Brain Tumor Detection using Magnetic Resonance Imaging and Machine Learning

CMPE 351 | Machine Learning Design Project
Final Presentation

Group 6: Madison Boem, Caroline Kim, Eric Venditti, Jade Watson
April 13th, 2022

Table of Contents

01 Introduction

Background

Overview

Early diagnosis of brain tumors is important for patient prognosis

Magnetic resonance imaging (MRI) is used for glioma diagnosis

Currently-adopted practices

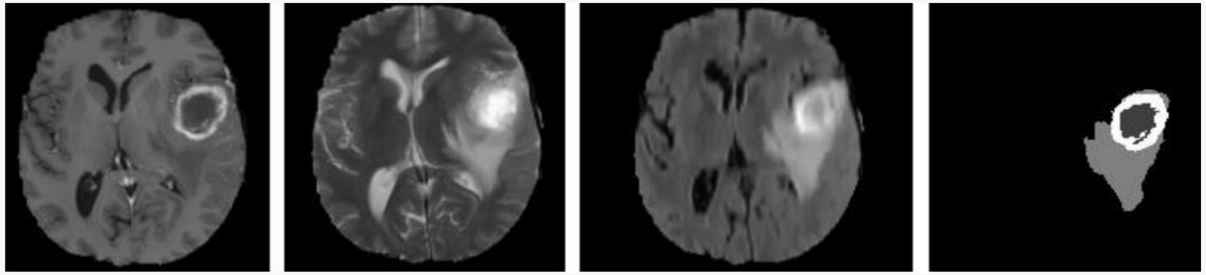
Manual segmentation of brain tumors by radiologists

LIMITATIONS

Time-consuming and resource-heavy

Inter- and intra-rater variability → fragile replicability

Complicated data



Background

Automatic segmentation

Convolutional neural networks (CNN) used for automatic segmentation

Focus on network architecture, e.g. 4D vs. 2D architectures

Current progress

Mature task propelled by BraTS competition

Research now focuses on reducing computational complexity and combining CNNs with other classification techniques

REMAINING CHALLENGES

Reducing computational complexity for usage in clinical settings

Increasing accuracy of segmentation due to irregularity of tumor shapes, sizes, and boundaries

Research Questions

RQ 1: **How can transfer learning be effectively utilized?**

RQ 2: **Can pre-processing be reduced and what are the implications?**

RQ 3: **Can model accuracy be sustained with lower computational complexity?**

02 Dataset

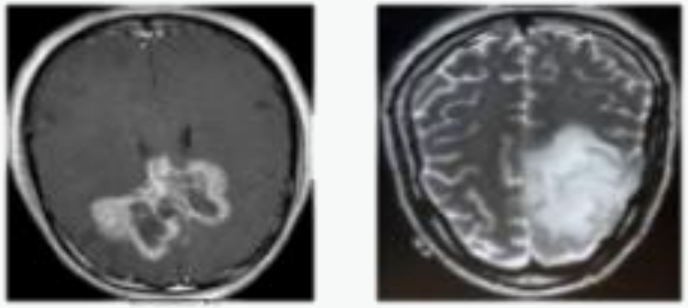
Dataset

Overview

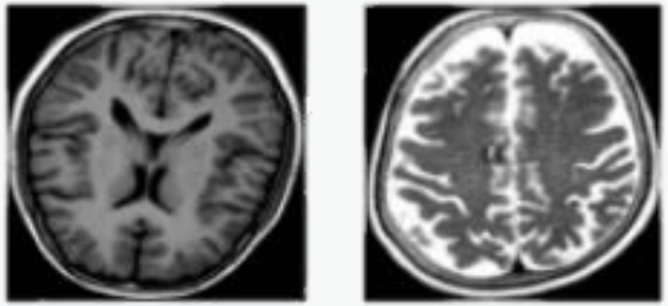
Open-source dataset from Kaggle provided by Navoneel Chakrabarty and his research in Brain MRI Images for Brain Tumor Detection

Contains 253 T1-weighted images; 98 healthy scans and 155 scans contain lesions

Since the dataset is small, data augmentation is used to create more images and solve the imbalance between tumorous and non-tumorous images



CONTAINS TUMOR
155 images



DOES NOT CONTAIN TUMOR
98 images

Summary Statistics

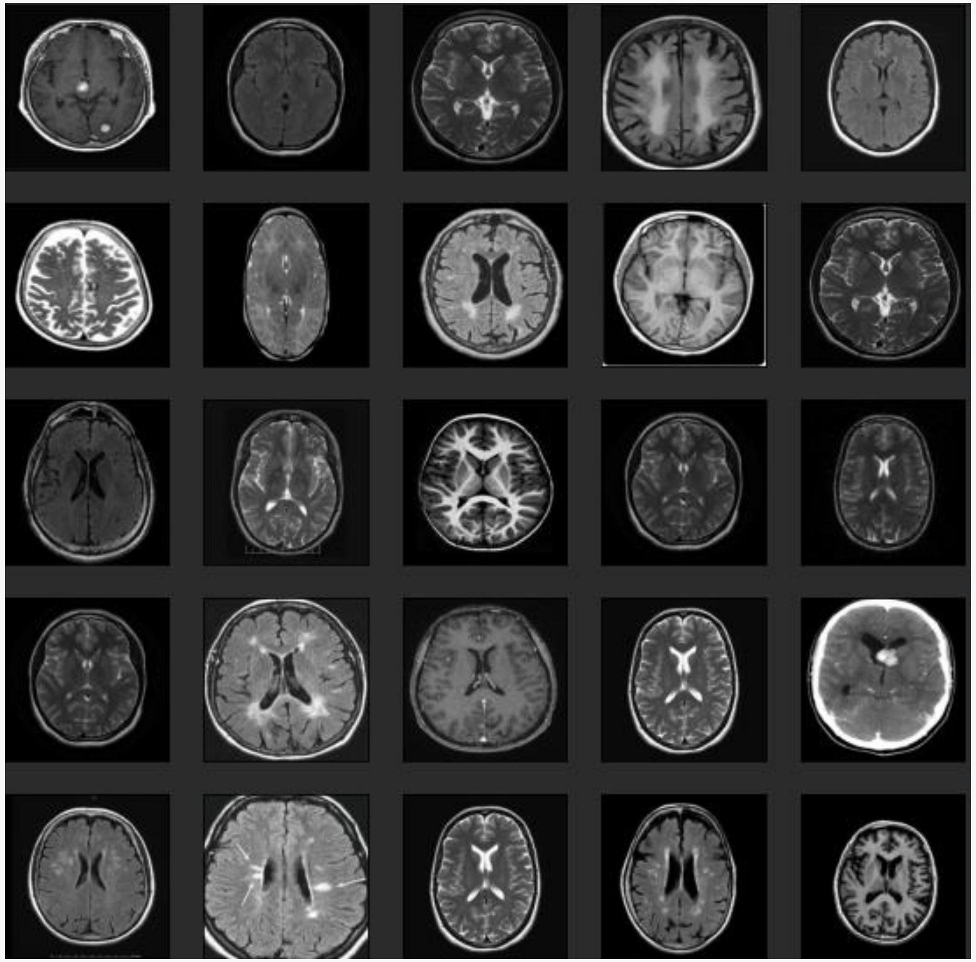
98 healthy scans and **155 tumorous** scans

All the files are **.jpg images**

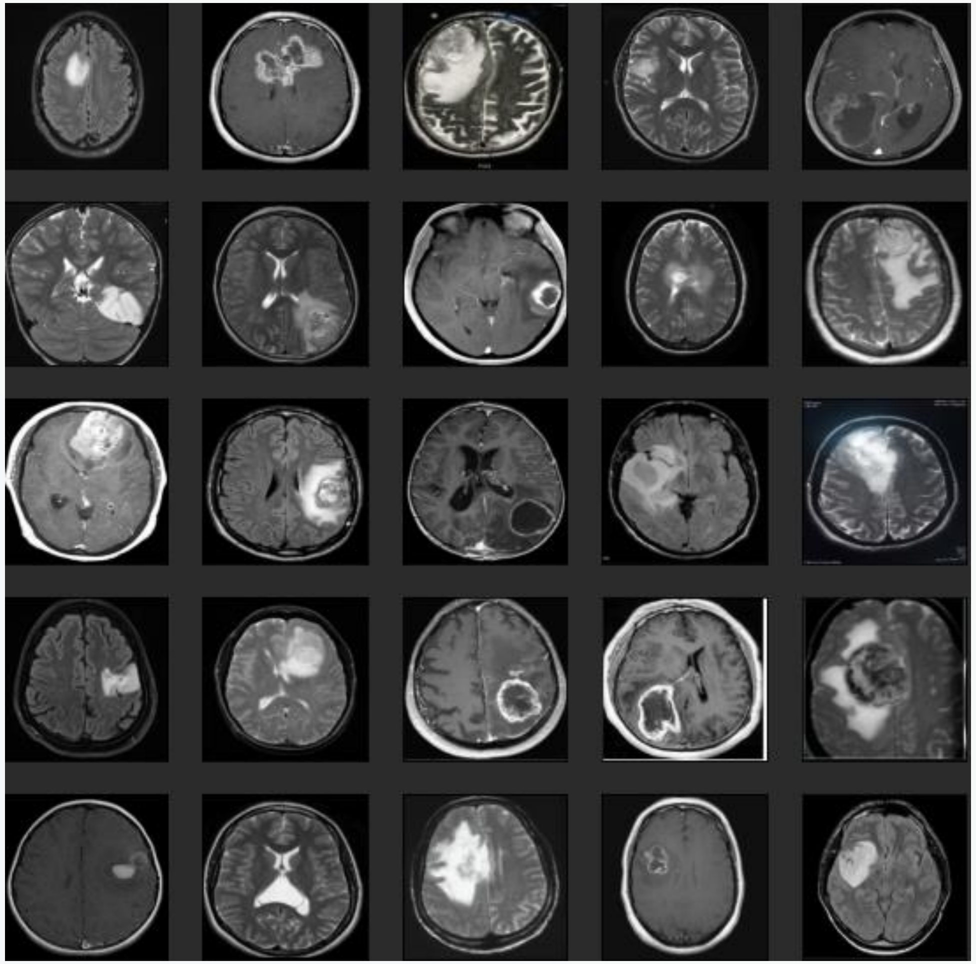
All images have **3 channels**

	MIN HEIGHT	MEAN HEIGHT	MAX HEIGHT	MIN WIDTH	MEAN WIDTH	MAX WIDTH	MEAN CHANNELS
Healthy	168	342.23	1080	150	343.16	1920	3.0
Tumorous	173	413.70	1427	178	361.24	1275	3.0

Data Exploration



Healthy Set



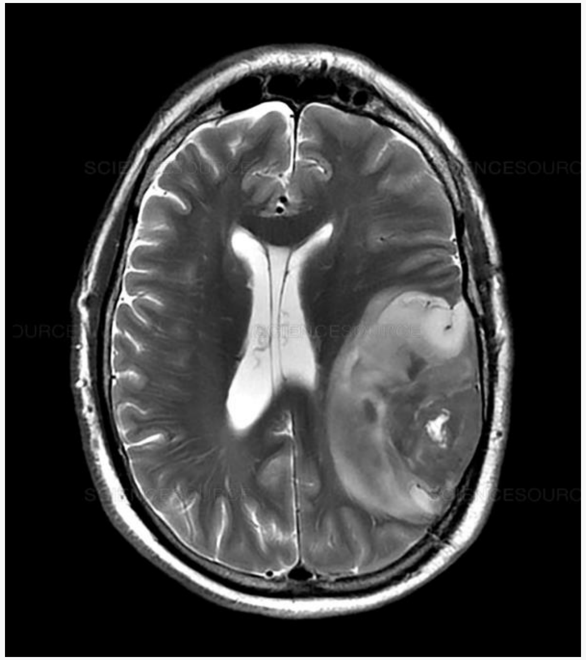
Tumorous Set

03 Methodology

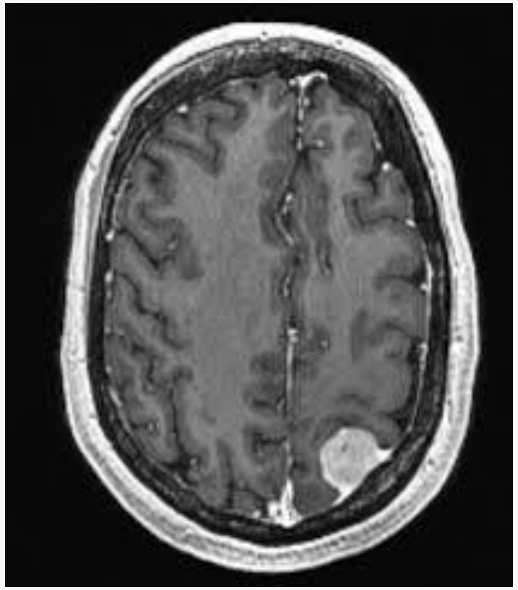
Data Preprocessing

Size Differences and Wasted Space

- Images in the dataset do not have the same dimensions
- CNNs input layer in Keras required images to be of the same size (height, width, channels)
- A significant portion of each image contains insignificant data
- The CNN only needs information about the brain itself, not the black border



y162.jpg



y56.jpg

IMAGE	
Dimensions	1059x1200
Width	1059 pixels
Height	1200 pixels

IMAGE	
Dimensions	211x239
Width	211 pixels
Height	239 pixels

Data Preprocessing

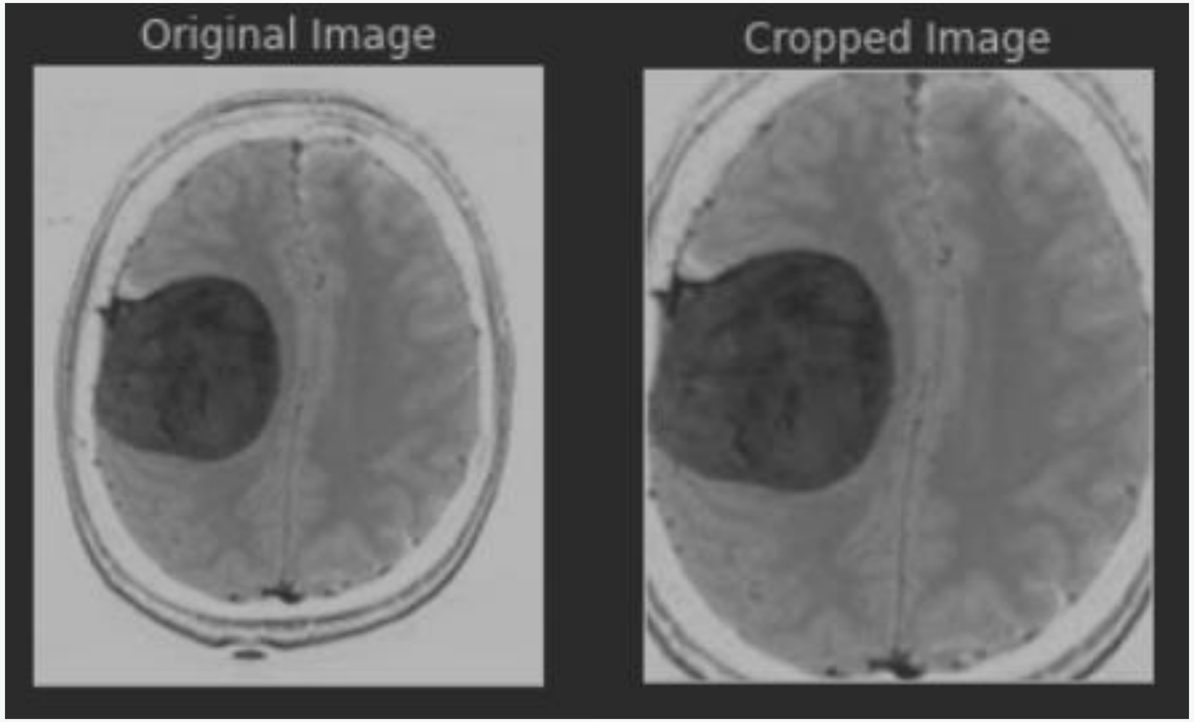
Size Differences and Wasted Space

Average Image Dimensions

	Height	Width	Channels
Healthy	342.23	343.16	3.0
Tumorous	412.70	361.24	3.0

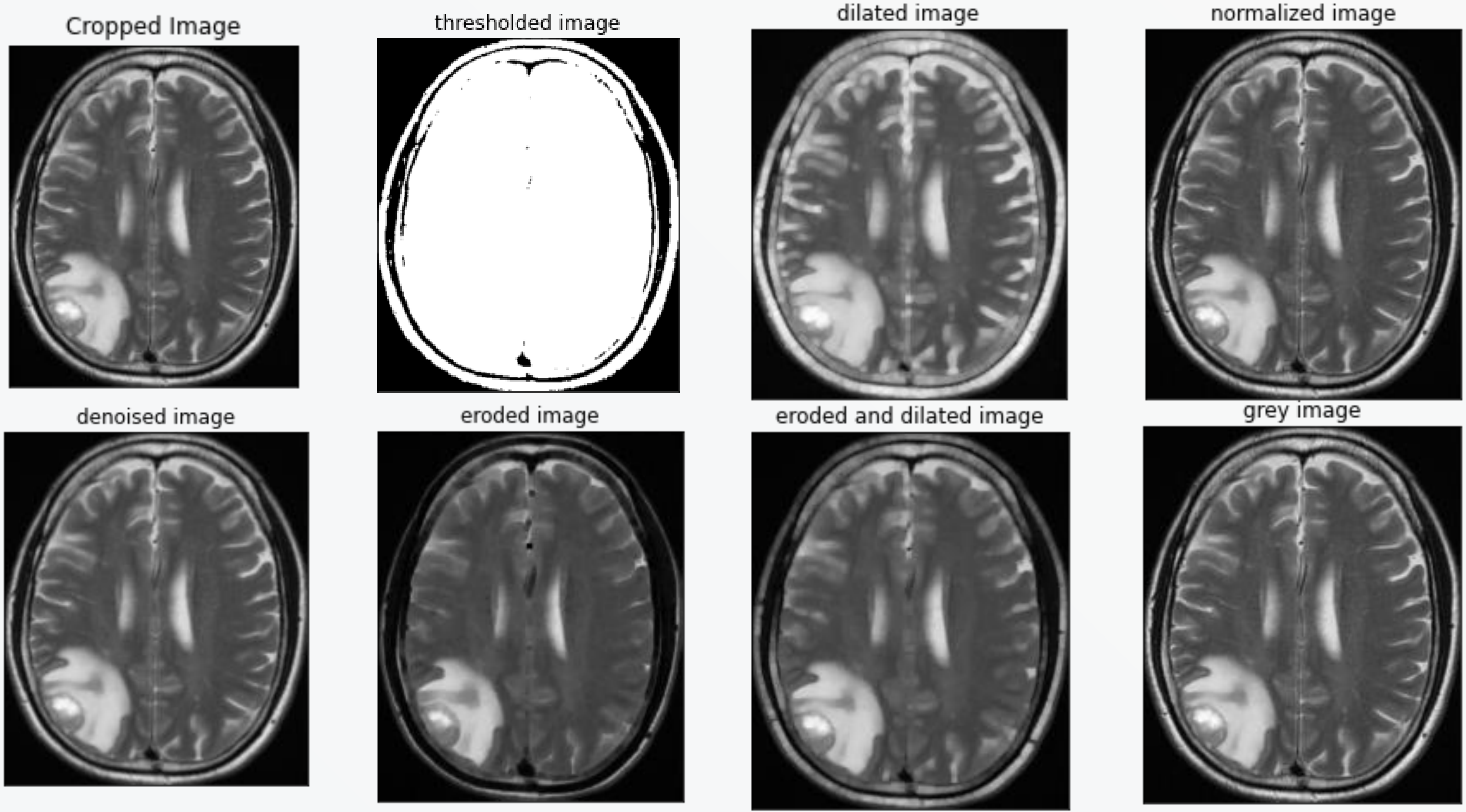
- Brain contour: four most extreme points in each corner
- Cropped based on brain contour

Image Cropping



Y1.jpg before and after applying the cropping technique

Data Preprocessing: *Other techniques*



Feature Engineering

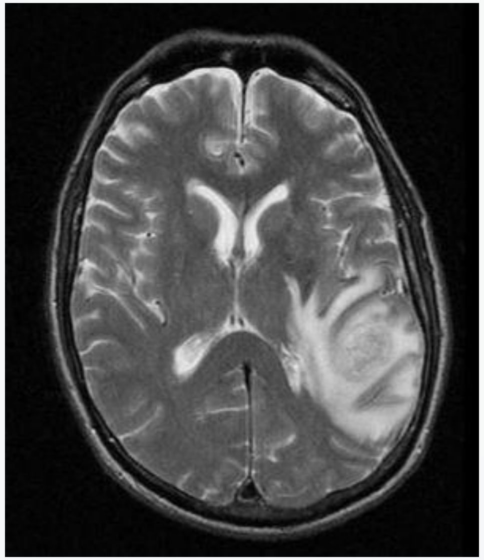
Data Augmentation

- **61%** of images are tumorous and **39%** of images are healthy

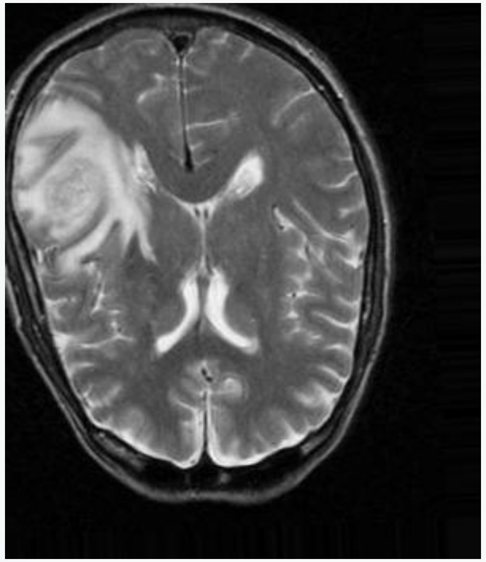
- Create 9 new images for each healthy scan
- Create 6 new images for each tumorous scan

- **rotation_range:** range for random rotations (in degrees)
- **width_shift_range:** shift, fraction of total width
- **height_shift_range:** shift, fraction of total height
- **shear_range:** shear angle in counter-clockwise direction in degrees
- **brightness_range:** range for randomly choosing a brightness shift
- **horizonal_flip:** randomly flips an image horizontally
- **vertical_flip:** randomly flips an image vertically
- **fill_mode:** how to fill points outside the input, but within the image boundary

```
ImageDataGenerator(rotation_range=10,
                   width_shift_range=0.1,
                   height_shift_range=0.1,
                   shear_range=0.1,
                   brightness_range=(0.3, 1.0),
                   horizontal_flip=True,
                   vertical_flip=True,
                   fill_mode='nearest'
                  )
```



Y71 original



Y71 Augmented

Data Loading and Splitting

Crop Images
- Apply cropping technique defined in pre-processing

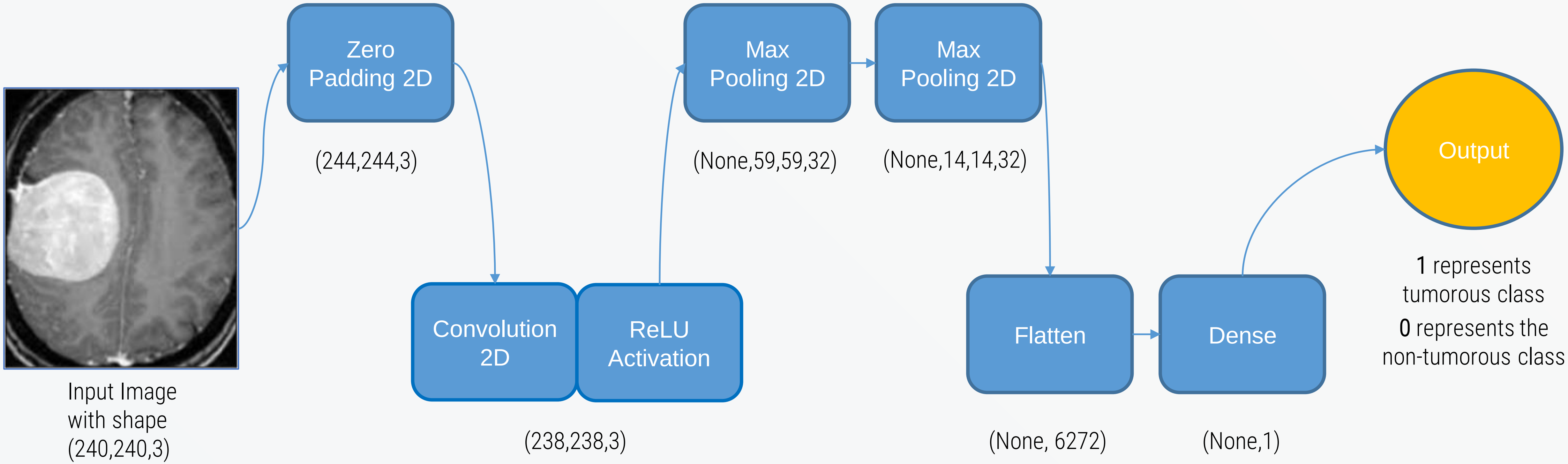
Resize Images
- Resize all images to (240,240,3)

Normalize Images
- Normalize pixel intensity

Classify Images
- Pair each image with its corresponding classification: tumorous or non-tumorous

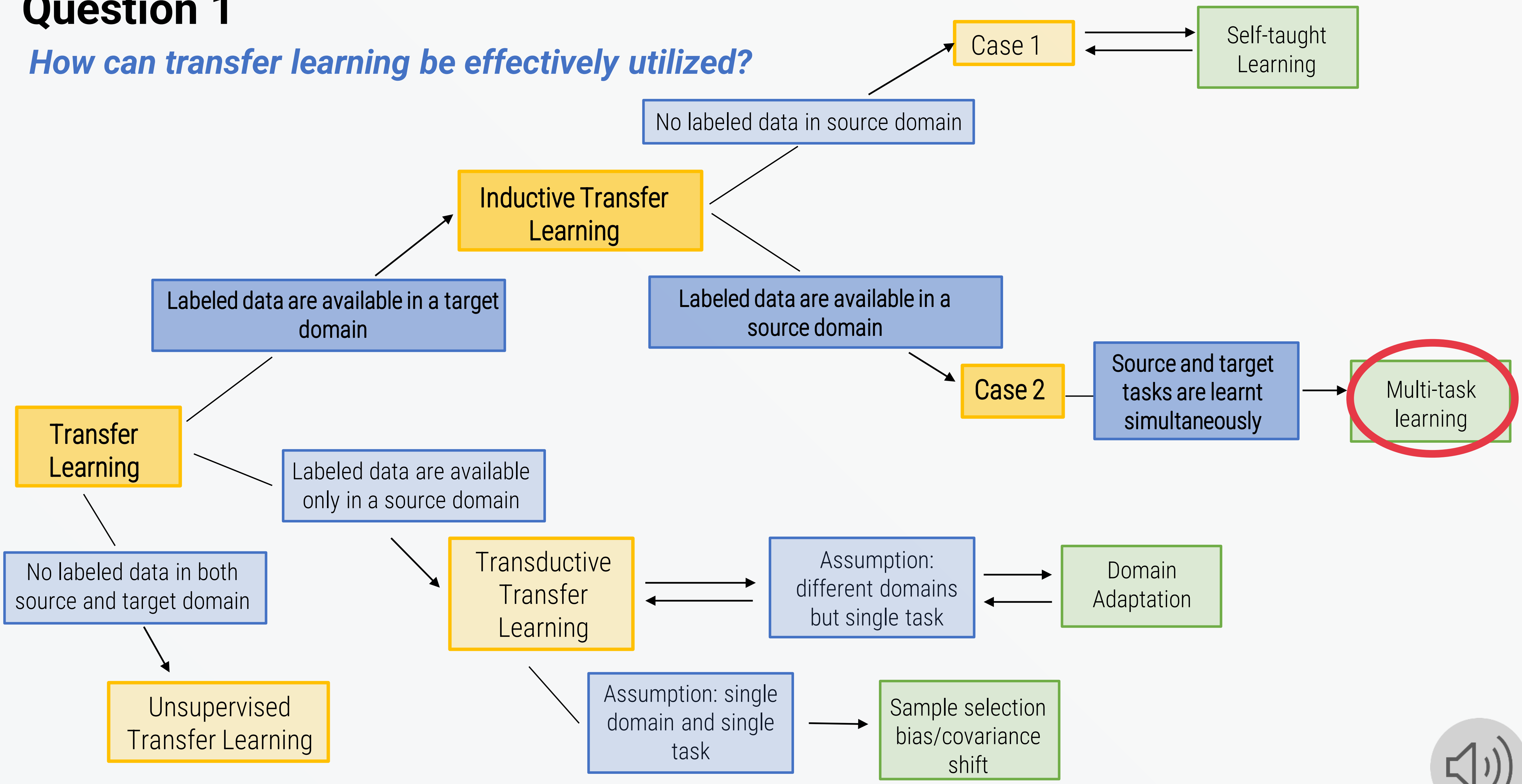
Shuffle Images
- Shuffle images to mix the two categories

Neural Network Architecture



Question 1

How can transfer learning be effectively utilized?



[A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning | by Dipanjan (DJ) Sarkar | Towards Data Science]

Question 2

Can preprocessing be reduced and what are the implications?

Pre-processing step(s):	Justification:
1. No pre-processing	• Use as a baseline for comparison
1. Cropping 2. Dilation	• Reduce memory required for preprocessing
1. Crop 2. Dilation 3. Normalization	• Normalize pixel values for memory considerations
1. Crop 2. Dilation 3. Denoise	• Eliminate artifacts and noise from images to improve true tumour detection
1. Crop 2. Dilation 3. Denoise 4. Normalization	• Normalize pixel values after denoising to avoid losing information about artifacts and noise
1. Crop 2. Dilation 3. Normalization 4. Denoise	• Normalize pixel values before denoising for memory considerations

With image dimensions (340, 340, 3)

With cropping and normalization

Image Dimensions
(140, 140, 3)
(240, 240, 3)
(340, 340, 3)
(440, 440, 3)

Question 3

Can model accuracy be sustained with lower computational complexity?

1 Convolutional Layer and Batch Size of 32

Epochs	Acc	Loss	Time
5			
10			
15			
20			

1 Convolutional Layer and 10 Epochs

Batch Size	Acc	Loss	Time
32			
64			

4 Epochs and Batch Size of 32

Convolutional Layers	Acc	Loss	Time
1			
2			

04 Results

RQ2 Results

With image dimensions (340, 340, 3)

Pre-processing step(s):	Justification:	Acc	Loss	Time
1. No pre-processing	• Use as a baseline for comparison	**Model:** 0.8672 **Validation:** 0.7629	**Model:** 0.3042 **Validation:** 0.6508	0:12:59.9
1. Cropping 2. Dilation	• Reduce memory required for preprocessing	**Model:** 0.9072 **Validation:** 0.8362	**Model:** 0.2064 **Validation:** 0.5987	0:12:37.5
1. Crop 2. Dilation 3. Normalization	• Normalize pixel values for memory considerations	**Model:** 0.9547 **Validation:** 0.8621	**Model:** 0.1442 **Validation:** 0.3361	0:14:38.5
1. Crop 2. Dilation 3. Denoise	• Eliminate artifacts and noise from images to improve true tumour detection	**Model:** 0.9002 **Validation:** 0.7974	**Model:** 0.2312 **Validation:** 0.6396	0:13:53.3
1. Crop 2. Dilation 3. Denoise 4. Normalization	• Normalize pixel values after denoising to avoid losing information about artifacts and noise	**Model:** 0.9271 **Validation:** 0.8793	**Model:** 0.2053 **Validation:** 0.3175	0:13:39.2
1. Crop 2. Dilation 3. Normalization 4. Denoise	• Normalize pixel values before denoising for memory considerations	**Model:** 0.9406 **Validation:** 0.8750	**Model:** 0.1549 **Validation:** 0.2906	0:14:08.3

RQ2 Results (Continued)

With Cropping and Normalization

Image Dimensions	Acc	Loss	Time
(140, 140, 3)	Model: 0.8656 Validation: 0.8017	Model: 0.3236 Validation: 0.4038	0:02:26.4
(240, 240, 3)	Model: 0.9212 Validation: 0.8707	Model: 0.2146 Validation: 0.2828	0:12:17.6
(340, 340, 3)	Model: 0.9498 Validation: 0.8836	Model: 0.1475 Validation: 0.3196	0:13:21.4
(440, 440, 3)	Model: 0.9504 Validation: 0.8750	Model: 0.1461 Validation: 0.3433	0:42:5.4

RQ3 Results

1 Convolutional Layer and Batch Size of 32

Epochs	Acc	Loss	Time
5	Model: 0.8721 Validation: 0.8362	Model: 0.3050 Validation: 0.3867	0:7:26.9
10	Model: 0.9498 Validation: 0.8836	Model: 0.1475 Validation: 0.3196	0:13:21.4
15	Model: 0.9892 Validation: 0.8276	Model: 0.0695 Validation: 0.5006	0:20:15.9
20	Model: 0.9978 Validation: 0.8879	Model: 0.0224 Validation: 0.3900	0:27:49.5

1 Convolutional Layer and 10 Epochs

Batch Size	Acc	Loss	Time
32	Model: 0.9342 Validation: 0.8578	Model: 0.1765 Validation: 0.3844	0:13:50.6
64	Model: 0.9153 Validation: 0.8362	Model: 0.2293 Validation: 0.3803	0:19:50.6

4 Epochs and Batch Size of 32

Convolutional Layers	Acc	Loss	Time
1	Model: 0.8322 Validation: 0.8190	Model: 0.3889 Validation: 0.4016	0:5:32.1
2	Model: 0.8554 Validation: 0.6681	Model: 0.3353 Validation: 0.8161	0:39:38.8

05 Discussion

Discussion

Machine Power

Compute Resources Available

Availability of Data

Large-scale, industry- data is not open-source

Novelty

Boosting accuracy while applying transfer learning techniques

What worked? What Didn't?

Hyper-parameter tuning

Thank you for listening!

Please feel free to reach out with any additional questions:
Madison Boem – 16mab11@queensu.ca
Caroline Kim – kim.caroline@queensu.ca
Eric Venditti – eric.venditti@queensu.ca
Jade Watson – 16jw73@queensu.ca