

Classifying Brain Scans with Machine Learning

Violet Davis and Annie Friar

Presentation Outline

- Dataset & Data Exploration
- Data Pre-Processing
- Models
 - KMeans & CNN
 - CNN
- Tuning & Results

Our Dataset

Dataset from Kaggle:

“Uncovering Knowledge: A Clean Brain Tumor Dataset for Advanced Medical Research”

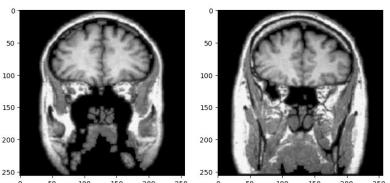
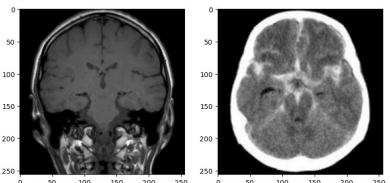
MRI Brain Scans with 4 categories:

No Tumor, Glioma Tumor, Meningioma Tumor, Pituitary Tumor

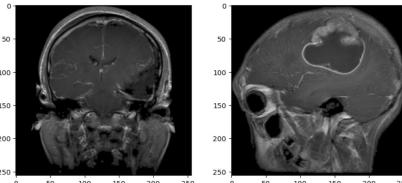
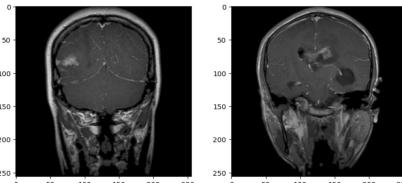
Problem Statement: Can we use machine learning to help radiologists classify brain scans quicker and with less error?

Sample Images

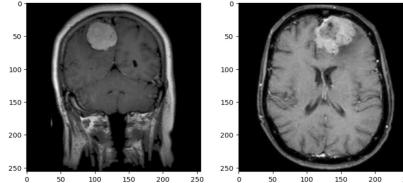
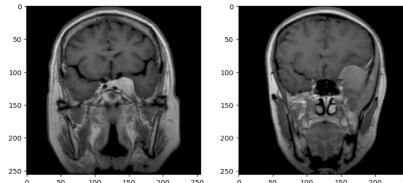
Normal



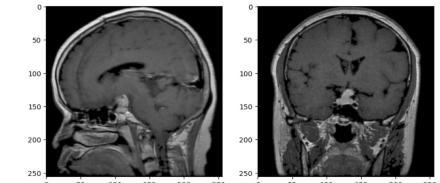
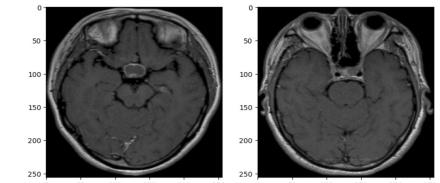
Glioma



Meningioma

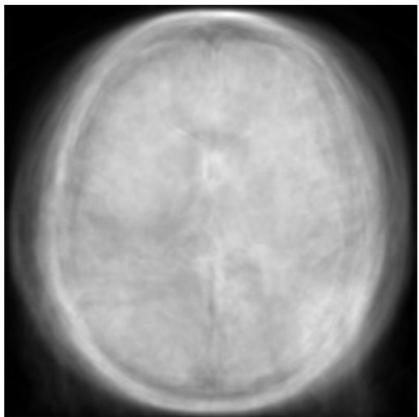


Pituitary

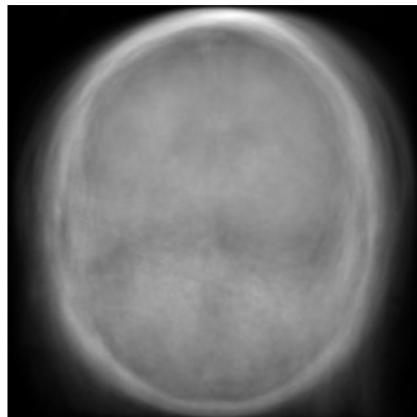


Average Images

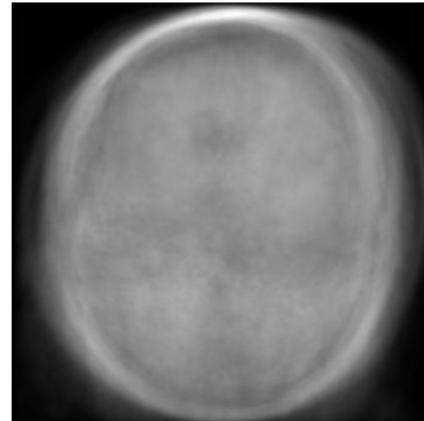
Normal



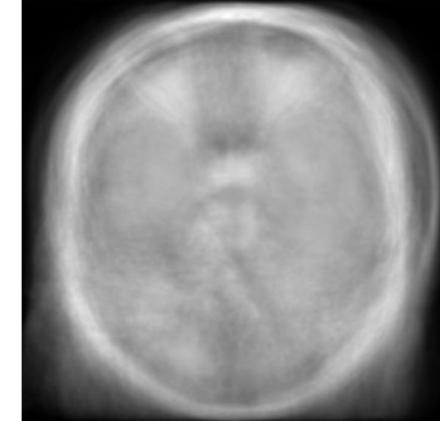
Glioma



Meningioma



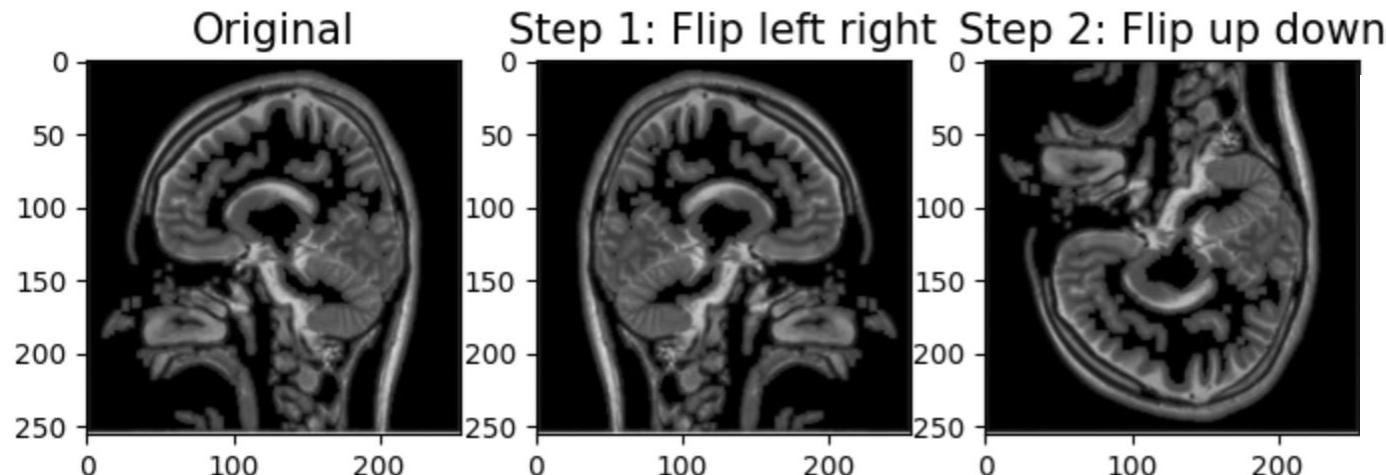
Pituitary



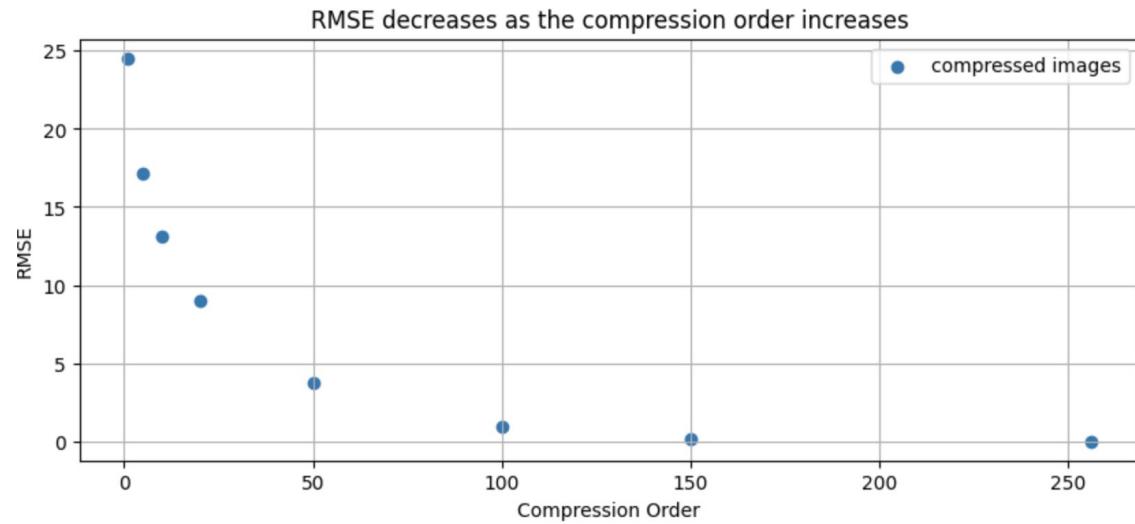
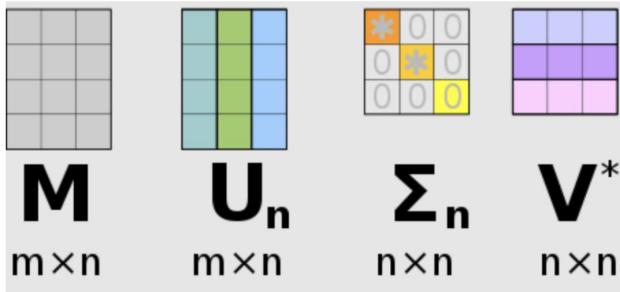
Data Pre-Processing

- Image Augmentation
- Singular Value Decomposition (SVD)

Image Augmentation

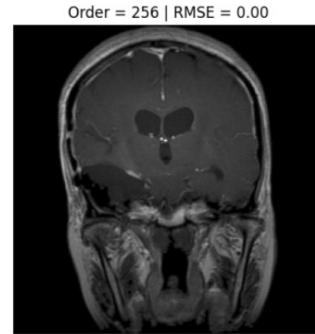
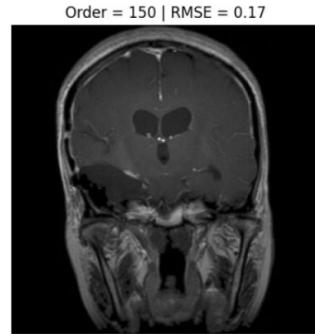
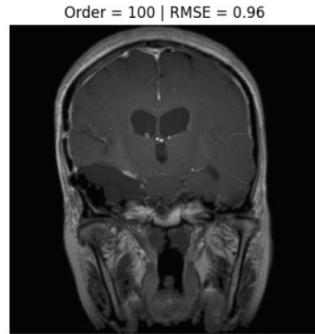
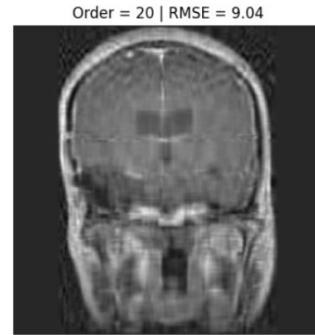
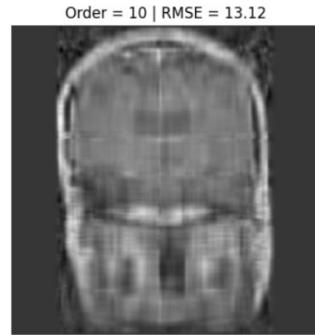
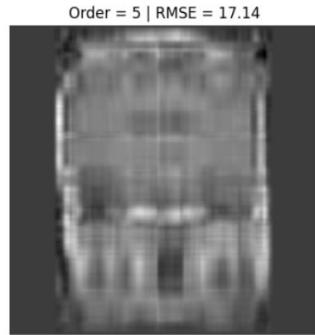
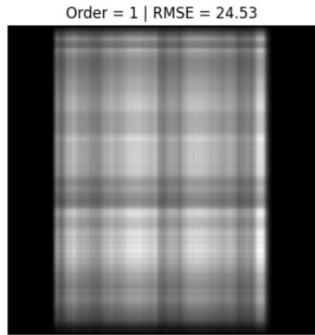


Singular Value Decomposition (SVD)



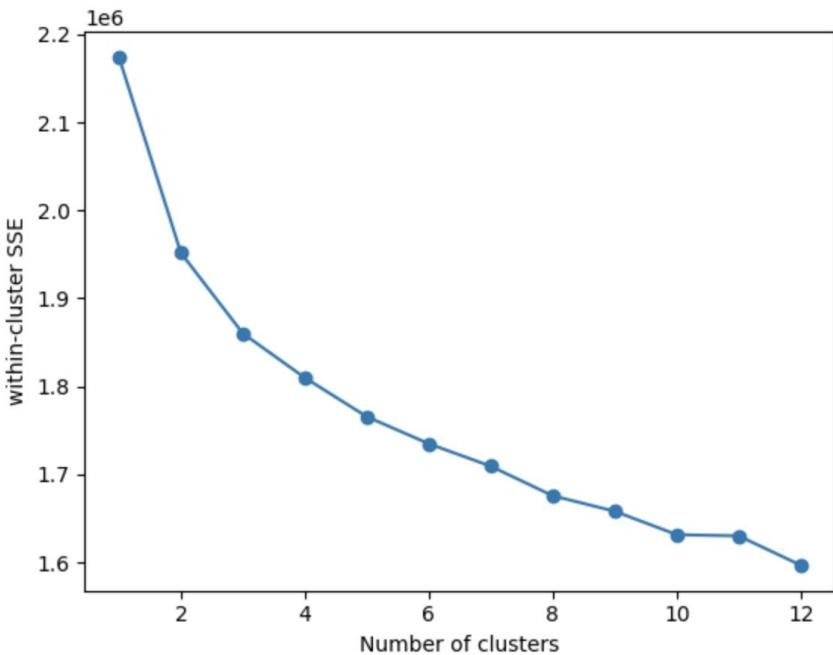
SVD Dimensionality Reduction

Compression at different orders



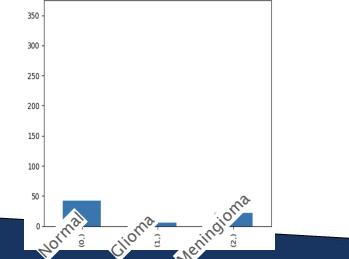
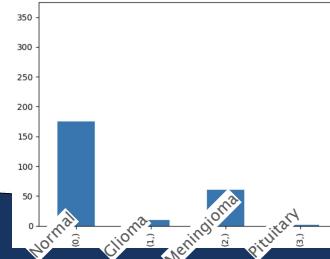
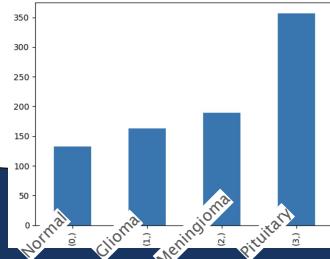
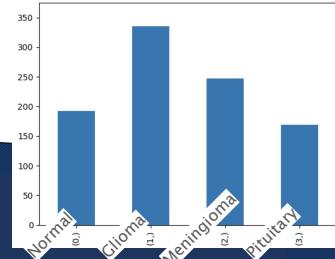
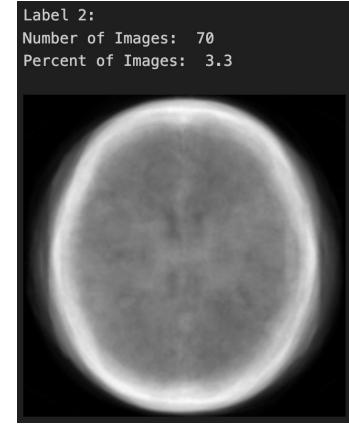
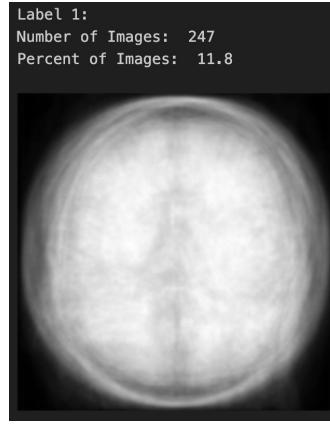
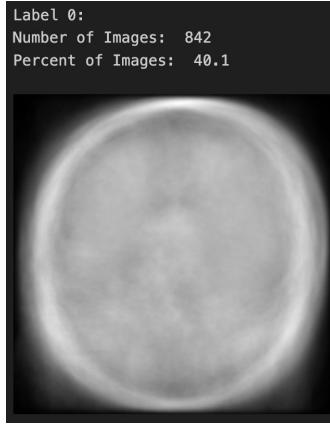
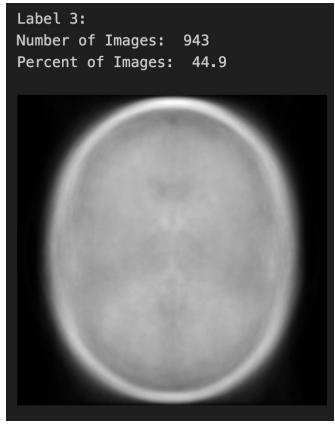
Unsupervised Learning: KMeans

- We know we have four categories of tumors
- We can also use the elbow method to determine the appropriate number of clusters



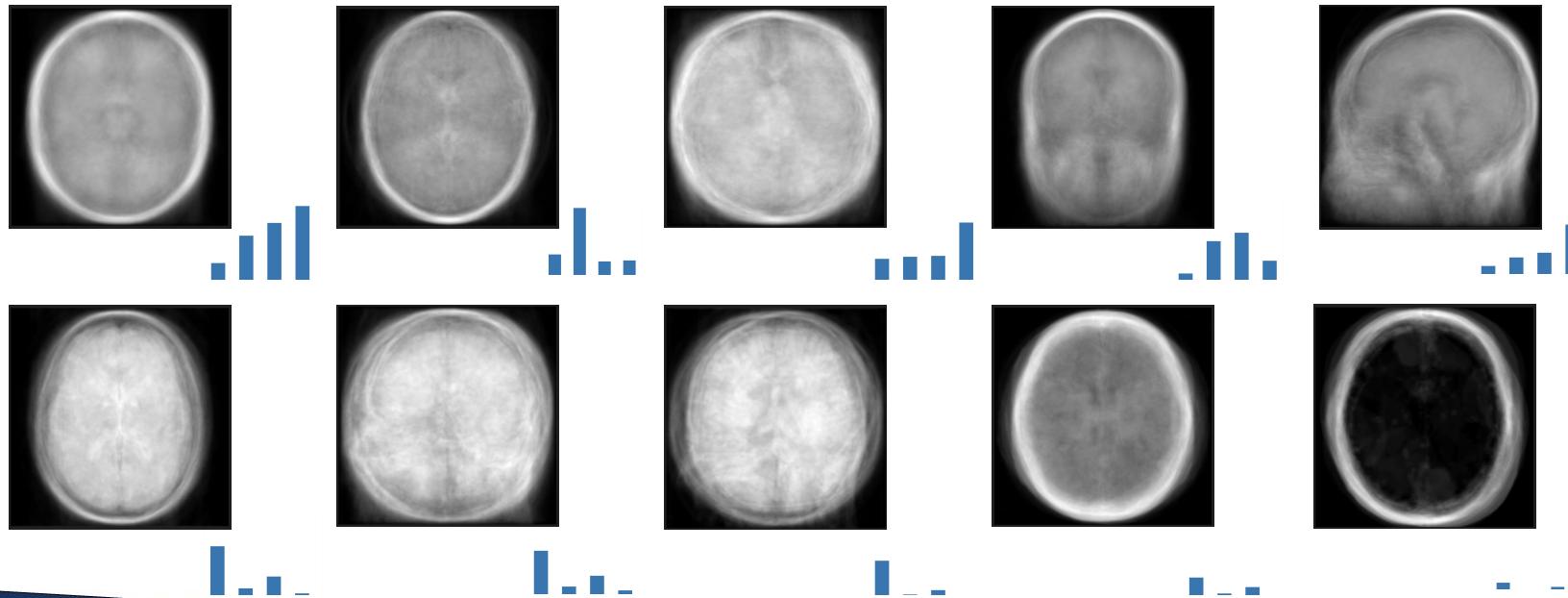
KMeans Categories: Average Images

Four Groups



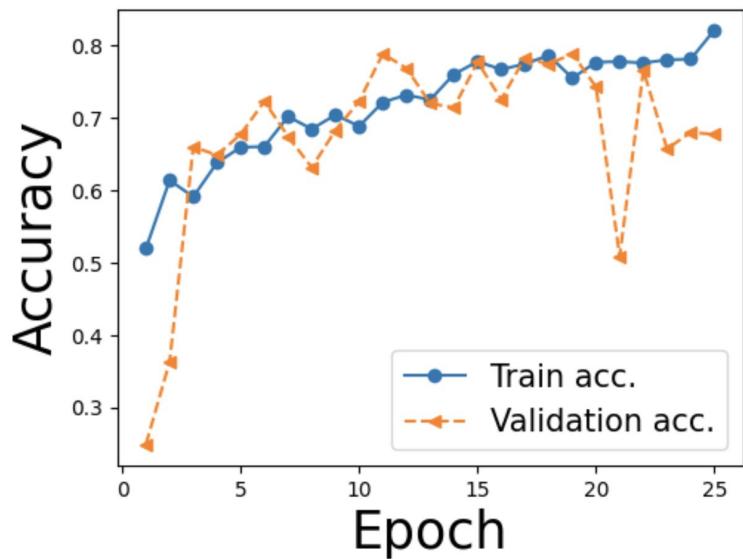
KMeans Categories: Average Images

Ten Groups

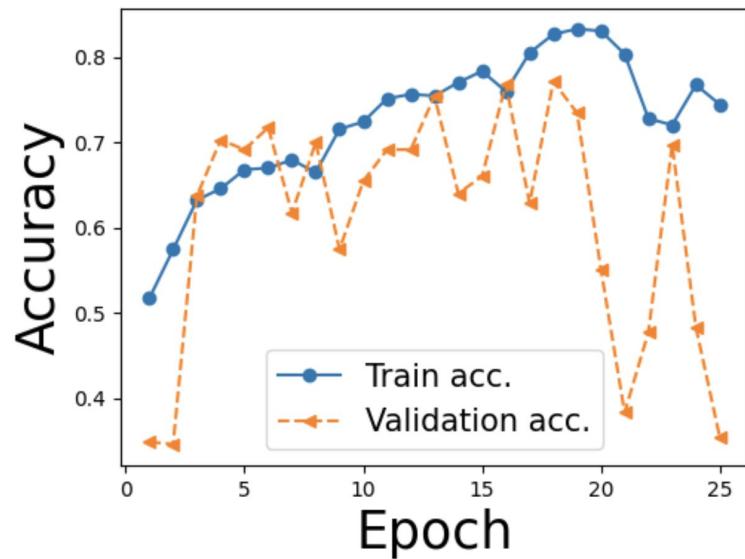


CNN & KMeans Models

Four Groups



Ten Groups



CNN Model Architecture

- 3 Layers each:
 - Convolutional with ReLU
 - Batch Normalization
 - Max Pooling
- 2 Dense Layers with Dropout in between:
 - ReLU
 - Softmax

Model: "CNN"

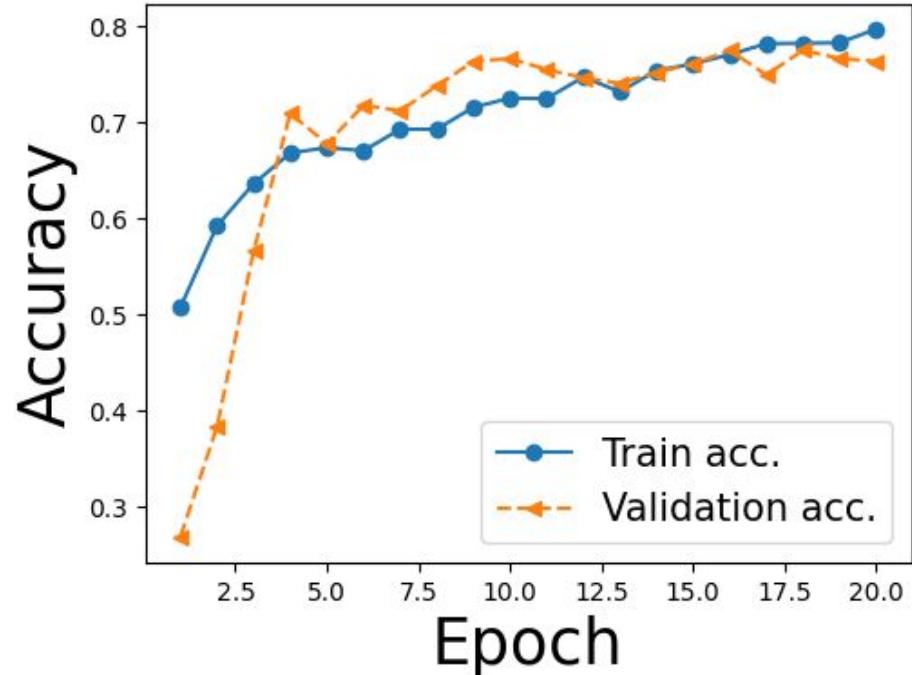
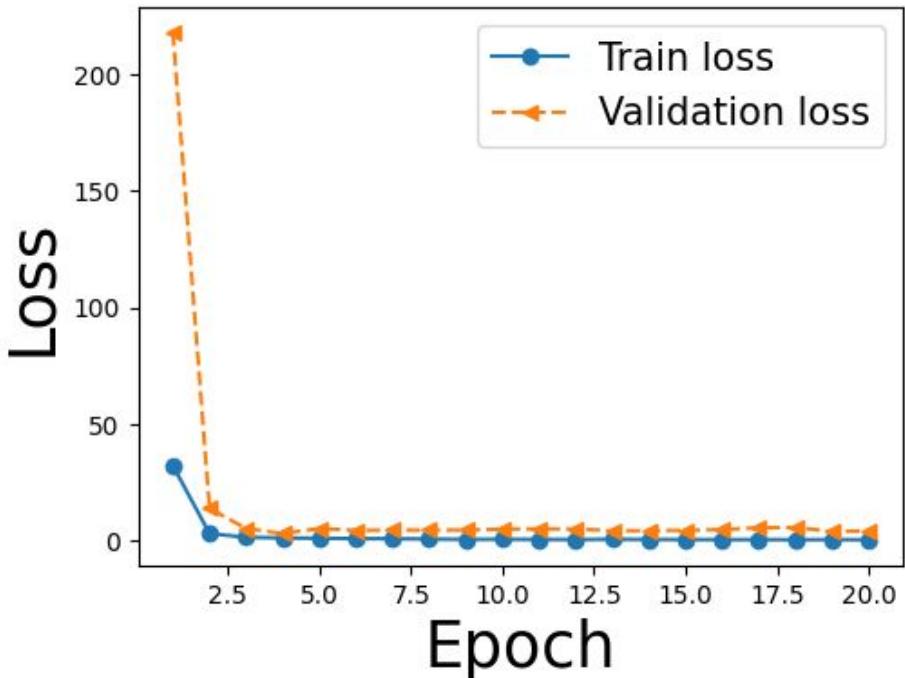
Layer (type)	Output Shape	Param #
Images (InputLayer)	[(None, 256, 256, 1)]	0
add (Add)	(None, 256, 256, 1)	0
conv_1 (Conv2D)	(None, 256, 256, 32)	832
batch_normalization (Batch Normalization)	(None, 256, 256, 32)	128
pool_1 (MaxPooling2D)	(None, 128, 128, 32)	0
conv_2 (Conv2D)	(None, 128, 128, 64)	51264
batch_normalization_1 (BatchNormalization)	(None, 128, 128, 64)	256
pool_2 (MaxPooling2D)	(None, 64, 64, 64)	0
conv_3 (Conv2D)	(None, 64, 64, 128)	204928
batch_normalization_2 (BatchNormalization)	(None, 64, 64, 128)	512
pool_3 (MaxPooling2D)	(None, 32, 32, 128)	0
flatten (Flatten)	(None, 131072)	0
dense (Dense)	(None, 1024)	134218752
dropout (Dropout)	(None, 1024)	0
Predictions (Dense)	(None, 4)	4100
<hr/>		
Total params: 134480772 (513.00 MB)		
Trainable params: 134480324 (513.00 MB)		
Non-trainable params: 448 (1.75 KB)		

Batch Normalization

- Layer that normalizes its inputs
- Batch normalization applies a transformation that maintains the mean output close to zero and the output standard deviation close to one

Original Model

- * kernel size: 5
- * strides: 1
- * pool size: 2
- * learning rate: 0.001
- * optimizer: Adam
- * SVD order: 50
- * flip: yes



Parameter Tuning

Hyperparameter Tuning:

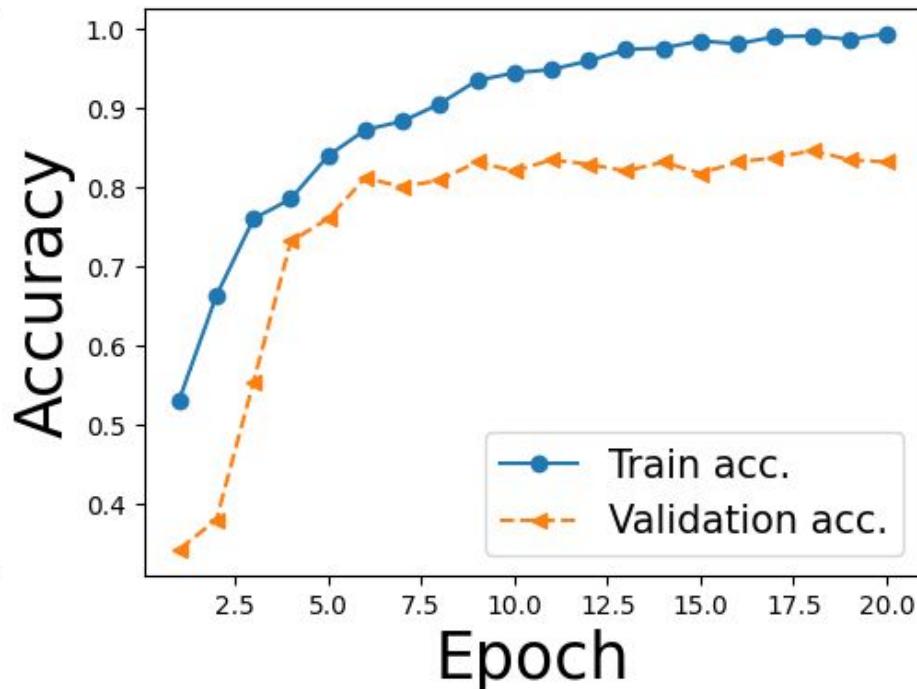
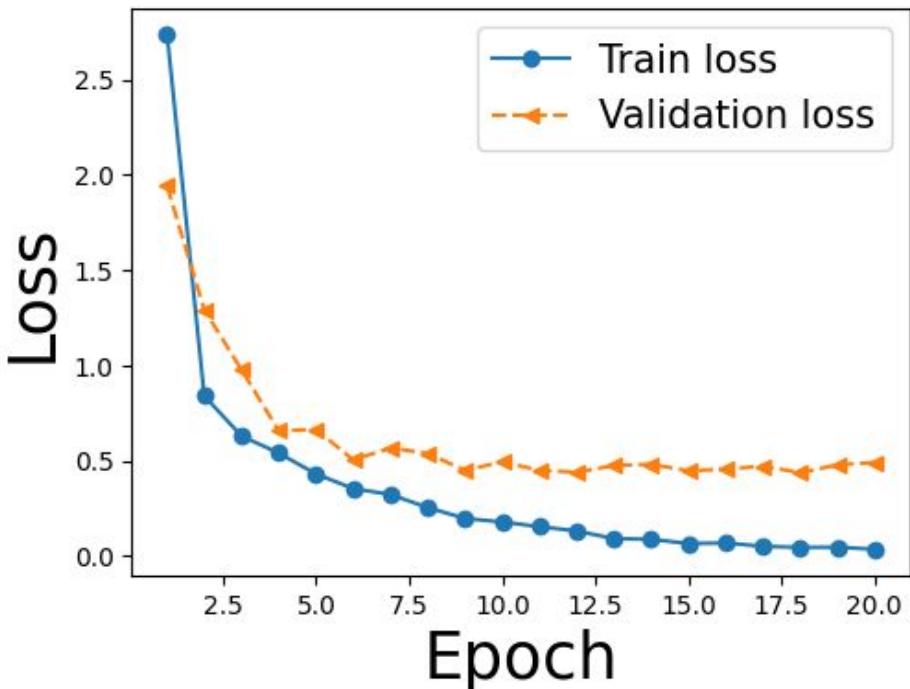
Training accuracy	Validation accuracy	kernel size	strides	pool size	learning rate	optimizer	SVD order	flip_on_train
0.7797	0.7971	5	1	2	0.001	Adam	50	yes
0.8544	0.8029	3	1	2	0.001	Adam	50	yes
0.9772	0.6629	5	2	2	0.001	Adam	50	yes
0.9177	0.8229	5	1	3	0.001	Adam	50	yes
0.2612	0.2629	5	1	2	0.01	Adam	50	yes
0.9600	0.8086	5	1	2	0.001	SGD	50	yes
0.8083	0.7629	5	1	2	0.001	Adam	100	yes
0.9096	0.8257	5	1	2	0.001	Adam	50	no

Additional tests:

Training accuracy	Validation accuracy	kernel size	strides	pool size	learning rate	optimizer	SVD order	flip_on_train
0.9510	0.8029	3	1	3	0.001	Adam	50	yes
0.9324	0.7829	3	1	3	0.001	Adam	50	no
0.8896	0.8086	3	1	2	0.001	Adam	50	no
0.8792	0.8114	5	1	3	0.001	Adam	50	no
0.9924	0.8400	3	1	3	0.001	SGD	50	yes
0.9838	0.8400	3	1	3	0.001	SGD	50	no
0.9857	0.8057	3	1	2	0.001	SGD	50	no
0.9800	0.8400	5	1	3	0.001	SGD	50	no

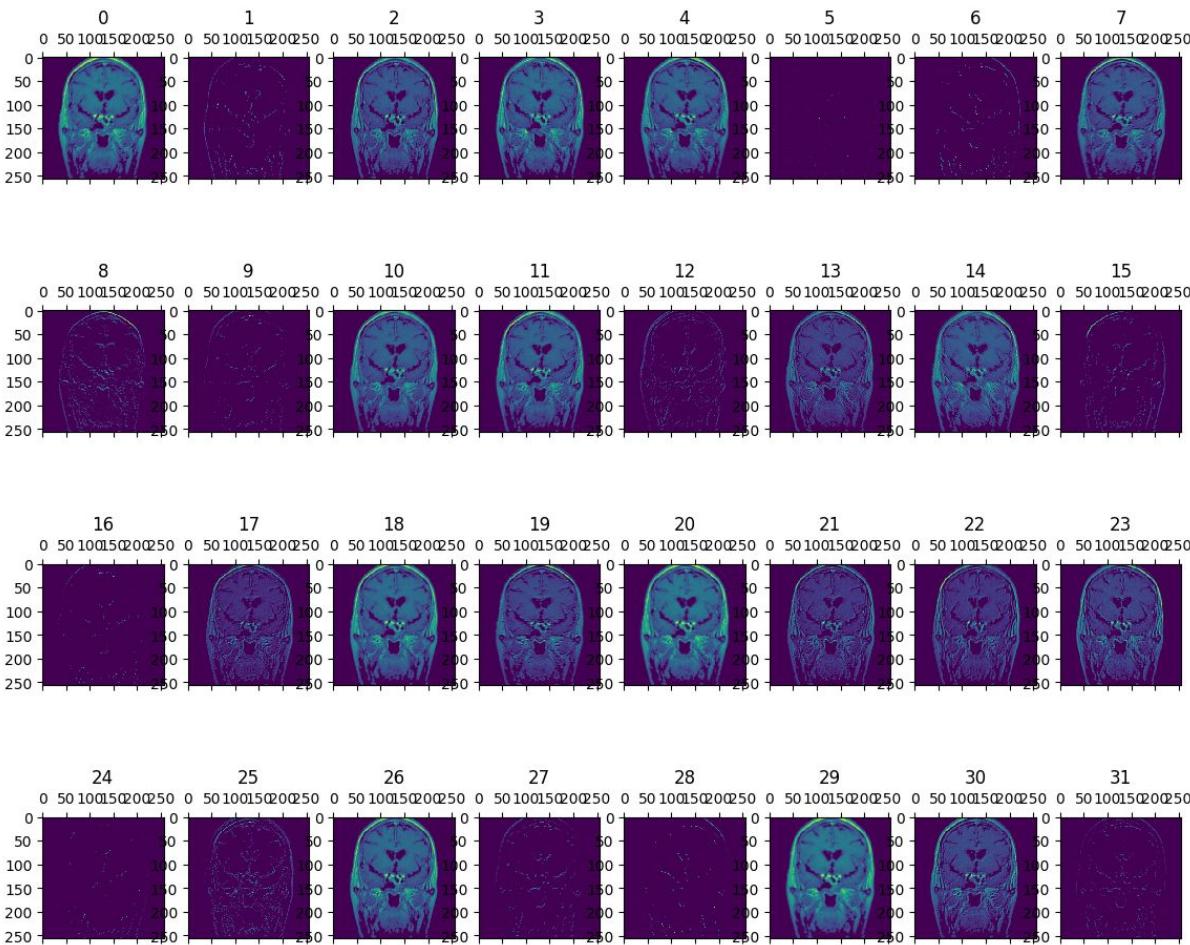
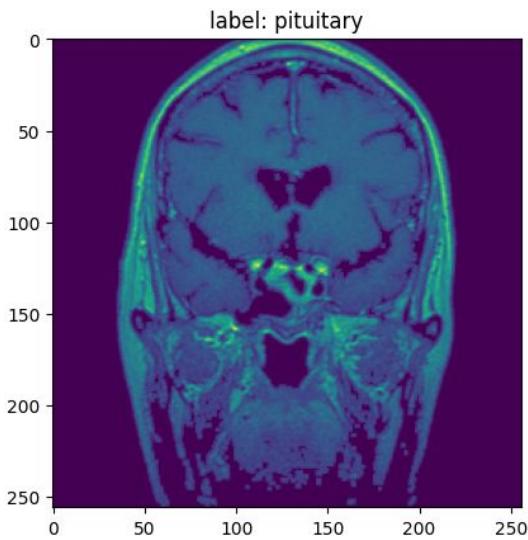
Final Model

- * kernel size: 3
- * strides: 1
- * pool size: 3
- * learning rate: 0.001
- * optimizer: SGD
- * SVD order: 50
- * flip: yes



What does the CNN learn?

Getting activations of layer 2 : conv 1
Images size is 256 x 256
Number of channels is 32

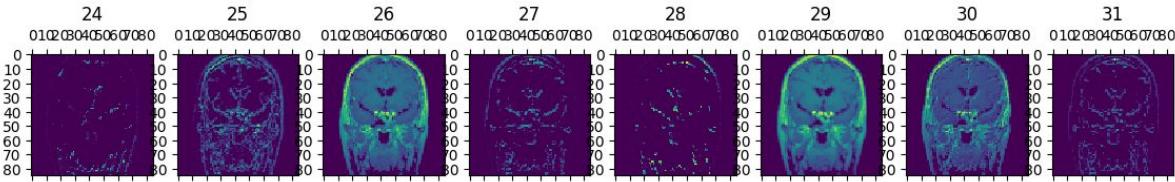
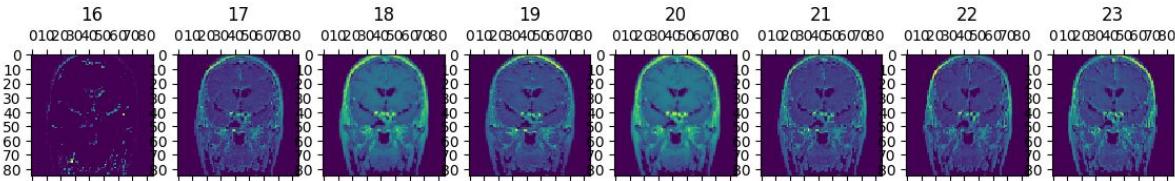
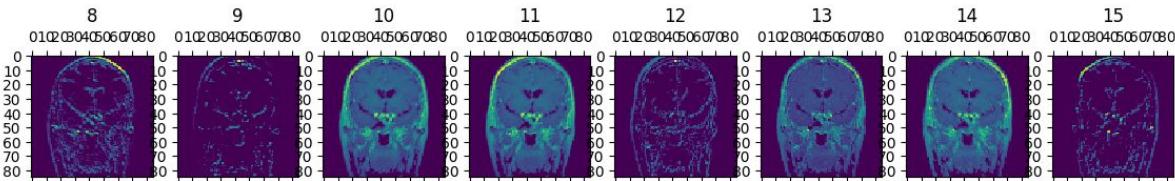
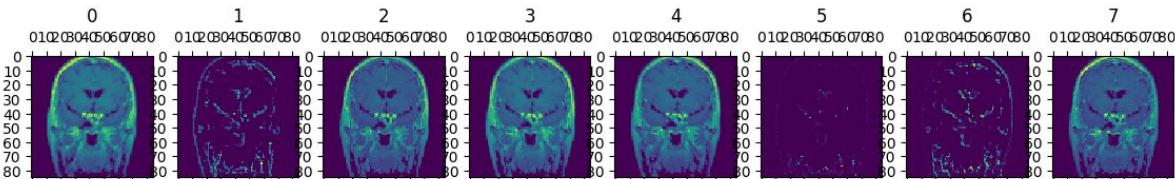


What does the CNN learn?

Getting activations of layer 4 : pool 1

Images size is 85 x 85

Number of channels is 32

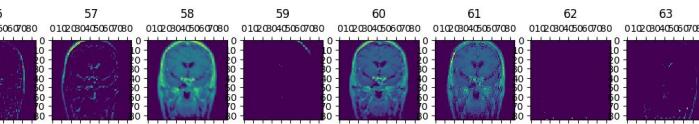
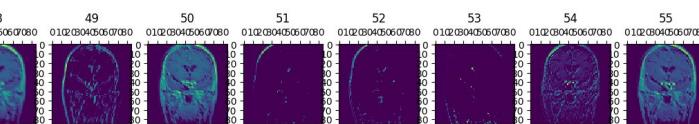
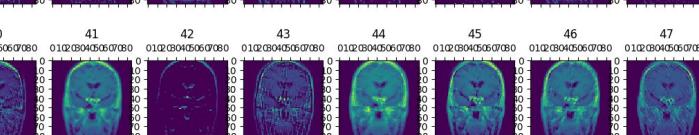
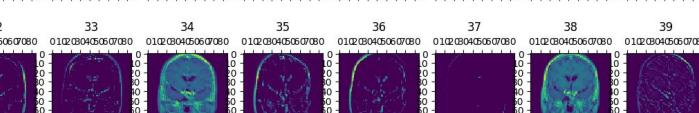
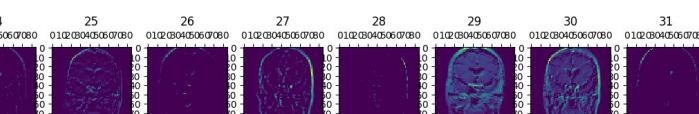
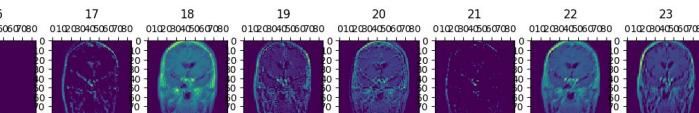
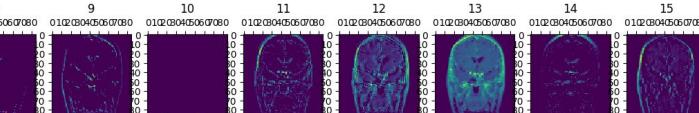
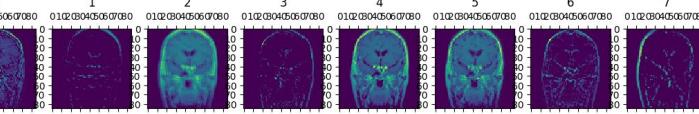


What does the CNN learn?

Getting activations of layer 5 : conv 2

Images size is 85 x 85

Number of channels is 64

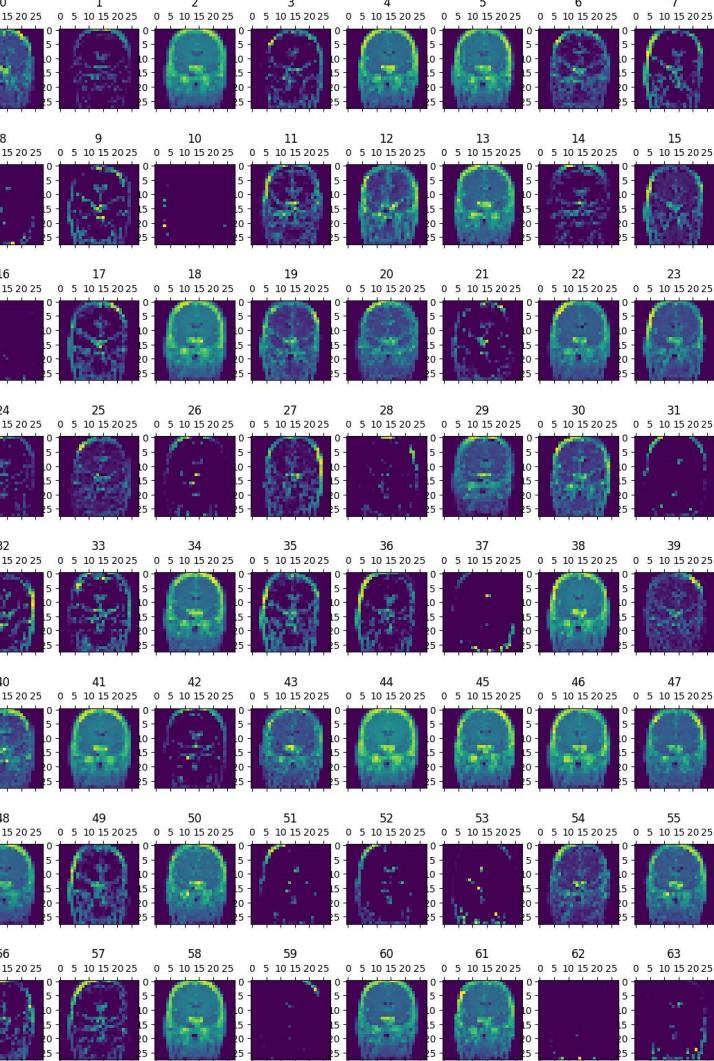


What does the CNN learn?

Getting activations of layer 7 : pool 2

Images size is 28 x 28

Number of channels is 64

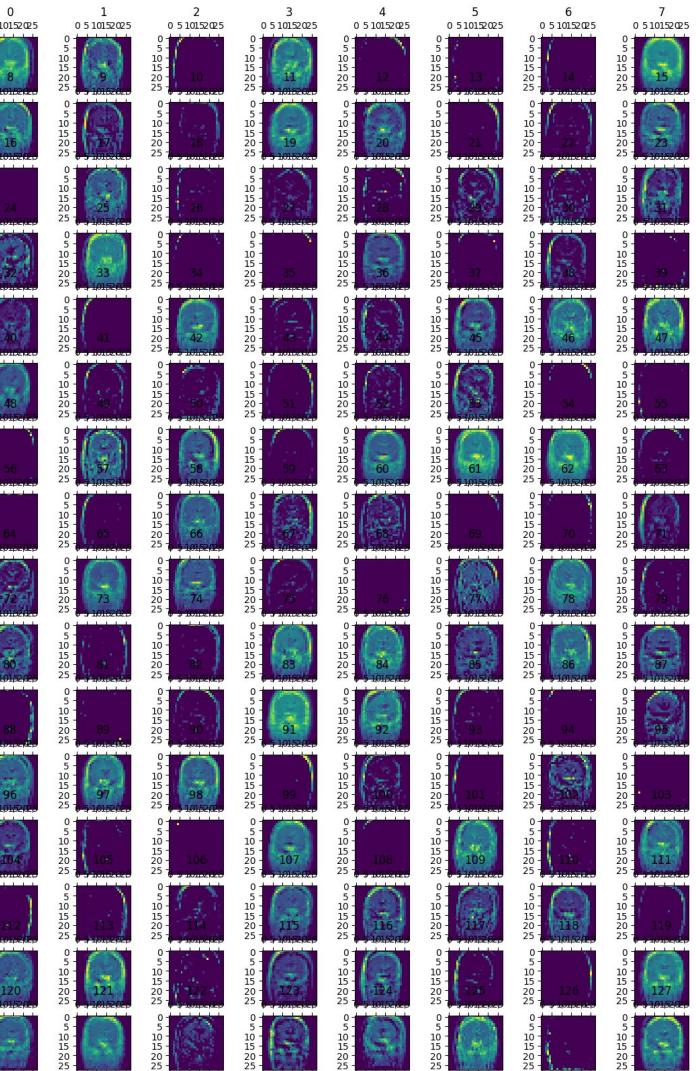


What does the CNN learn?

Getting activations of layer 8 : conv 3

Images size is 28 x 28

Number of channels is 128

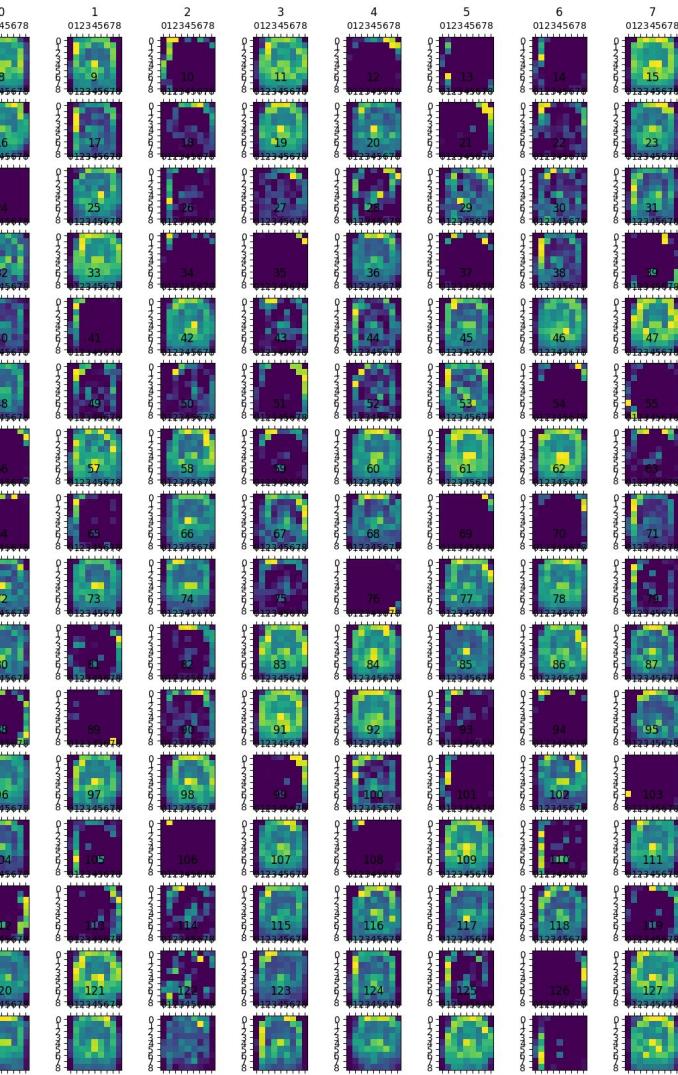


What does the CNN learn?

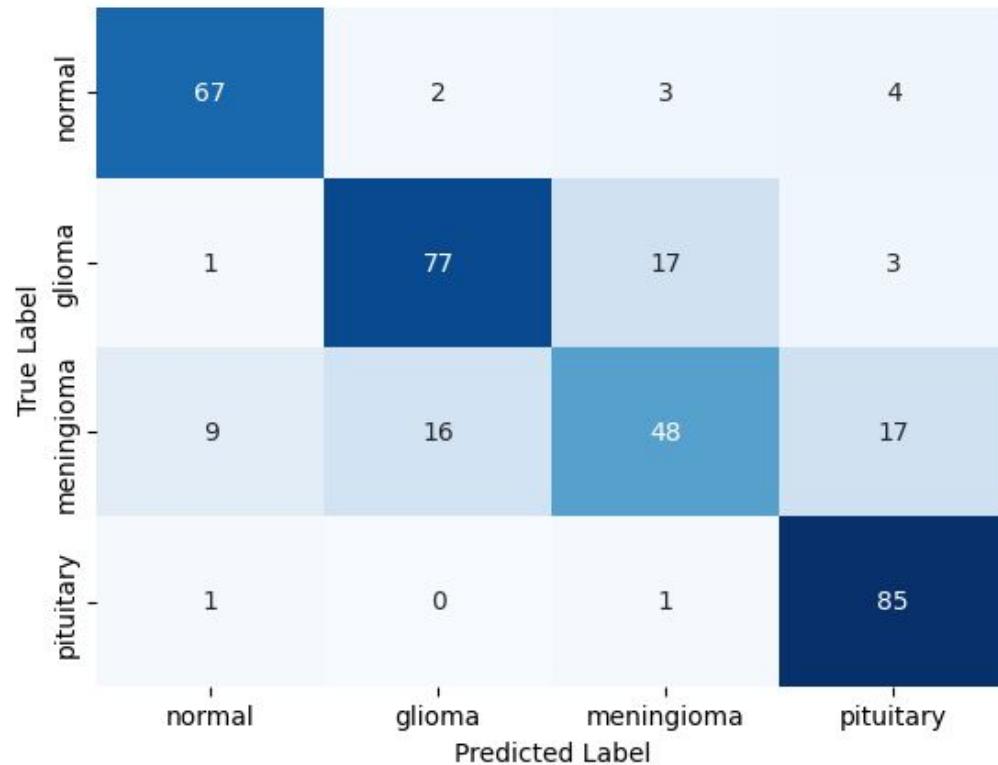
Getting activations of layer 10 : pool 3

Images size is 9 x 9

Number of channels is 128



Evaluation Metrics



	Precision	Recall
Normal	85.9%	88.1%
Glioma	81.1%	78.6%
Meningioma	69.6%	53.3%
Pituitary	78.0%	97.7%

Thank You!

Appendix

Brain Tumor Background

- Every year, around 11,700 people are diagnosed with a brain tumor.
- The 5-year survival rate for people with a cancerous brain tumor is approximately 34% for men and 36% for women.
- Brain tumors often have abnormalities in the sizes and location of the brain tumor(s), which make them difficult and time-consuming to classify correctly.

Problem Statement: Can we use machine learning to help radiologists classify brain scans quicker and with less error?

Brain Tumor Image Dataset

Original Data provided by: Sartaj, Savitribai Phule Pune University

“Uncovering Knowledge: A Clean Brain Tumor Dataset for Advanced Medical Research”

MRI Brain Scans with 4 categories: No Tumor, Glioma Tumor, Meningioma Tumor, Pituitary Tumor

Enhanced Data provided by: Thomas, Kaggle

Grayscale Normalization, Enhanced Quality, Removal of Redundant Data, Resizing to 256 x 256 pixels with Aspect Ratio Preservation

Categories Split in Dataset

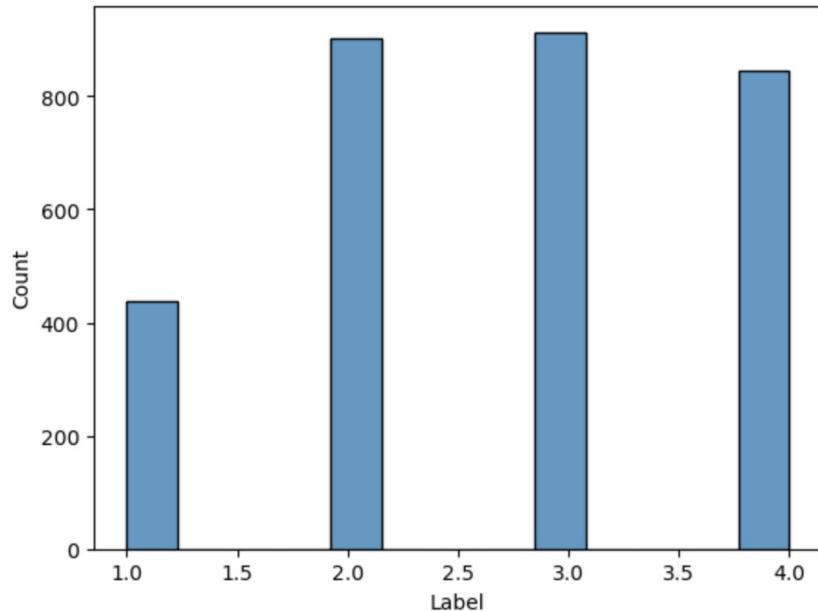
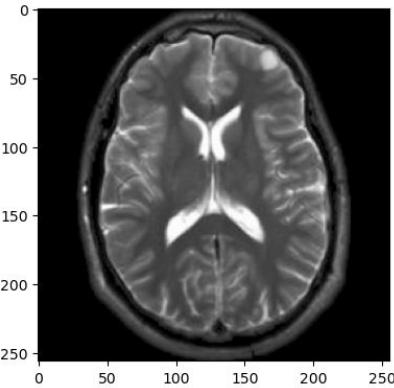
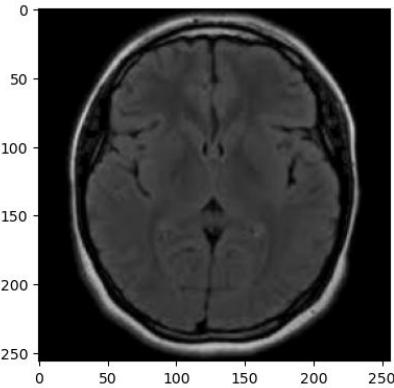
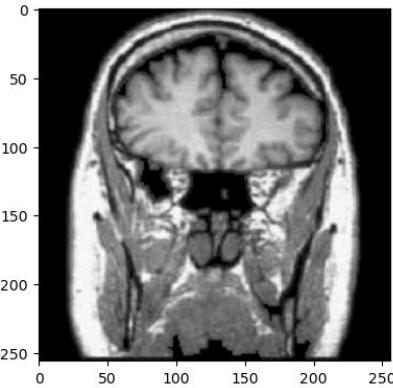
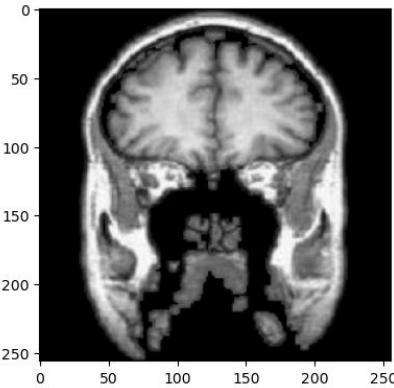
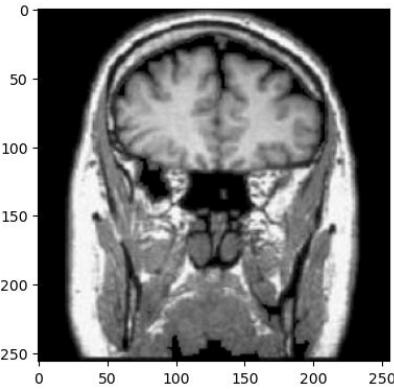
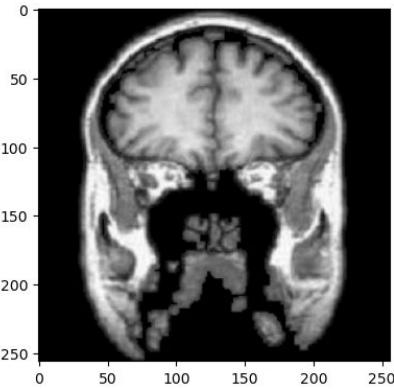
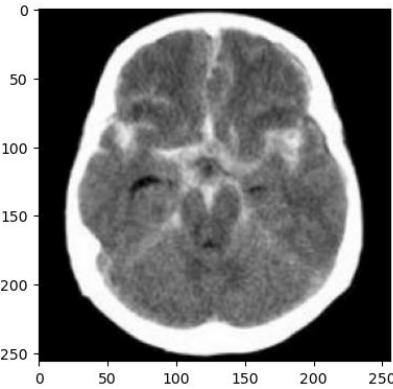
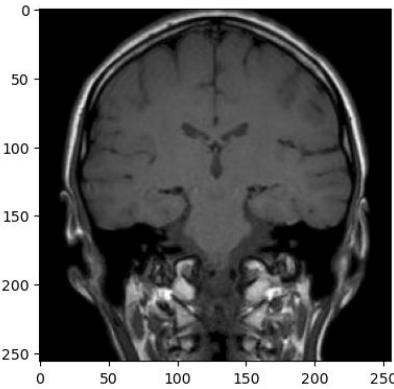


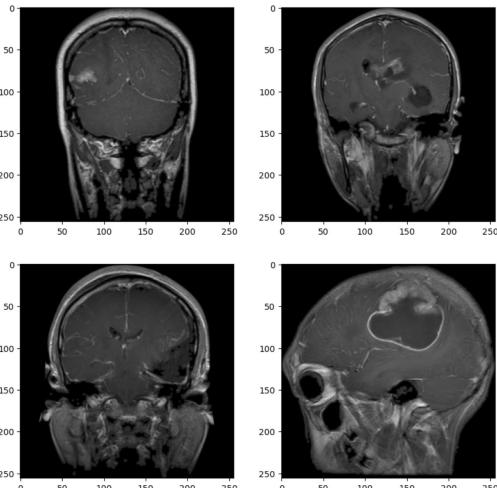
Image Type	Label	Count
Normal	1	438
Glioma	2	901
Meningioma	3	913
Pituitary	4	844

Sample of Normal Images

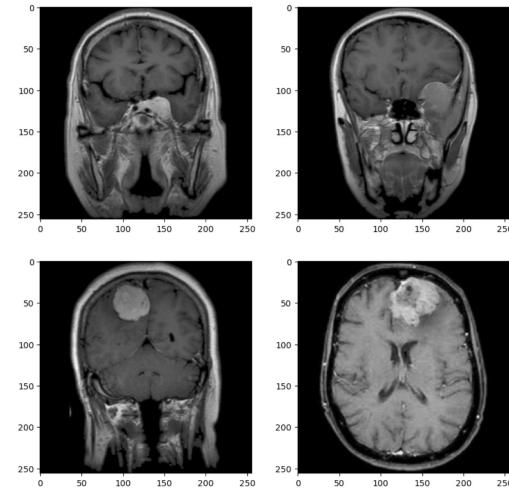


Sample of Images with Tumor Present

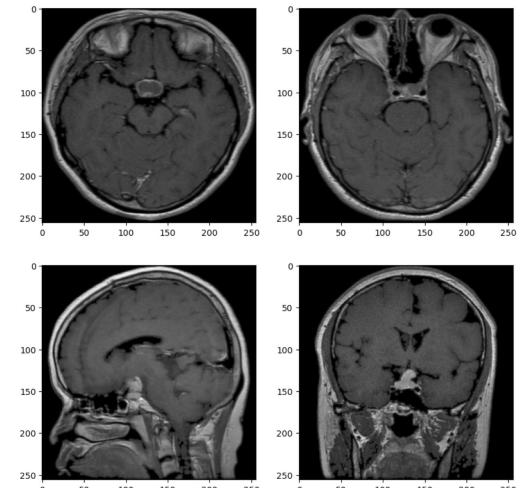
Glioma



Meningioma

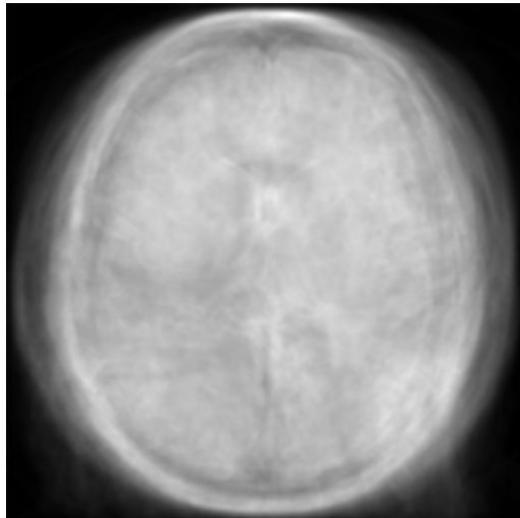


Pituitary

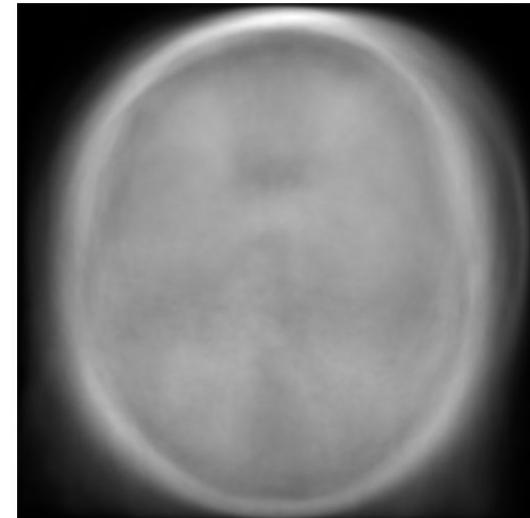


Average Images

Normal

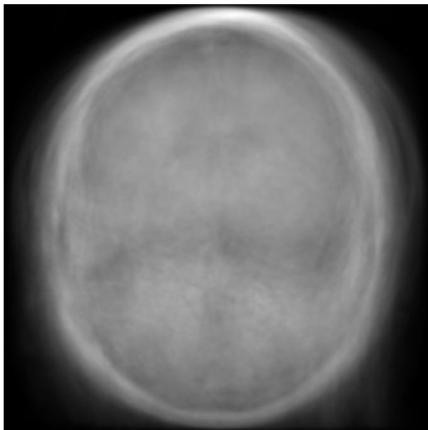


Tumor Present

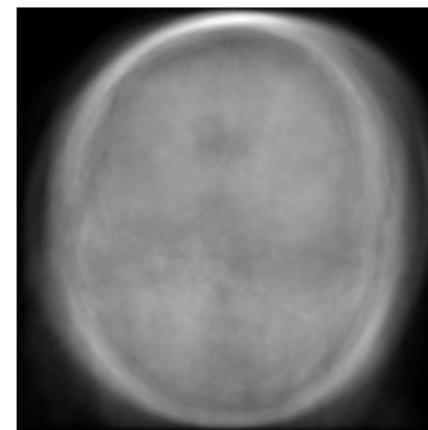


Average Images with Tumor Present

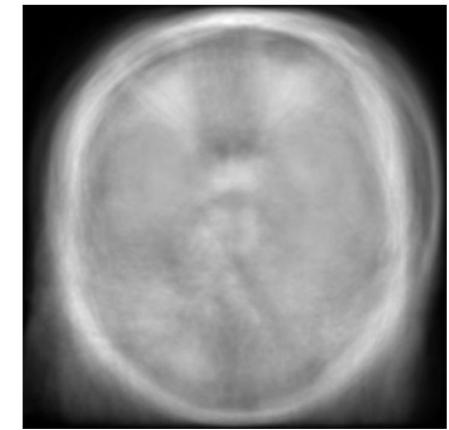
Glioma



Meningioma



Pituitary



Next Steps

Supervised Learning:

Feed Forward Neural Network

Convolutional Neural Network *

Unsupervised Learning:

Gaussian Mixture Model

**Feedback
welcome!**