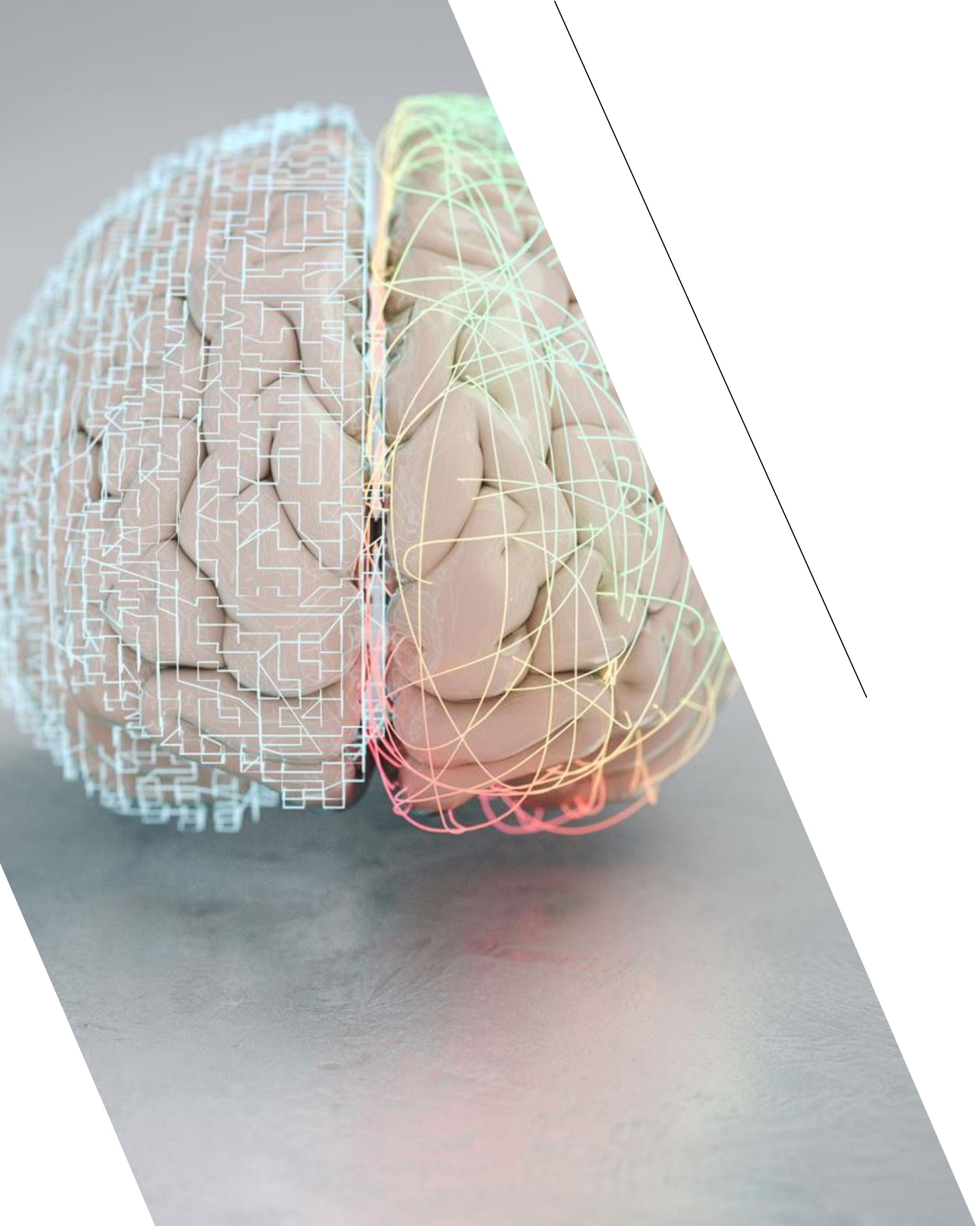


BRAIN TUMOR SEGMENTATION WITH DEEP NEURAL NETWORKS

Amirhossein Ebrahimi

Fall 2024

Havaei et al.

TABLE OF CONTENT

Introduction

Methodology

Plan and Progress

Potential Improvement

INTRODUCTION

- **Background**

  Accurate segmentation is crucial for **diagnosis, growth rate prediction**, and **treatment planning**.

  - It is challenging to segment due to variability in **shape, size, and contrast**.
  - Traditional segmentation approaches rely on handcrafted features, which is not as accurate and is time-consuming.

- **Goal**

  Develop an accurate and a fully automatic tumor segmentation method using DNN.

- **Novelty?**

  Next slides.

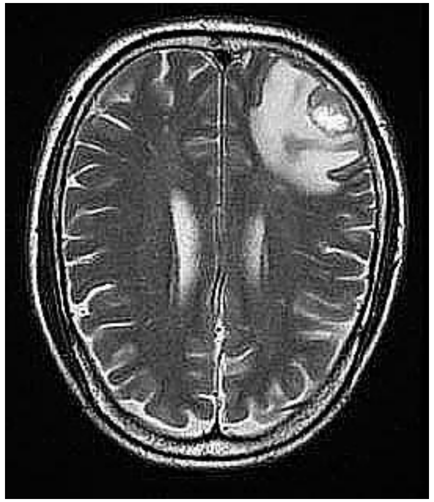
METHODOLOGY

- **Two-Pathway Convolutional Neural Network (CNN):**
  - Local
  - Global

- **Cascaded Architecture:**
  - Input Concatenation
  - Local Pathway Concatenation
  - Pre-output Concatenation

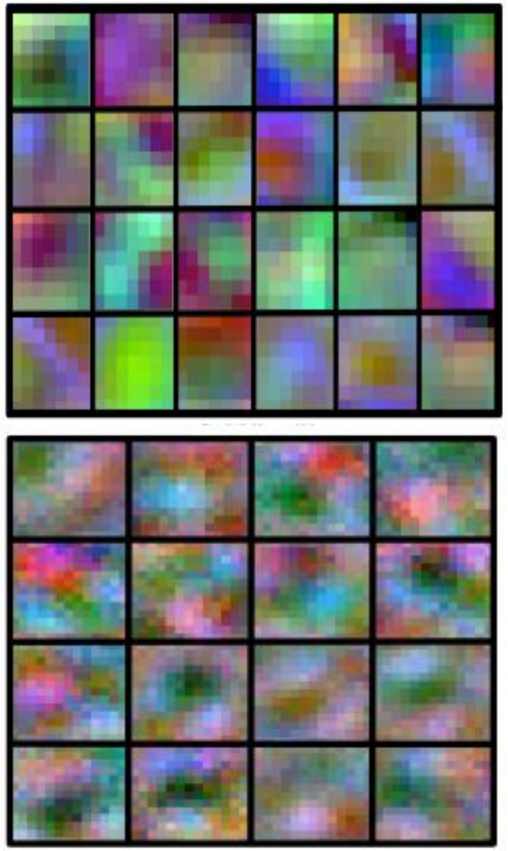
- **Two-Phase Training:**
  - *Phase 1:* Balanced Training
  - *Phase 2:* Recalibration

TWO-PATHWAY CONVOLUTIONAL NEURAL NETWORK (CNN)



Local – Kernel 3x3
Small receptive field

Global– Kernel 17x17
Larger receptive field



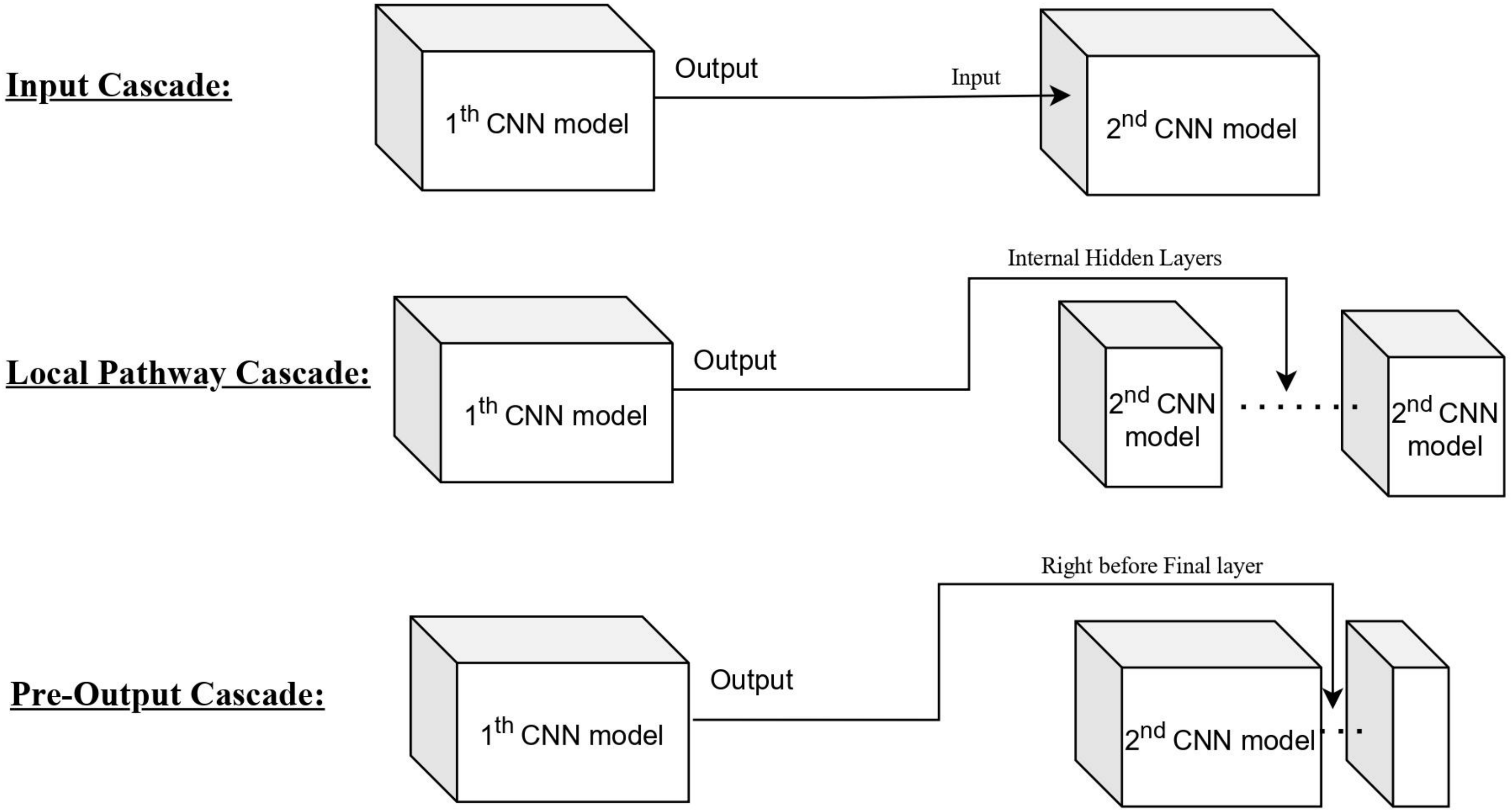
Concatenation

```
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])
```

CASCADED ARCHITECTURE

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

In this phase

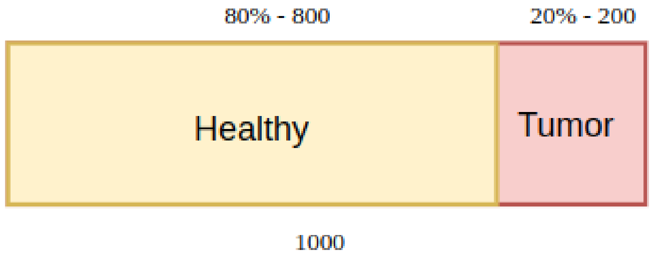


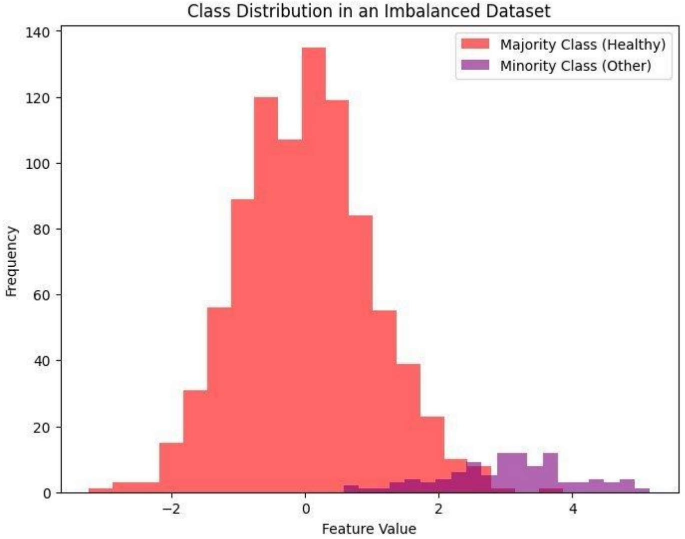
Input Cascade:

Local Pathway Cascade:

Pre-Output Cascade:

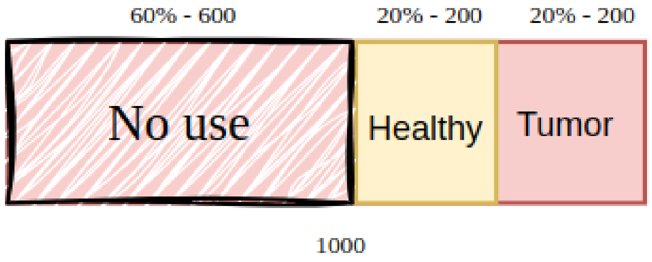
TWO-PHASE TRAINING



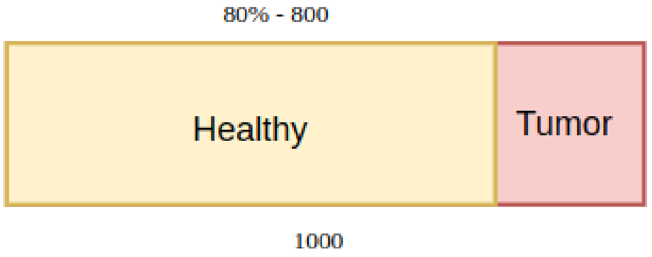

Class Distribution in an Imbalanced Dataset

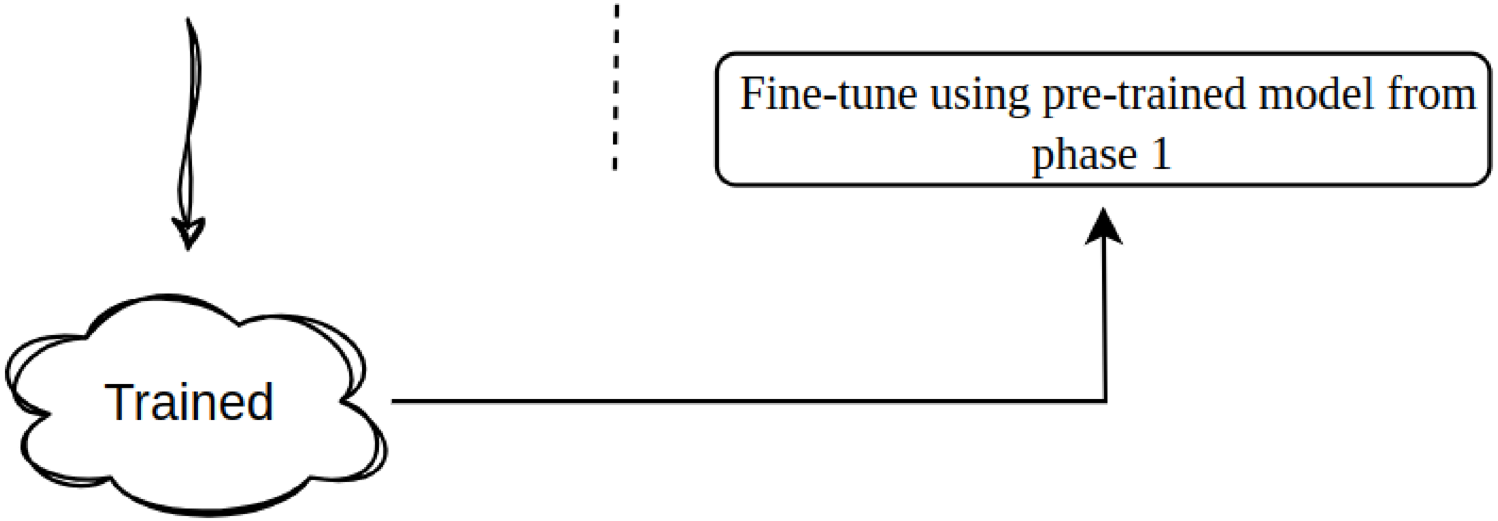
Phase 1_oversampling

Sees a roughly equal number of examples from each class



Phase 2





Fine-tune using pre-trained model from phase 1

PLAN AND PROGRESS

What have I done so far

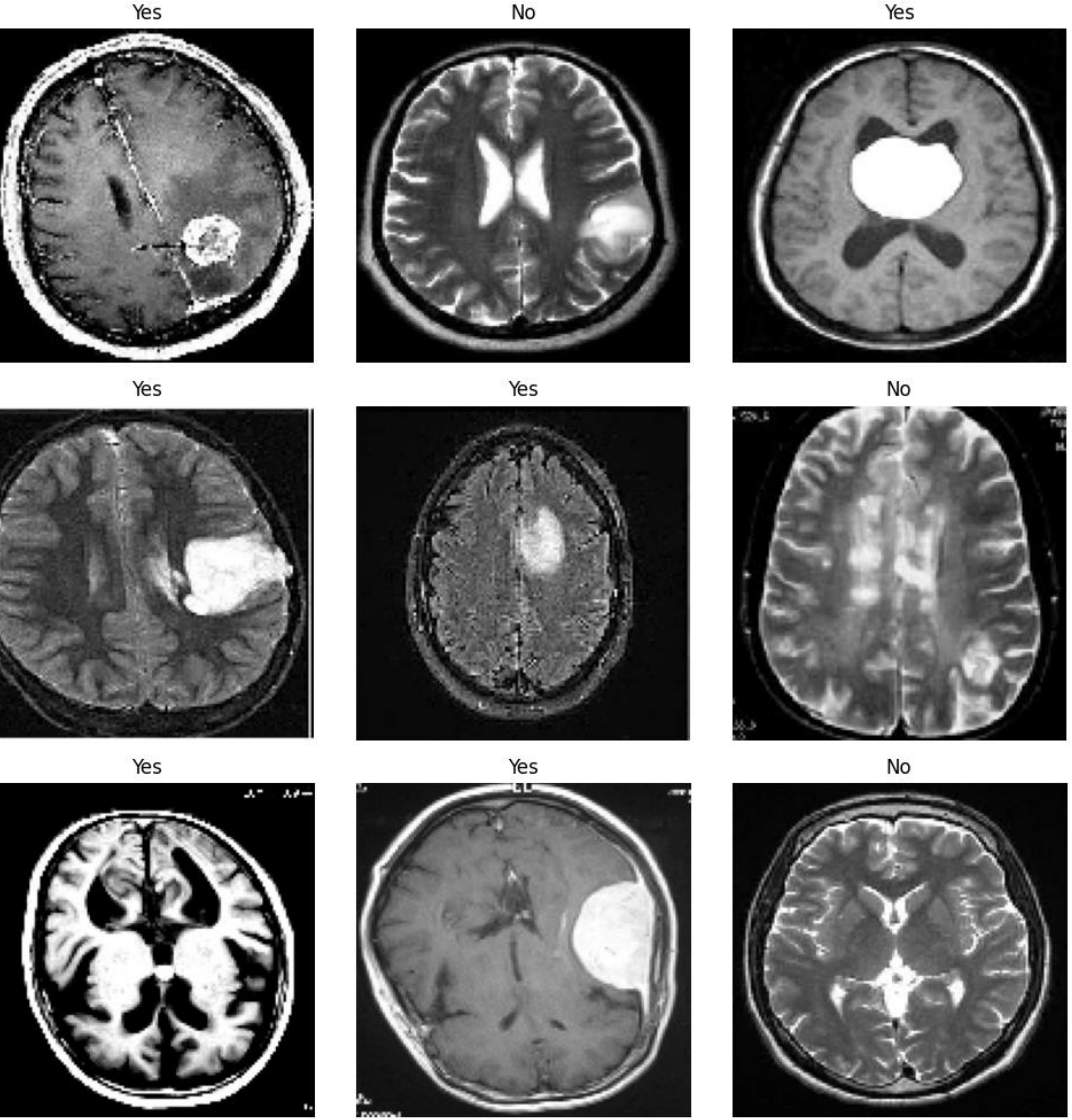
- Implemented their code [Colab]
- Added some preprocessing

- Using Classification Brats2019 dataset
- Not used two-phase training

Future steps

- Using Segmentation Brats2019 dataset
- Modify their model for segmentation task
- Add more preprocessing

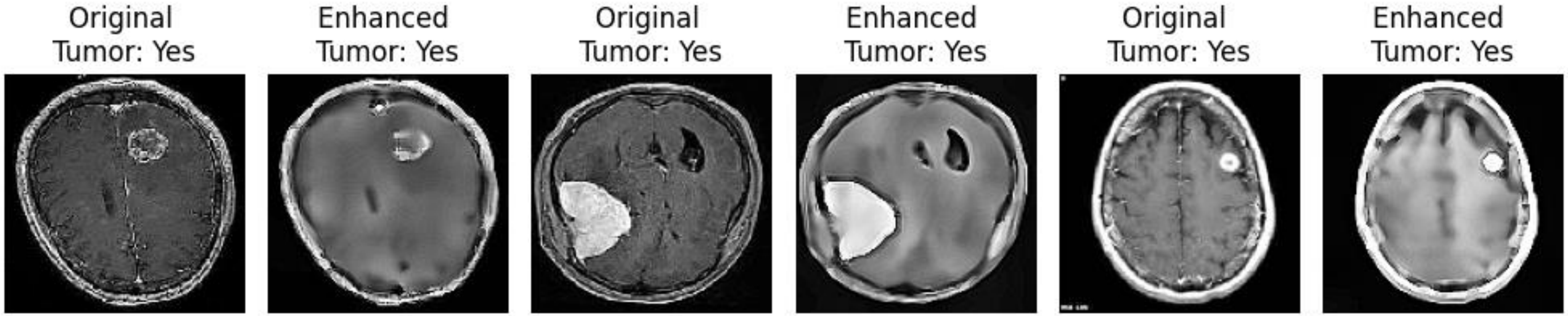
Current approach	Future steps
Cascaded Architecture	encode feature map instead of adding pixel-level classification at 4th channel
Cascaded Architecture	Using ResNet instead of Cascaded CNN

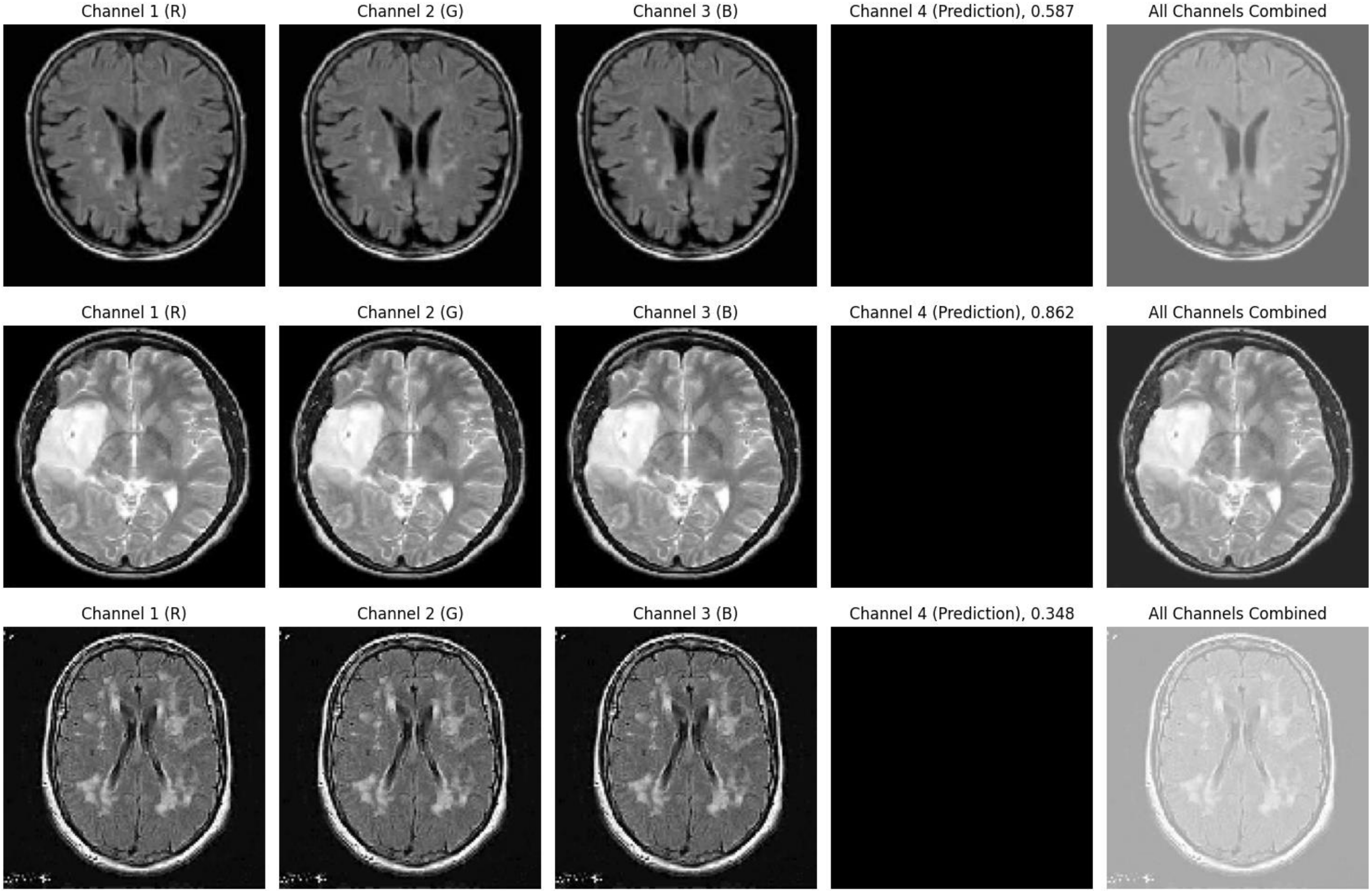


BRATS2019

What have I done so Far: *Preprocessing*

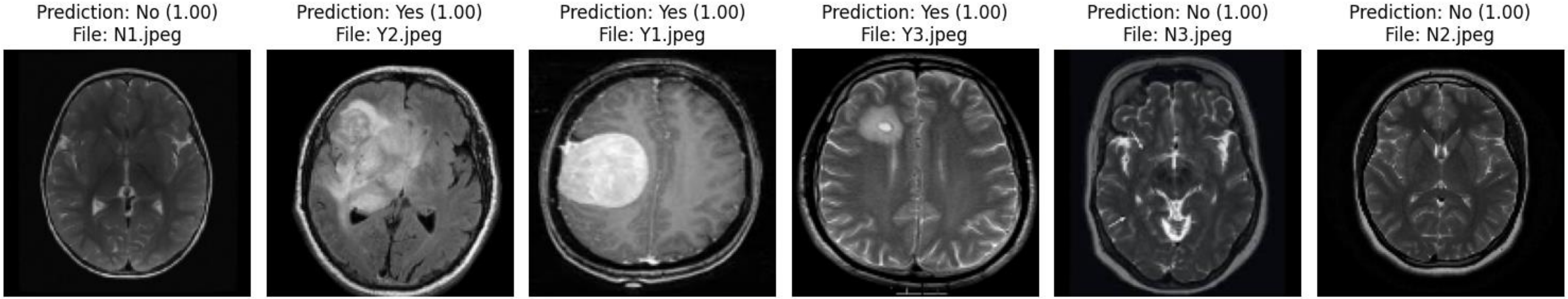
- Applied a median filter to reduce noise
- Applied a sharpening filter to enhance edges
- Contrast Limited Adaptive Histogram Equalization





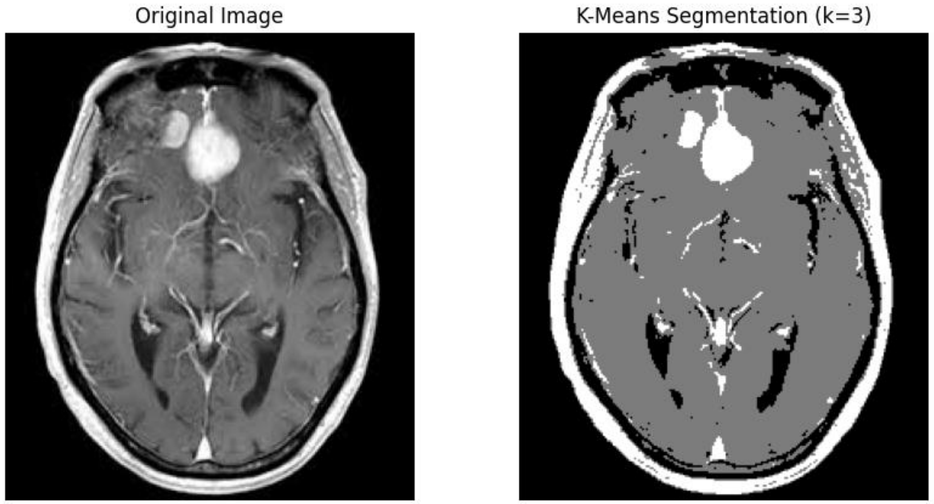
Channel 1 (R)
Channel 2 (G)
Channel 3 (B)
Channel 4 (Prediction), 0.587
All Channels Combined
Channel 1 (R)
Channel 2 (G)
Channel 3 (B)
Channel 4 (Prediction), 0.862
All Channels Combined
Channel 1 (R)
Channel 2 (G)
Channel 3 (B)
Channel 4 (Prediction), 0.348
All Channels Combined

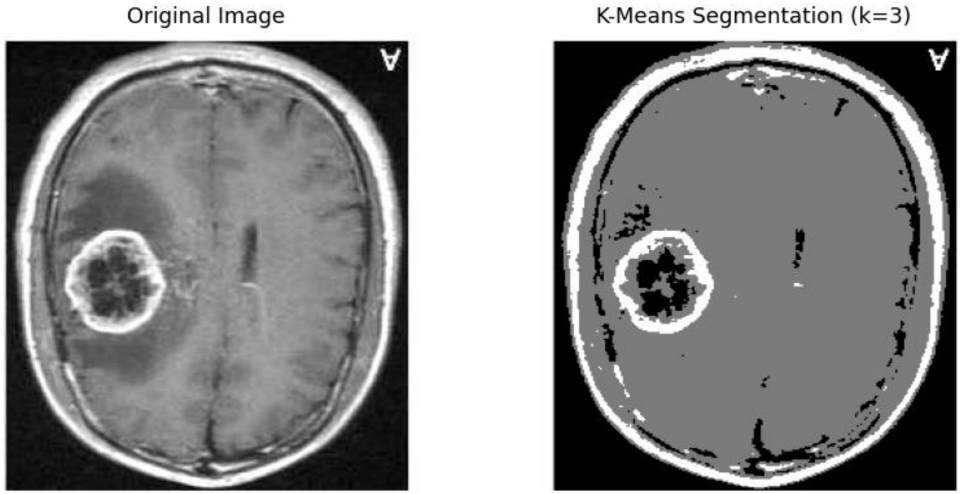
What have I done so Far: *Results*

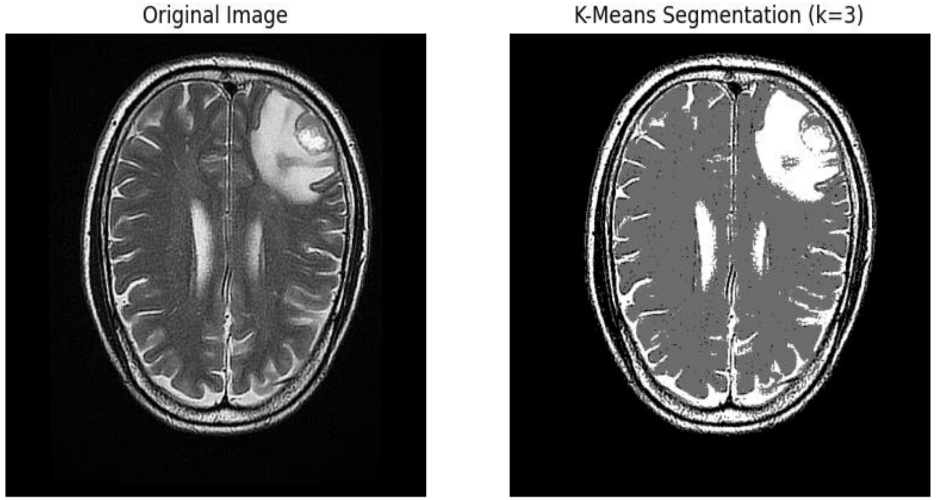


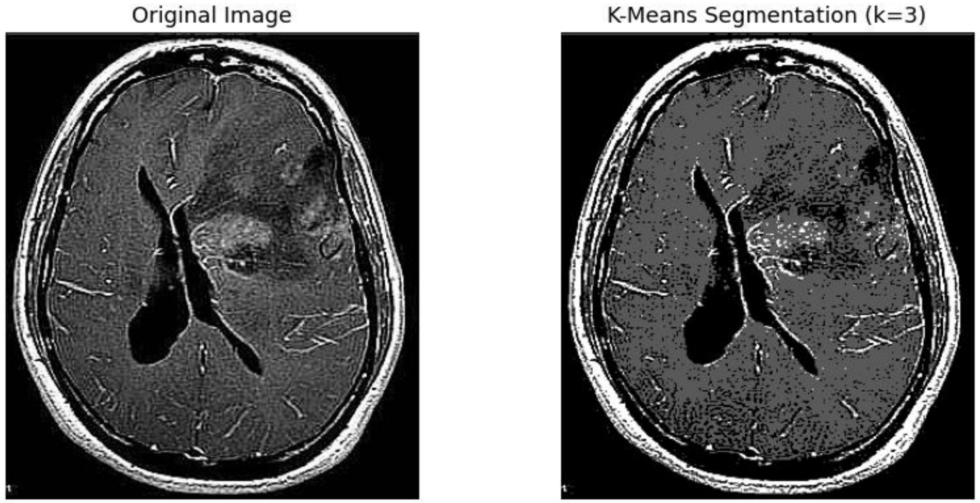
What have I done so Far:

Simple KNN?









THANKS FOR YOUR ATTENTION