	.7195	.8123	.7885
Recall	0.9484	.8050	.9989	1
F1	0.9642	.8369	.8960	.8817

EfficientNetB0	Default
Test Acc	0.7934
Precision	0.6661
Recall	0.7934
F1	0.7171