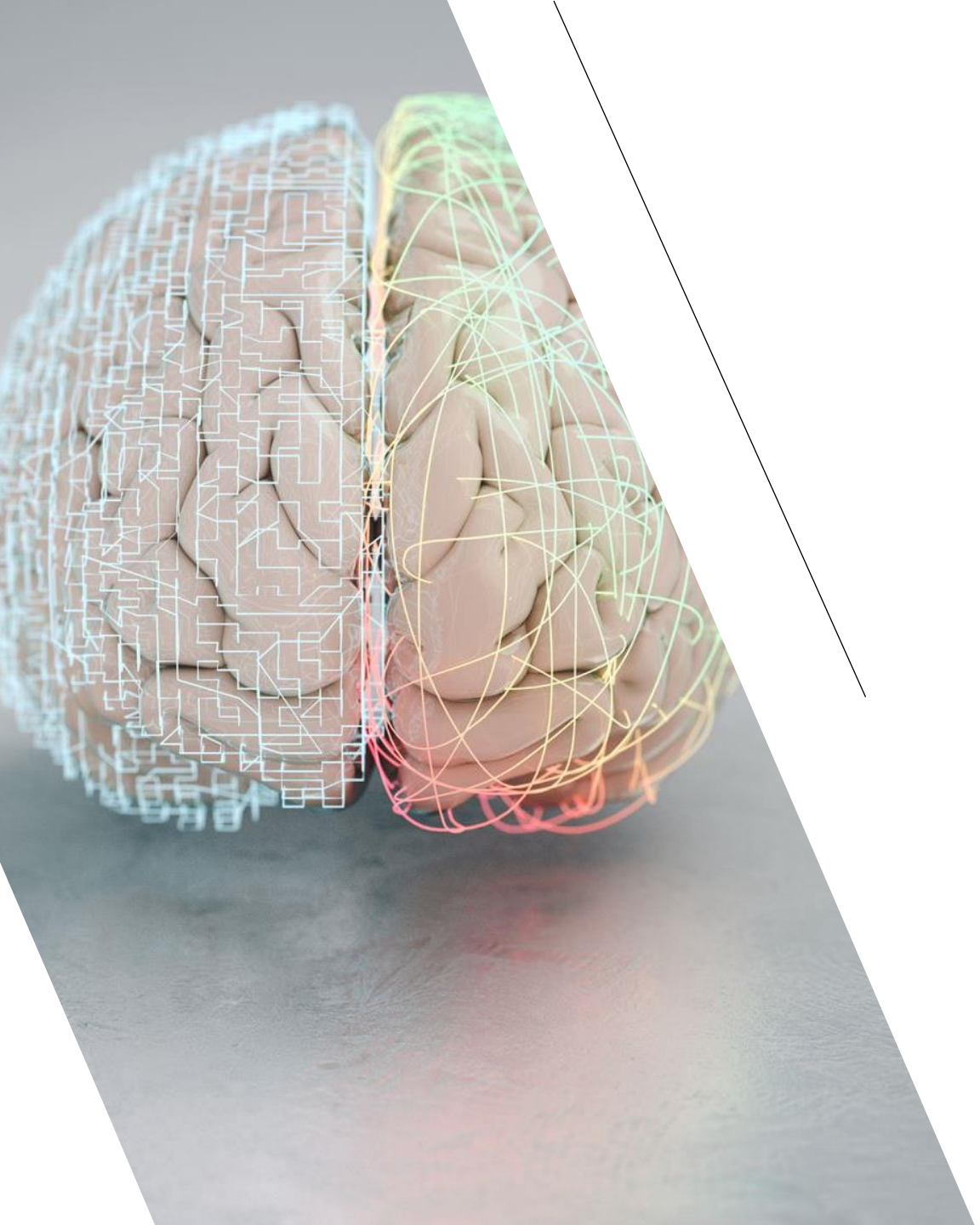


BRAIN TUMOR CLASSIFICAION WITH DEEP NEURAL NETWORKS

Amirhossein Ebrahimi

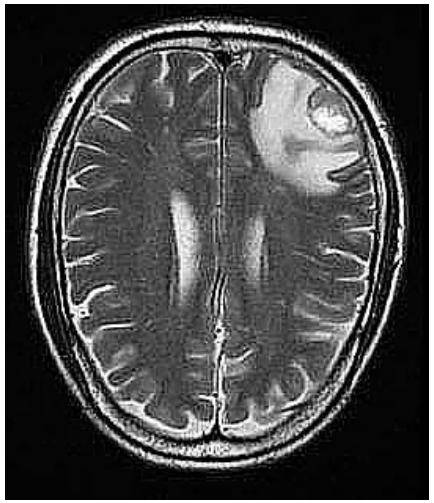
Fall 2024

Havaei et al.

TABLE OF CONTENT

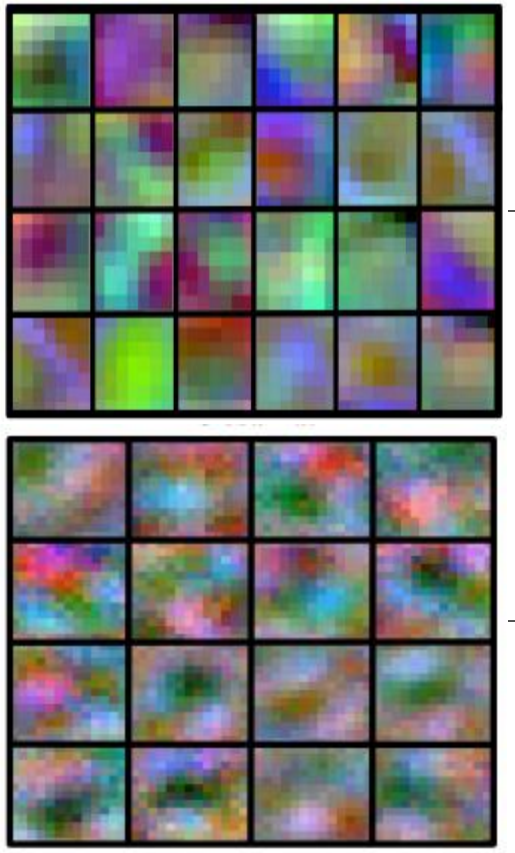
METHODOLOGY RECAP

TWO-PATHWAY CONVOLUTIONAL NEURAL NETWORK (CNN)



Local – Kernel 3x3
Small receptive field

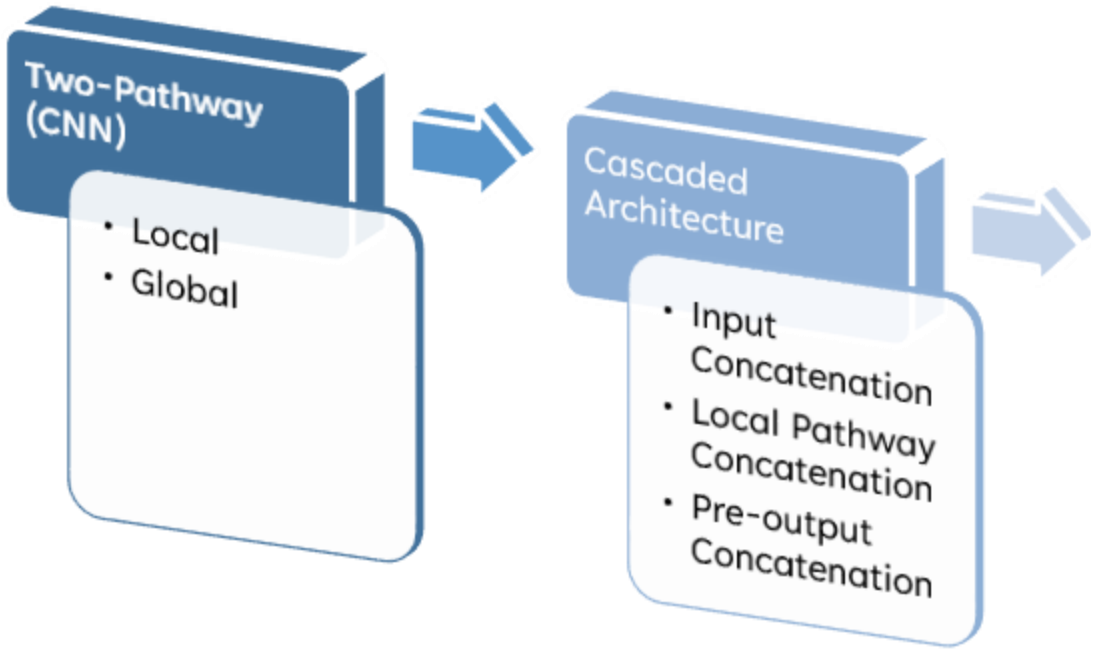
Global– Kernel 17x17
Larger receptive field



Concatenation

```
local_conv1 = tf.keras.layers.Conv2D(32, (7, 7), activation='relu', padding='same')(input_layer)
global_conv1 = tf.keras.layers.Conv2D(32, (13, 13), activation='relu', padding='same')(input_layer)
concatenated = tf.keras.layers.concatenate([local_conv1, global_conv1])
```

METHODOLOGY RECAP

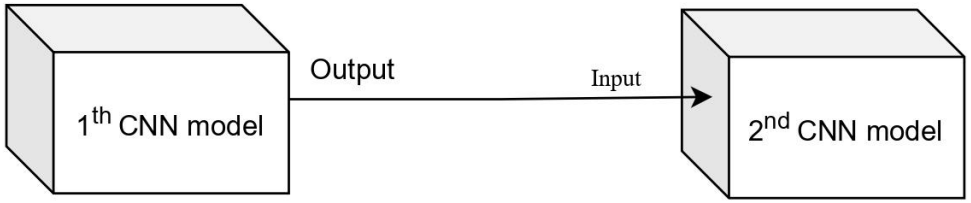


CASCADED ARCHITECTURE

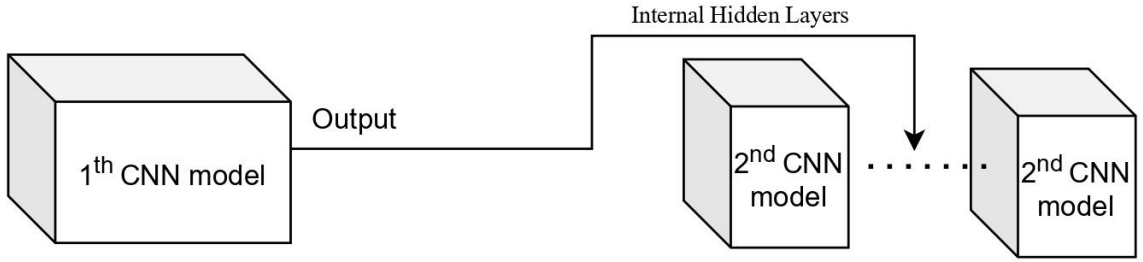
Adding the 1st CNN output as an 4th channel of 2nd CNN input

In this phase

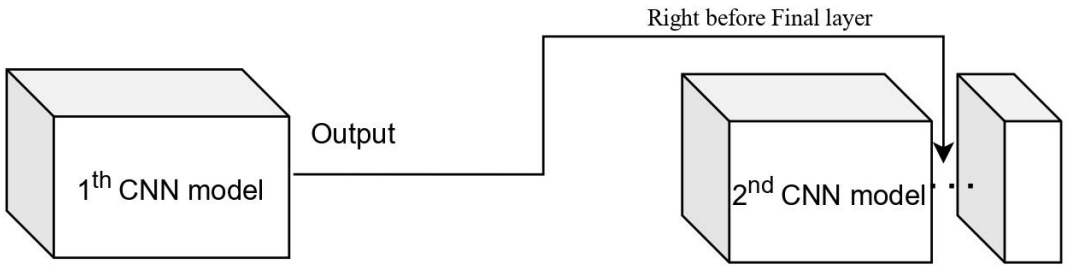
Input Cascade:



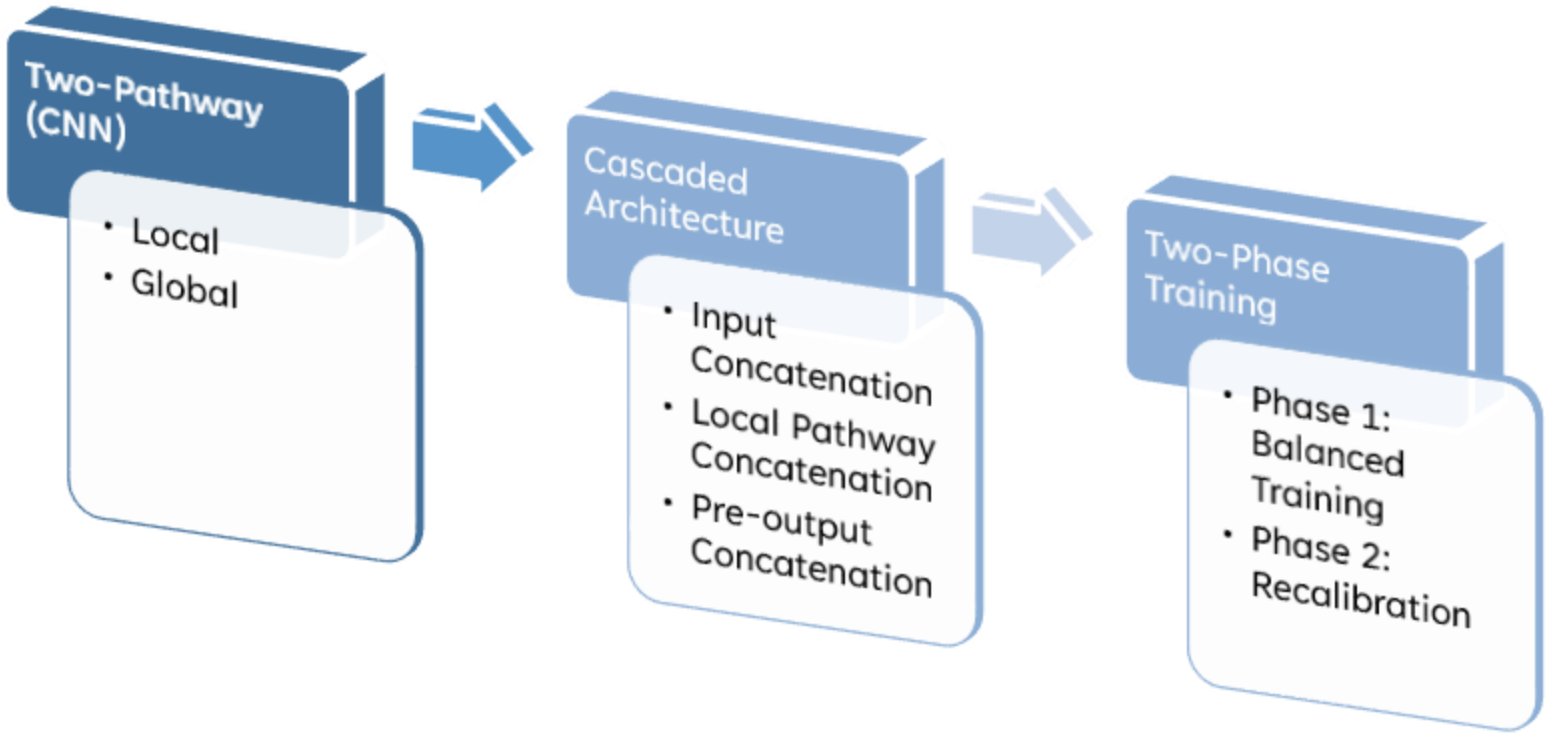
Local Pathway Cascade:



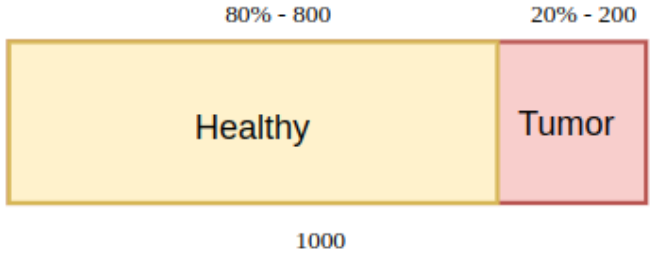
Pre-Output Cascade:

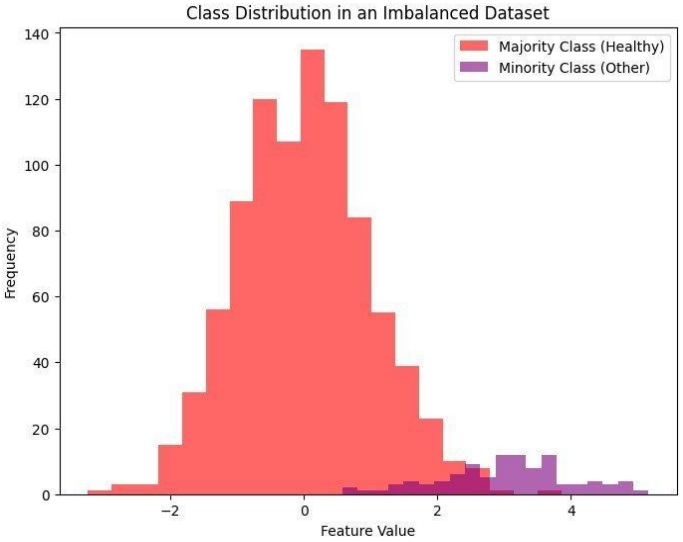


METHODOLOGY RECAP



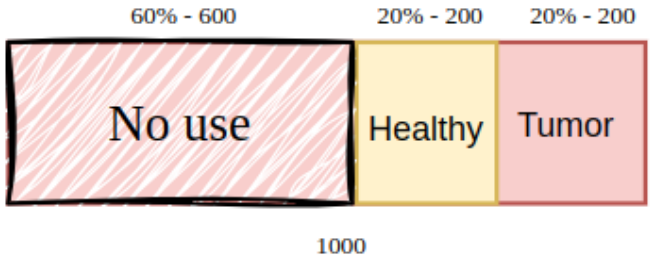
TWO-PHASE TRAINING



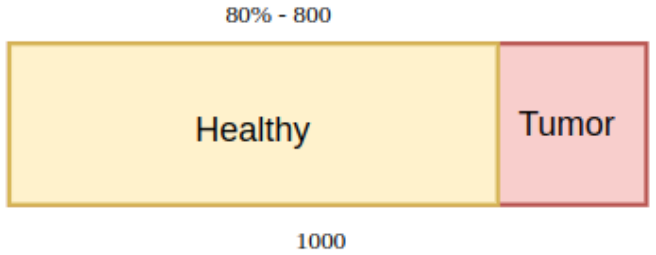


Phase 1_oversampling

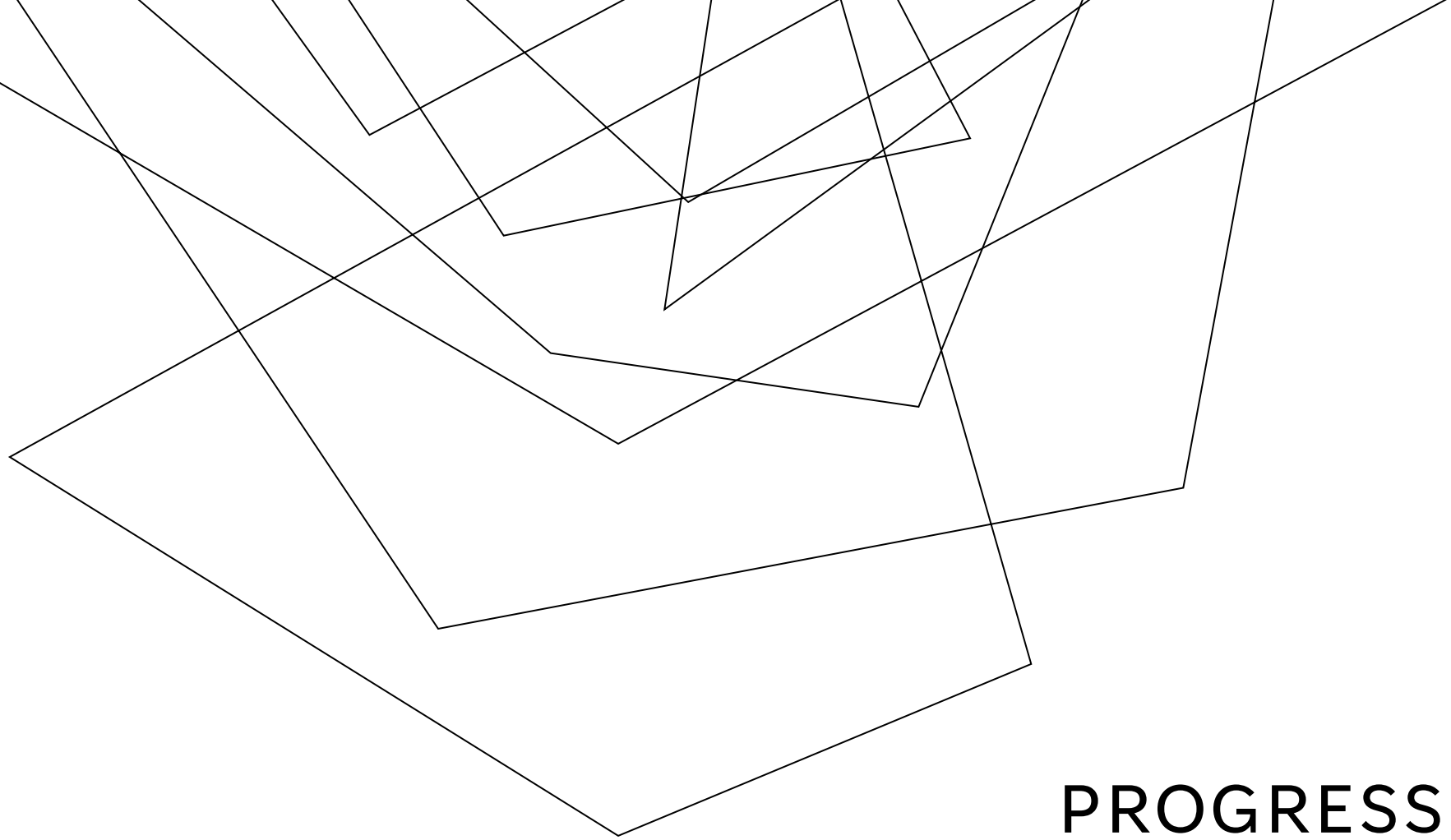
Sees a roughly equal number of examples from each class



Phase 2

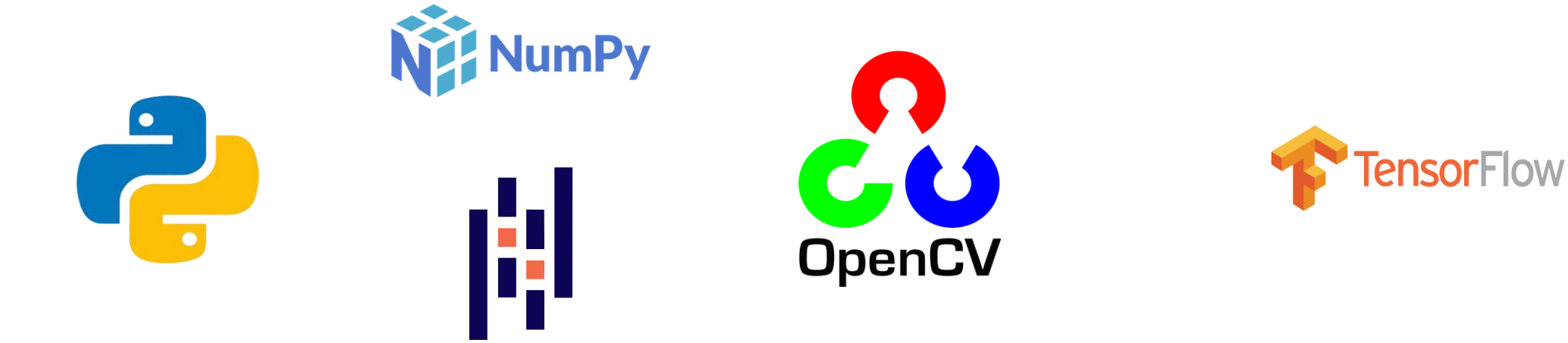


Fine-tune using pre-trained model from phase 1



PROGRESS

IMPLEMENTATION DETAILS



NumPy/Pandas: python library for dealing with large dimensional arrays/matrices

OpenCV: Python library for image processing

TensorFlow: Deep learning library for training neural networks

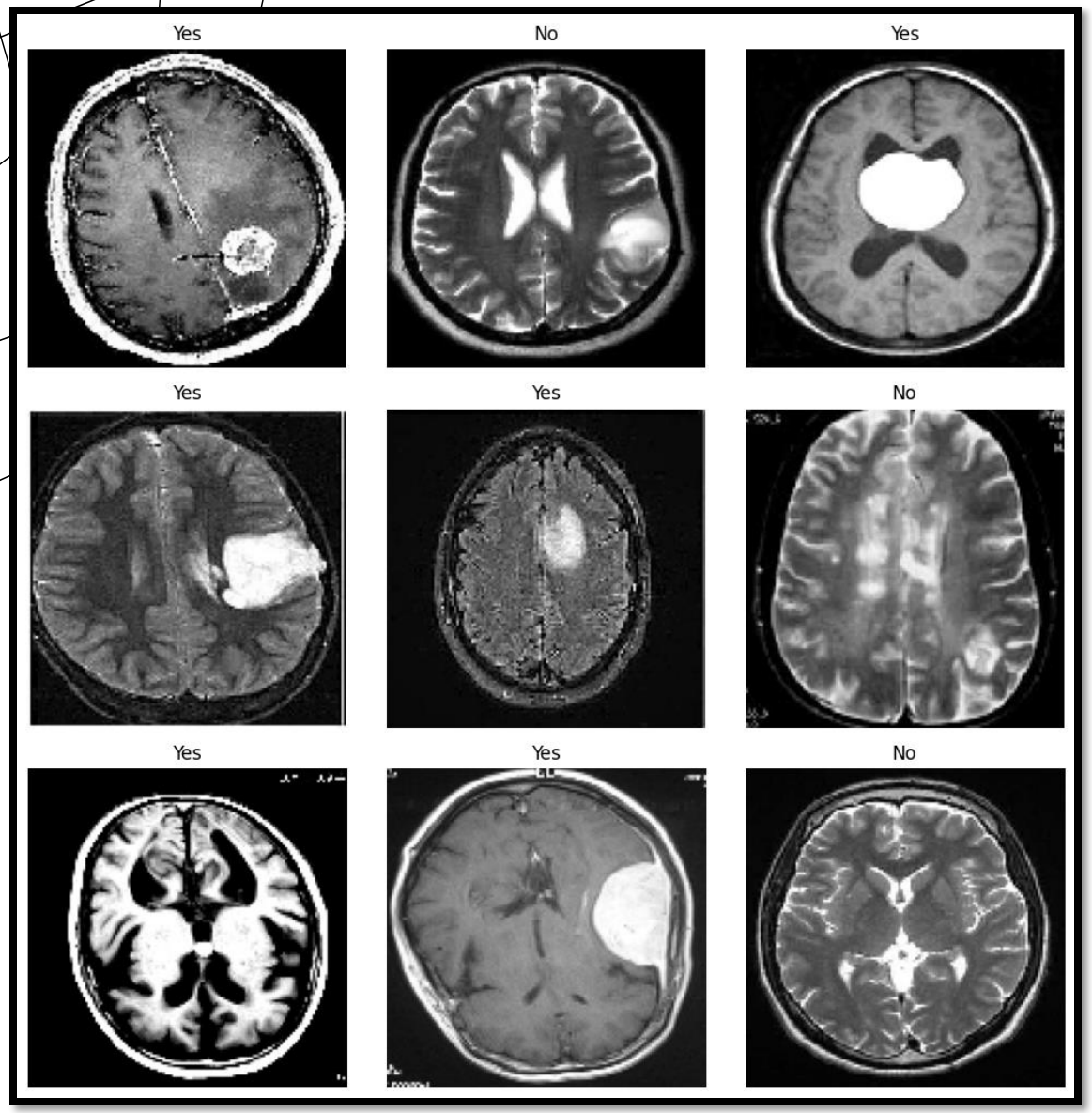
WHAT HAVE BEEN DONE

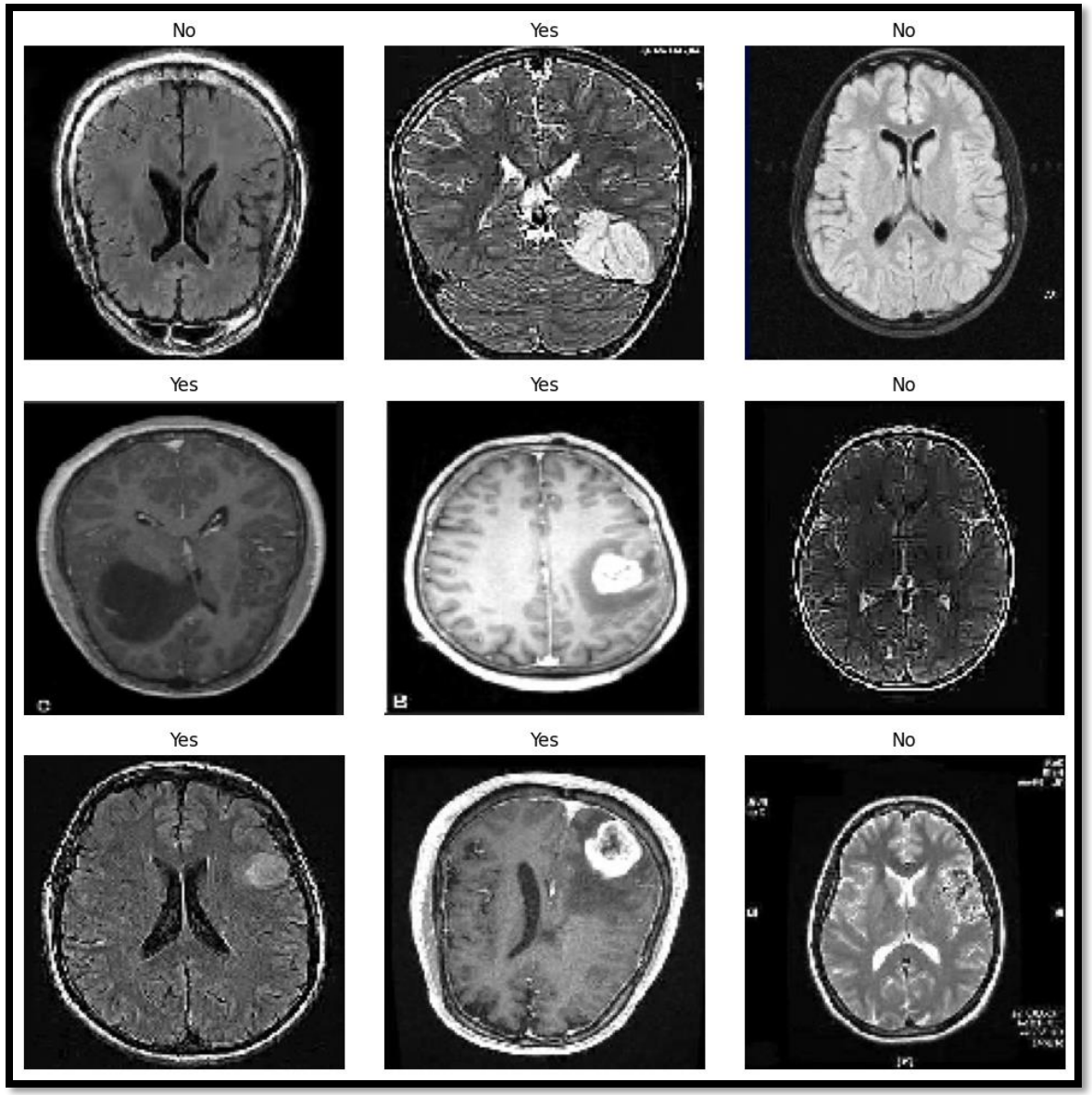
- Implemented their code from scratch
- Data Augmentation
- Added some preprocessing
- Generated useful feature for extra channel using KNN
- Trained a shallow CNN for classification

Code available at: https://colab.research.google.com/drive/1lVvElEqtCqJH5GRDFuxJ899Fn3fR-ct2?usp=sharing

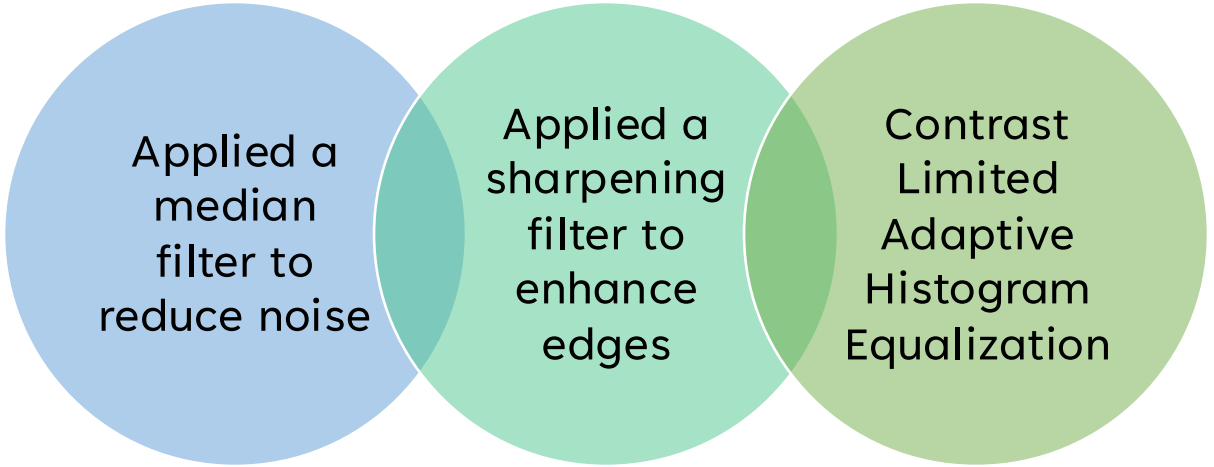
Data visualization

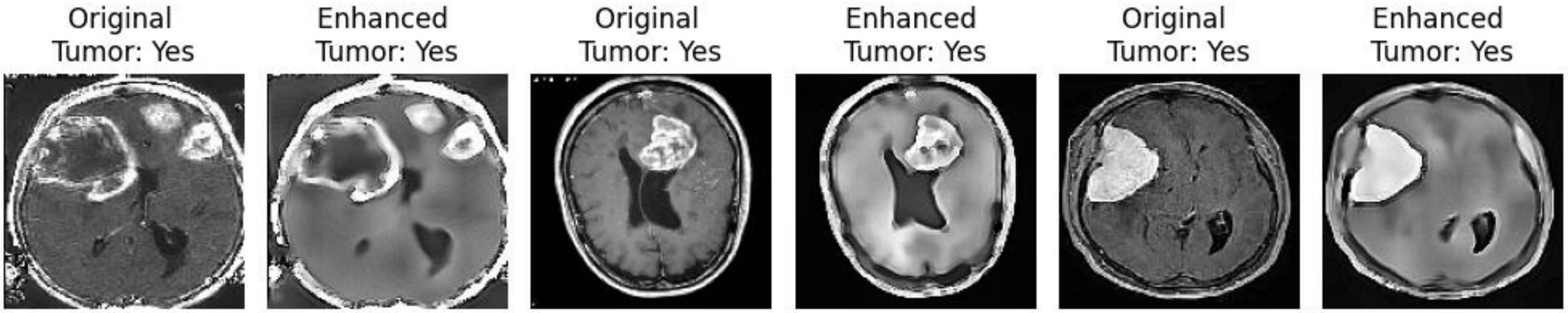
BRATS2019





PREPROCESSING

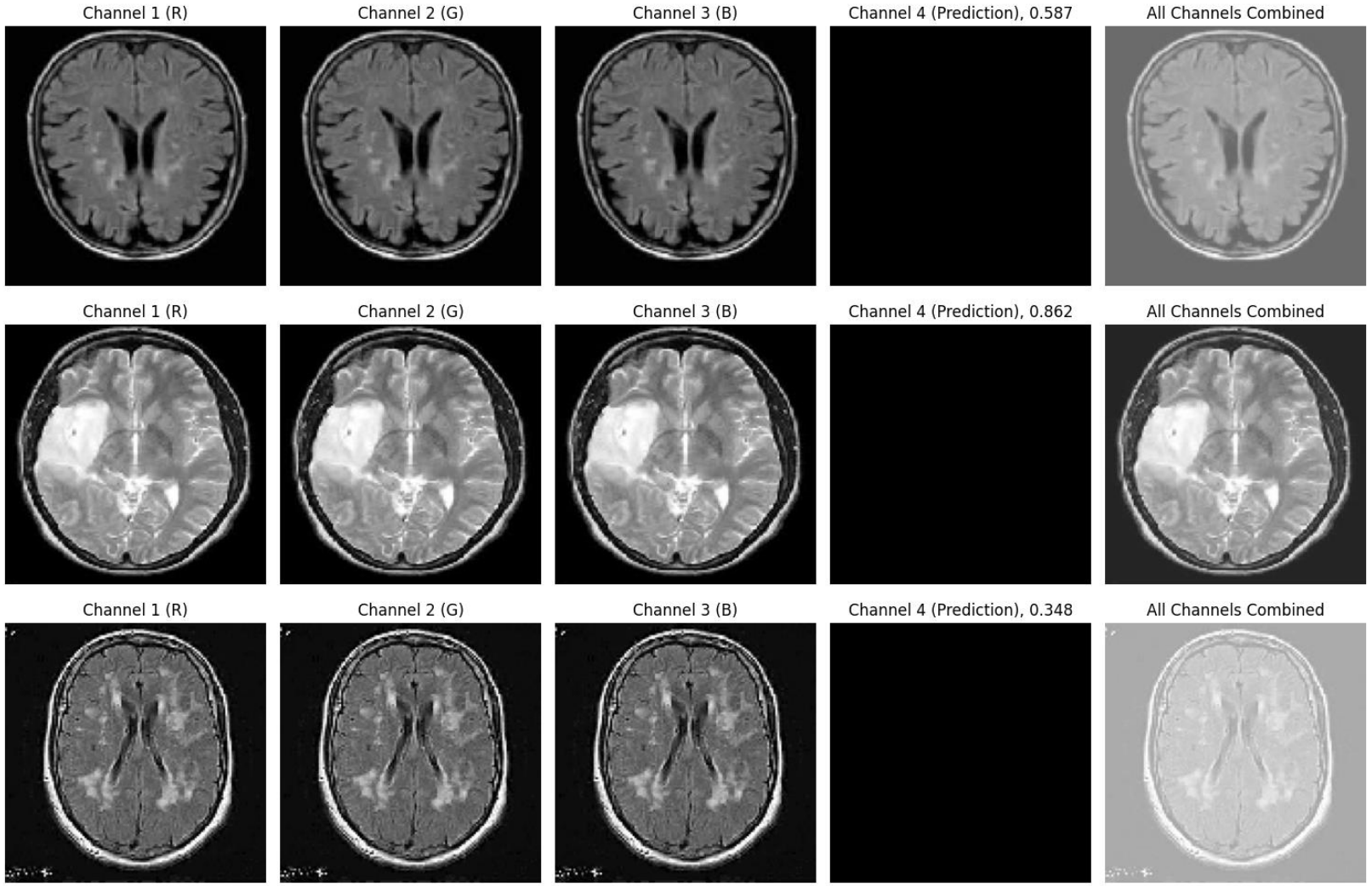




ENHANCEMENT

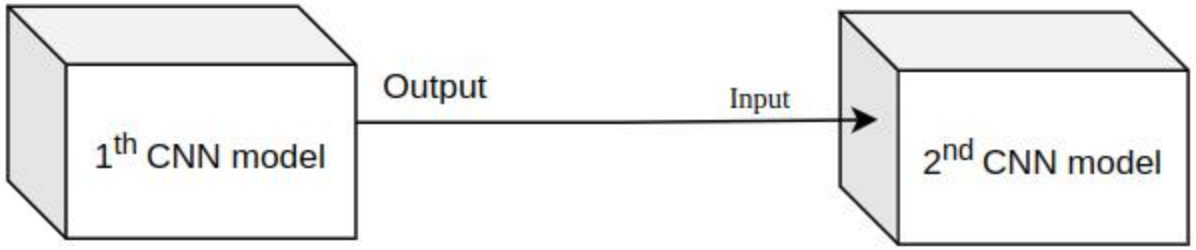
Why RGB when it's grayscale?

Meaningless feature at 4th channel

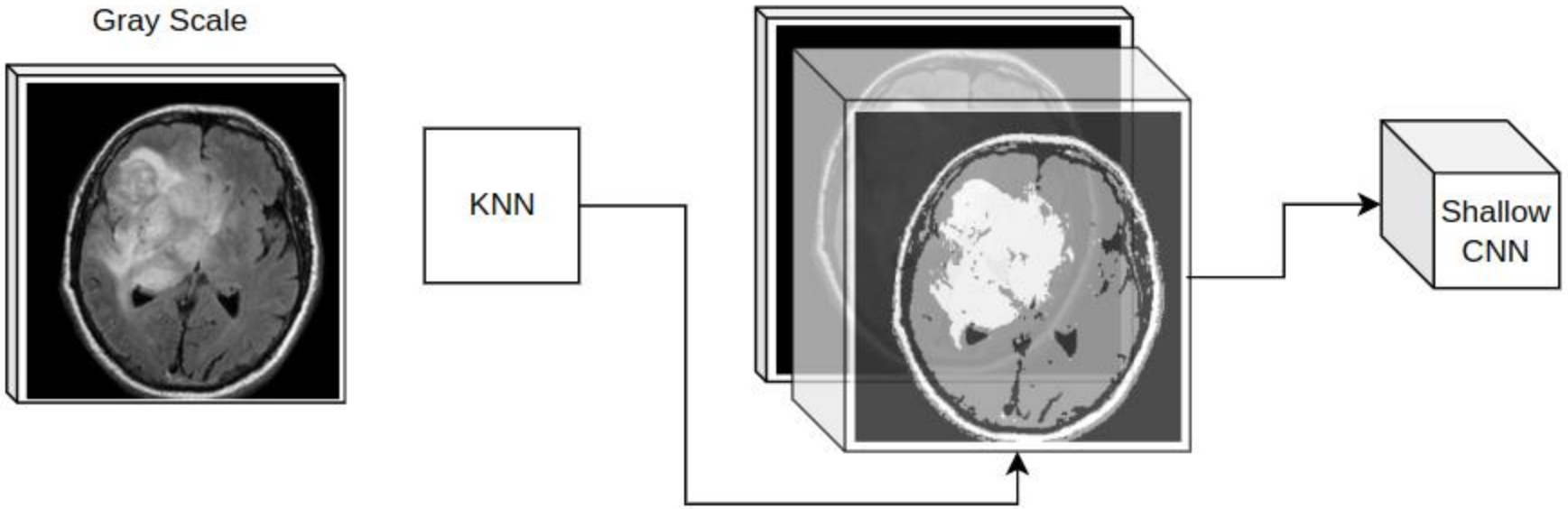


ENHANCEMENT

Input Cascade:

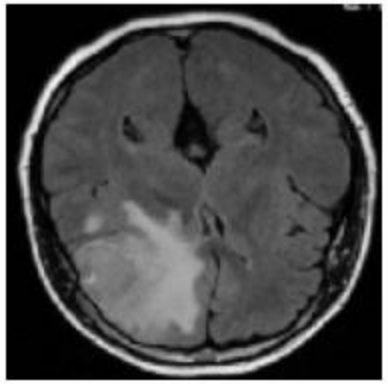


My Method

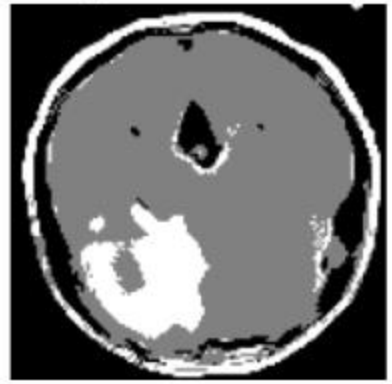


ENHANCEMENT

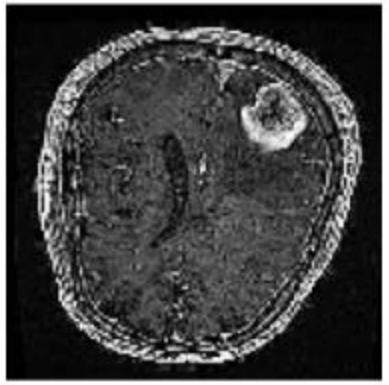
Channel 1



Channel 2 (Generated Mask feature)

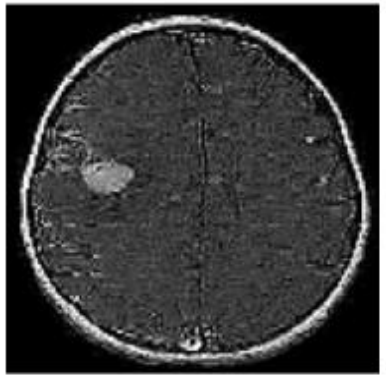


Channel 1

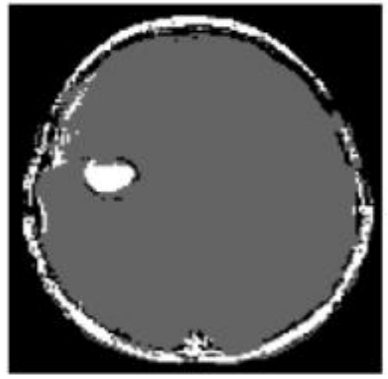


Channel 2 (Generated Mask feature)

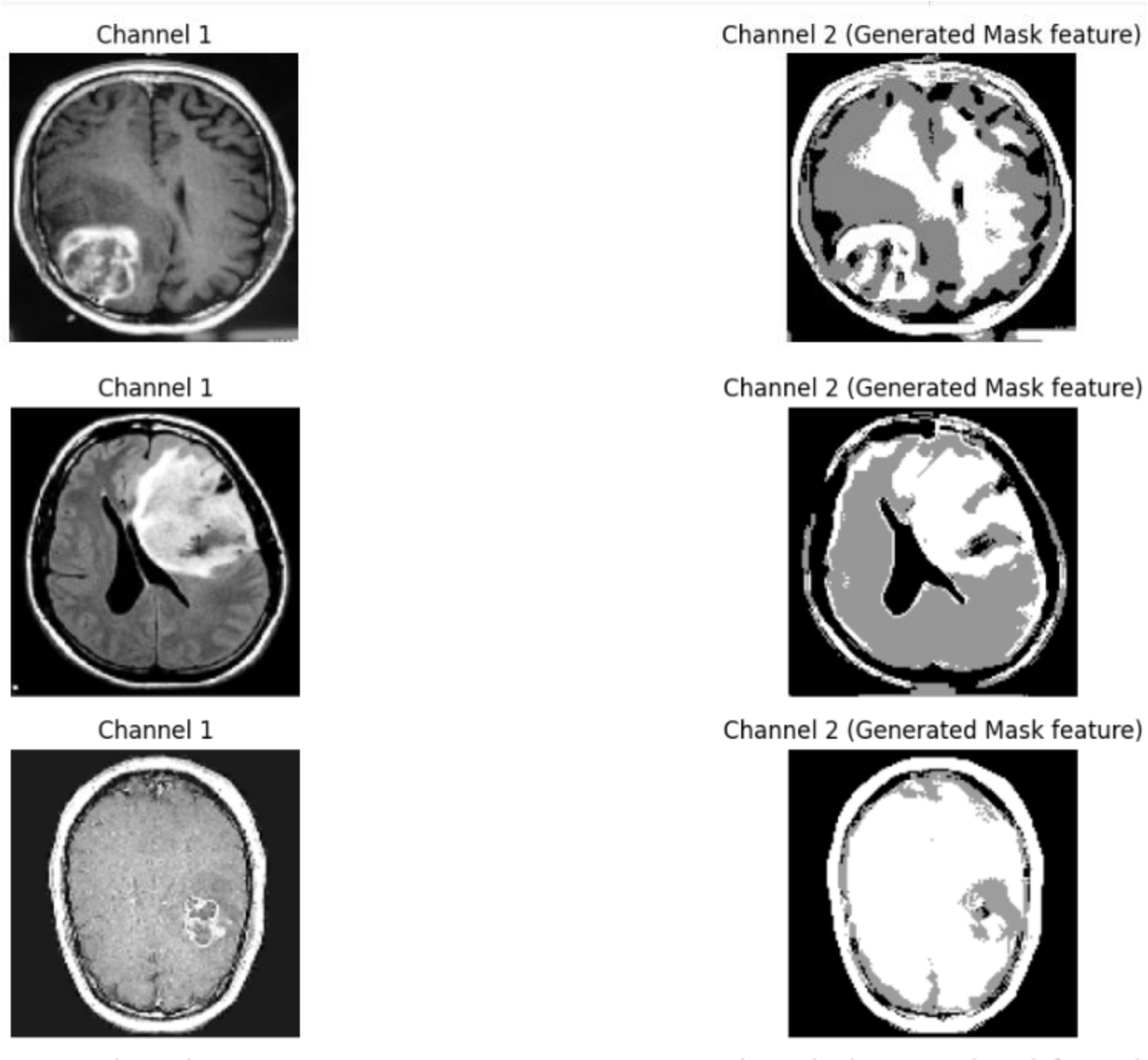
Channel 1



Channel 2 (Generated Mask feature)

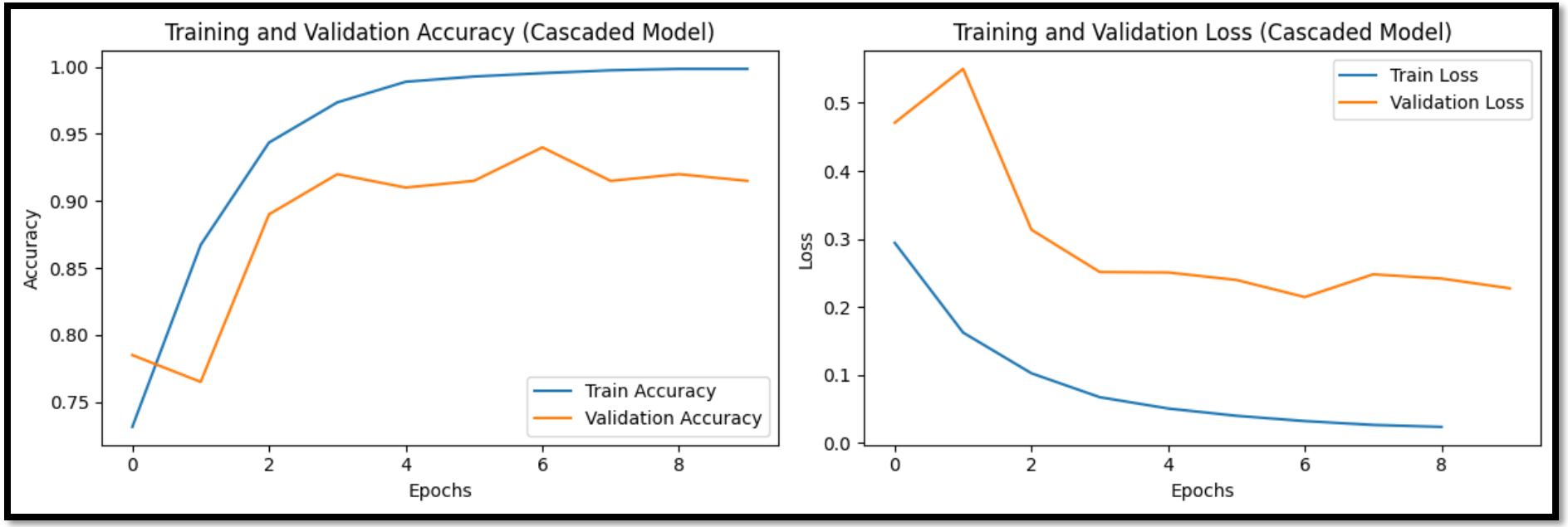


ENHANCEMENT



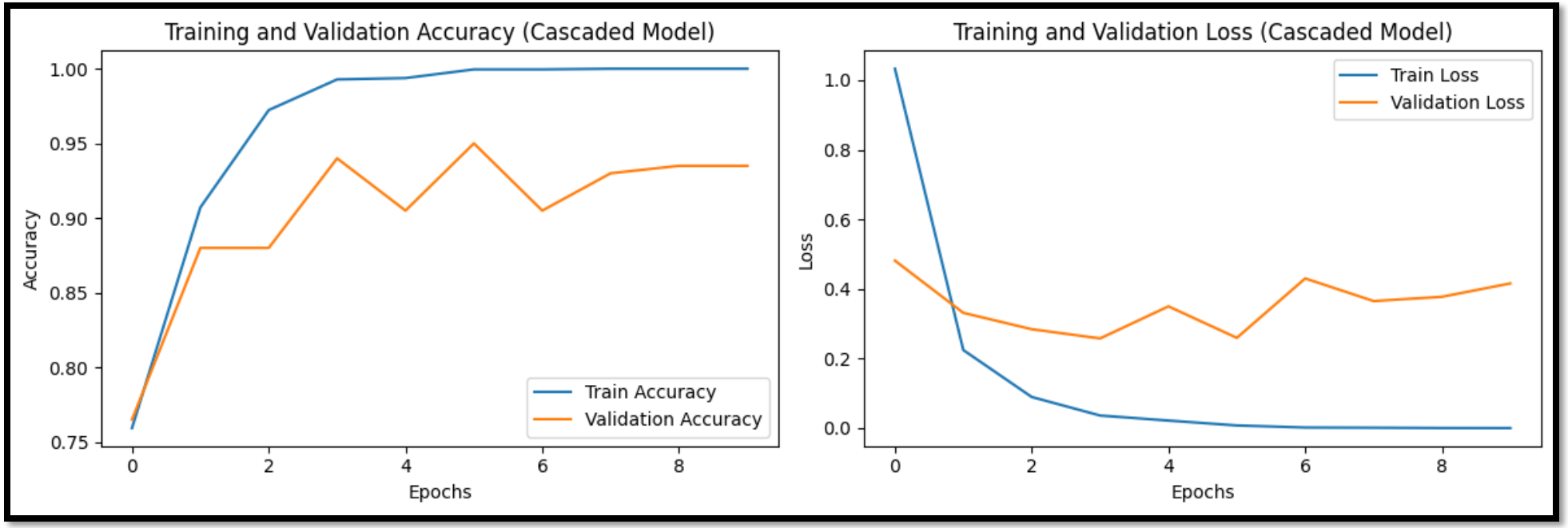
RESULTS

MY RESULTS



Model	Parameters	Exec Time
Shallow CNN	Total params: 67,152,708	45s
Total	67M	45s

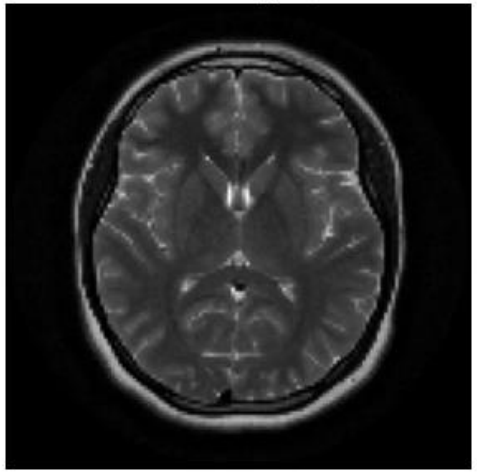
PAPER RESULTS



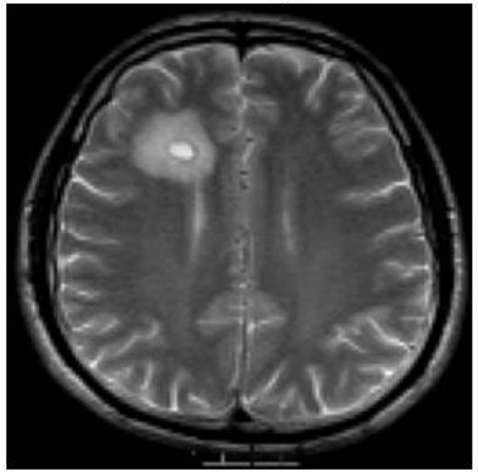
Model	Parameters	Exec Time
Cascaded model	Total params: 100,738,469	3m
first_stage model	Total params: 8,594,689	1m
Total	110M	4m + two-phase training

MY RESULTS

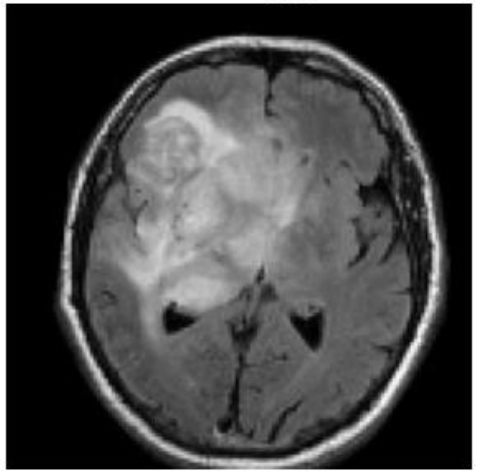
Prediction: No (1.00)
File: N2.jpeg



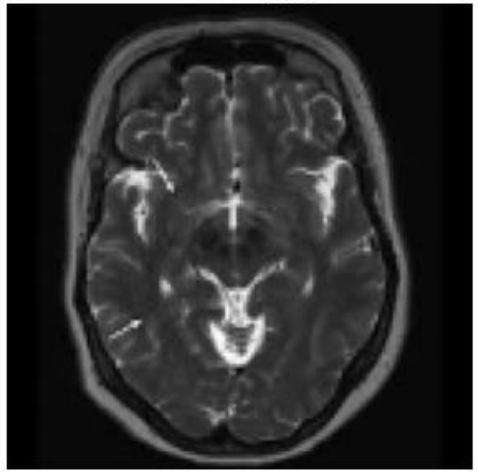
Prediction: Yes (1.00)
File: Y3.jpg



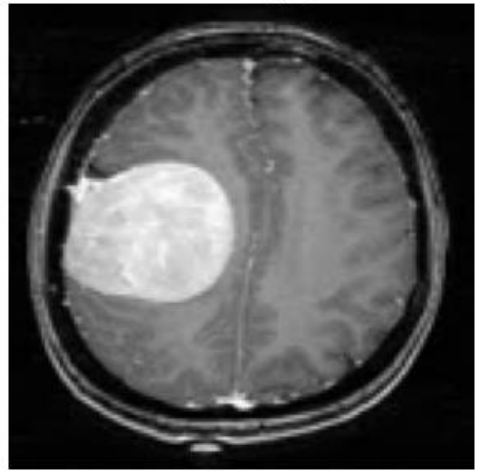
Prediction: Yes (0.96)
File: Y2.jpg



Prediction: No (1.00)
File: N3.jpg

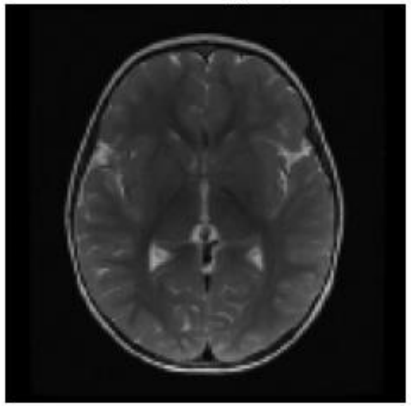


Prediction: Yes (1.00)
File: Y1.jpg

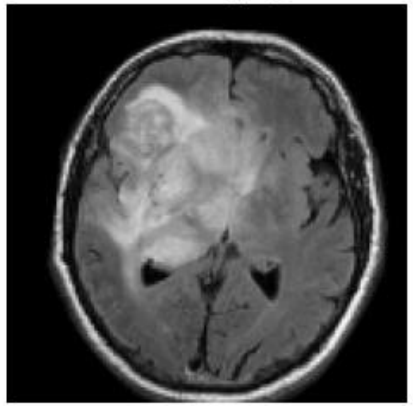


PAPER RESULTS

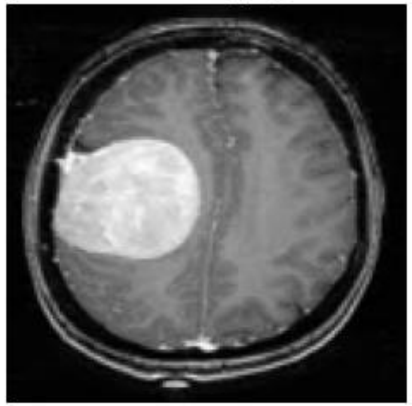
Prediction: No (1.00)
File: N1.jpeg



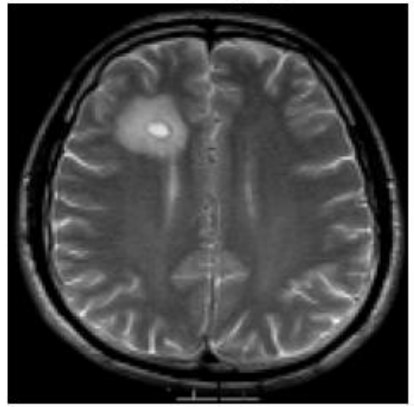
Prediction: Yes (1.00)
File: Y2.jpeg



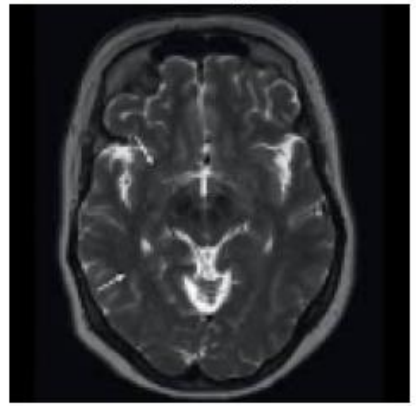
Prediction: Yes (1.00)
File: Y1.jpeg



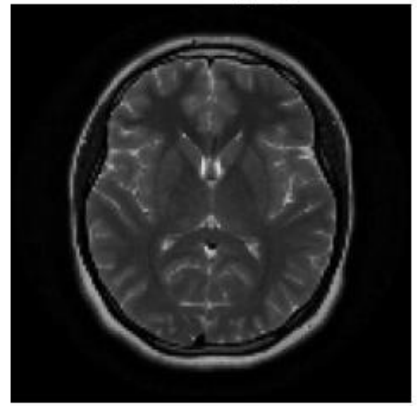
Prediction: Yes (1.00)
File: Y3.jpeg



Prediction: No (1.00)
File: N3.jpeg



Prediction: No (1.00)
File: N2.jpeg



CONCLUSION

- **Goal:** Can we achieve the same accuracy with simpler method?
- **Answer:** YES
  - More Computationally efficient
  - Less Execution time
  - Feeding meaningful feature (KNN)
    - Much more efficient for segmentation task

THANKS FOR YOUR ATTENTION