Brain Tumor Detection System

#2 View of Progress as of 2/18/2024

Abdullah Alshamrani
Saint Joseph’s University

BIG CHANGE

Dataset modified to another dataset

DUO to the big time will be spent to classify each image to a “YES” has tumor and to “No” Image does not have tumor.

I was able to find another dataset that is already classified as a “Yes”, “No” tumor and a “Pred” classification to predict if the image has a tumor to serve as the final delivery of this project where images will be chosen by me to test my model.

Reference to the NEW Dataset:
https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection/data

Last View 2/4/2024

I have concluded that I had done the following:

- **Set up repository for the project**

- **Collected/downloaded the data set for the brain tumors**

- **Google Colab upload to initial exploration**

Since 2/4/2024: Two weeks passed

Sine 2/4/2024, Two weeks have passed, So I am expected to finish all tasks of week 1-2 and a start of tasks in week 3-4: The following is a reminder of what week 1-2 includes of tasks and week 3-4 includes. After that what have been done will be shown.

Biweekly Outline

- Weeks 1-2: Project start point and Data Collection

- Tasks:

Set up the project repository.

Collect and download the brain tumor dataset from Kaggle.

Upload the dataset to Google Colab for initial exploration.

Data Visualization and understanding

Begin the exploratory data analysis (EDA) to understand the dataset's characteristics.

Week 1-2 Accomplished NEW Tasks

- **Data Visualization and Understanding**

- **Understanding Concepts outcomes**

1. **Data Augmentation Techniques**
2. **Optimization Techniques: Accuracy in mind**
3. **Keras “neural network API written in Python” tools to improve data …..**
4. **SVM & MPE Machine Learning Models**
5. **While using tools and discovering models/methods I keep in mind that a good model definition for my project is the necessity of High Test and validation accuracy**
6. **Gradients and HoG “Histogram of Gradients”**

Visualization

After Understanding Concepts and having something in mind and techniques I started my visualization for the data.

Simple Visualization was done to get me started to the NEXT Task which is EDA ‘Exploratory Data Analysis’. NEXT SLIDE will be an explanation of the code that was performed and a show of the data after visualization.

```python
# Path to the dataset
dataset_path = '/content/drive/MyDrive/BT-New dataset'

# Subdirectories to process
subdirs = ['no', 'yes', 'pred']

# Iterating through each subdirectory and listing all the files
for subdir in subdirs:
    subdir_path = os.path.join(dataset_path, subdir) #Creat a new path by combining with the subdirectories
    print(f"Listing files in {subdir_path}:")
    files = os.listdir(subdir_path) #list all files of the subdirectory
    for file in files:
        #path to the file for further processing when needed
        file_path = os.path.join(subdir_path, file) # Subdirectory "YES" or "NO" + file name = full path for the file
        print(file_path)
```

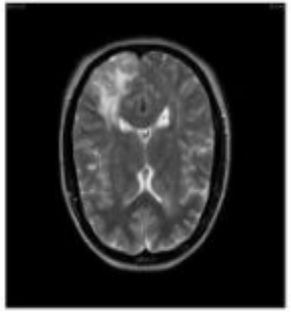
Sample of the output

```
Listing files in /content/drive/MyDrive/BT-New dataset/no:
/content/drive/MyDrive/BT-New dataset/no/no1023.jpg
/content/drive/MyDrive/BT-New dataset/no/no73.jpg
/content/drive/MyDrive/BT-New dataset/no/no98.jpg
/content/drive/MyDrive/BT-New dataset/no/no293.jpg
/content/drive/MyDrive/BT-New dataset/no/no287.jpg
/content/drive/MyDrive/BT-New dataset/no/no250.jpg
/content/drive/MyDrive/BT-New dataset/no/no1221.jpg
```

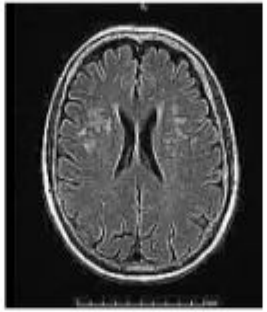
>> Image Visualization

```python
#Image Visualization for a sample of the DATA, 'YES' 'NO' Classes

def visualize_images(dataset_path, subdirs=['no', 'yes'], samples_per_class=3): # This is for visulizing samples from Class YES 'has tumors', Class NO 'Has NO Tumors'
# 3 samples each class 3 * 'NO'  - 3 * 'YES'

    plt.figure(figsize=(12, 4)) #specific size of the figure
    for idx, subdir in enumerate(subdirs): # 'enumerate' provides a counter (idx) along with the value (subdir) as you iterate over the list.

        subdir_path = os.path.join(dataset_path, subdir) #creating full path to the subdirectory
        files = os.listdir(subdir_path)[:samples_per_class]  # Get the first few samples.slicing to get the first three [:]
        for i, file in enumerate(files): # loop over within index
            img_path = os.path.join(subdir_path, file) #creat image path
            img = plt.imread(img_path) # read the image file
            plt.subplot(len(subdirs), samples_per_class, idx * samples_per_class + i + 1)
            # Subplot for each image. num of rows = 2 'number of subdirectories' = 'YES' , 'NO'
            # Columns = 3, 'Number of samples per class'

            plt.imshow(img, cmap='gray') #set to grayscale
            plt.title(f"Class: {subdir}")
            plt.axis('off')
    plt.tight_layout() # adjusts subplot params so that the subplot(s) fits in to the figure area
    plt.show()

visualize_images(dataset_path)
```

Class: no

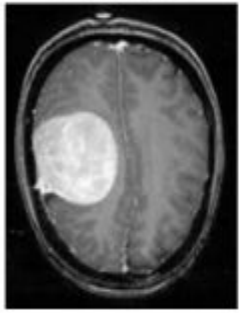


Class: no



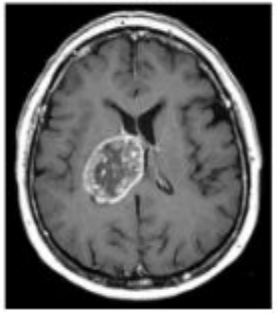
Class: no

Class: yes

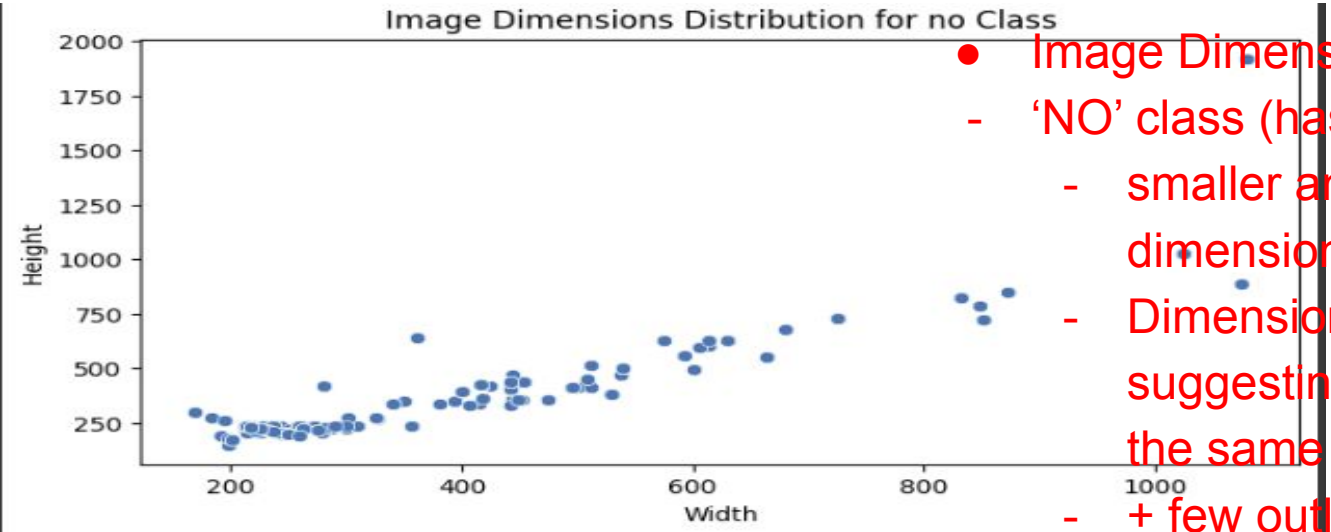


Class: yes

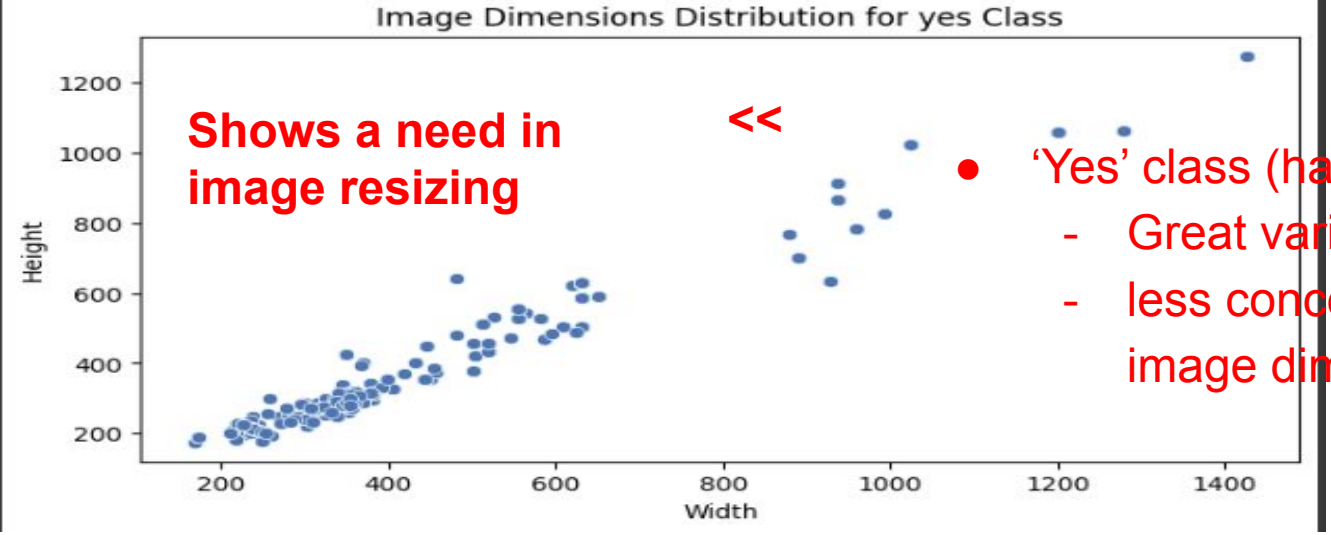
Class: yes



Initial Exploratory Data Analysis



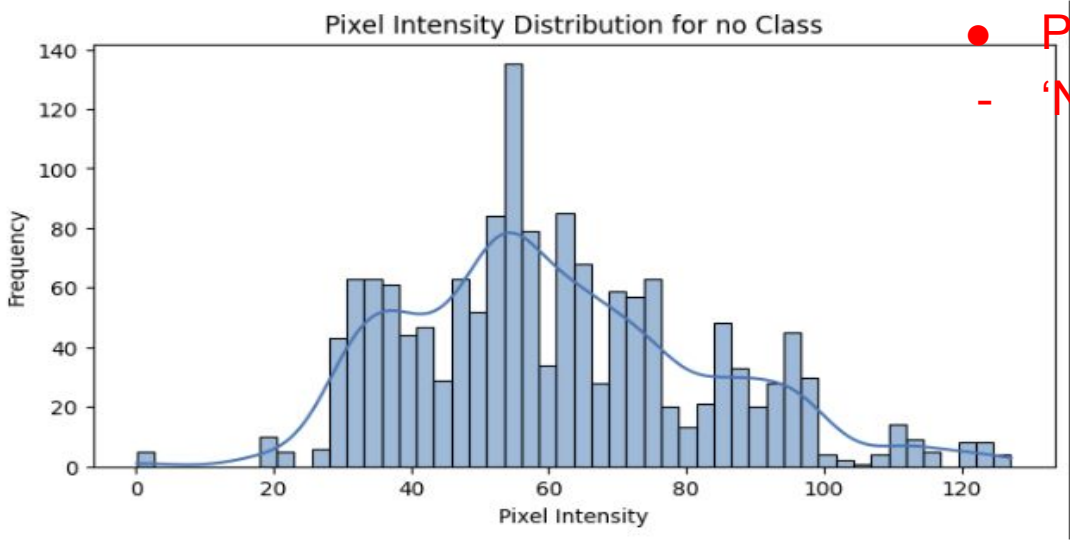
- Image Dimensions Analysis
  - ‘NO’ class (has no tumor)
    - smaller and clustered around small dimensions
    - Dimensions are relatively tight, suggesting they are generally having the same size
    - + few outliers



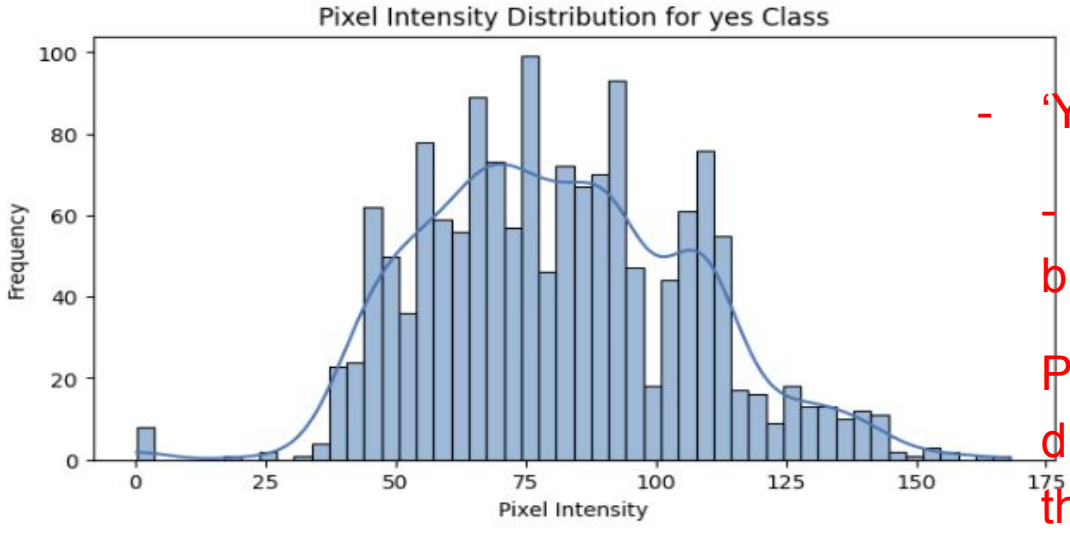
Shows a need in image resizing

<<

- ‘Yes’ class (has a tumor)
  - Great variability
  - less concentrated = a diverse set of image dimensions.



- Pixel Intensity distribution
  - ‘NO’ class (has no tumor)
    - Centered around mid-range = suggests moderate contrast
    - Symmetrical spread = consistent brightness



- ‘Yes’ class (has a tumor)

- Broad spread = variance in brightness and contrast

Peaks = presence of regions with different brightness - because of the tumors

Time measurement of new tasks of week 1-2

- Data Visualization and Understanding - 7 Hours of work:

While it was shown in my specification paper that Data Visualization and understanding will take around 5 hours of time, this changed when I measure my time in understanding and visualizing the data and increased up to “2” hours. So a total of 7 hours were spent.

- Initial EDA - 5 Hours of work

Week 1-2 Tasks are over

- Weeks 3-4: Data Preprocessing and Model Selection
- Tasks:

Preprocess the dataset for training, including resizing, normalization, and augmentation.

Split the dataset into training and testing sets.

Implement and train traditional machine learning models (SVM, Random Forest).

Begin the implementation of Convolutional Neural Networks (CNNs) for deep learning.

Preprocess the dataset → Resizing, Normalization and augmentation + Training And Testing Sets

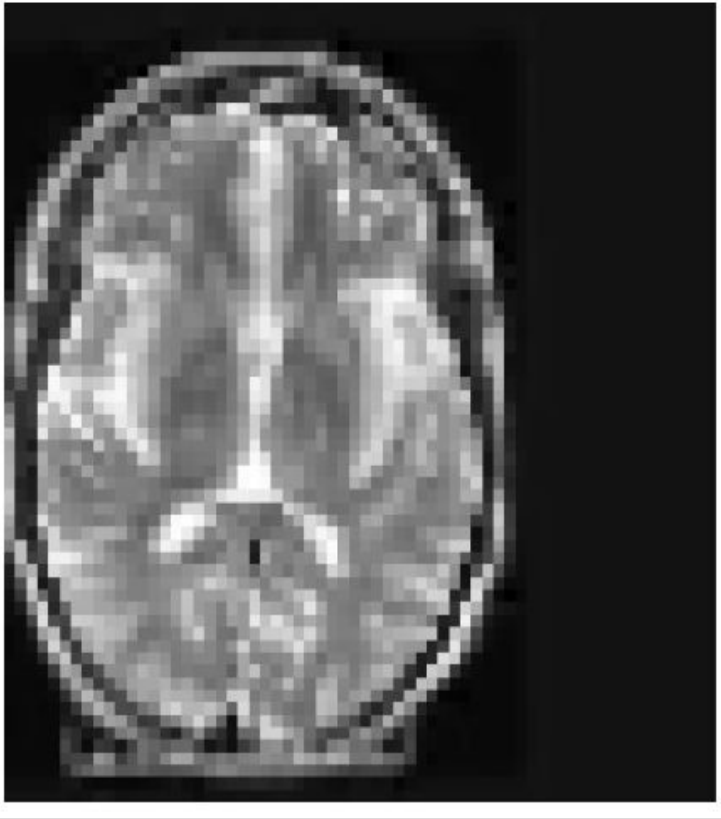
```
subdirs = ['no', 'yes']

#storing data and labels
data = []
labels = []

# Preprocessing
for subdir in subdirs:
    files = os.listdir(os.path.join(dataset_path, subdir))  # List files in subdir

    for file in files:
        img_path = os.path.join(dataset_path, subdir, file)
        image = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
        image = cv2.resize(image, (64, 64))  # Resizing

        # Normalizing the image
        image = image / 255.0

        data.append(image)  # Append proccessed to the data list
        labels.append(subdir)

# data and lables to numpy arrays to proccess with machine learning
data = np.array(data)
labels = np.array(labels)

# Spliting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
```

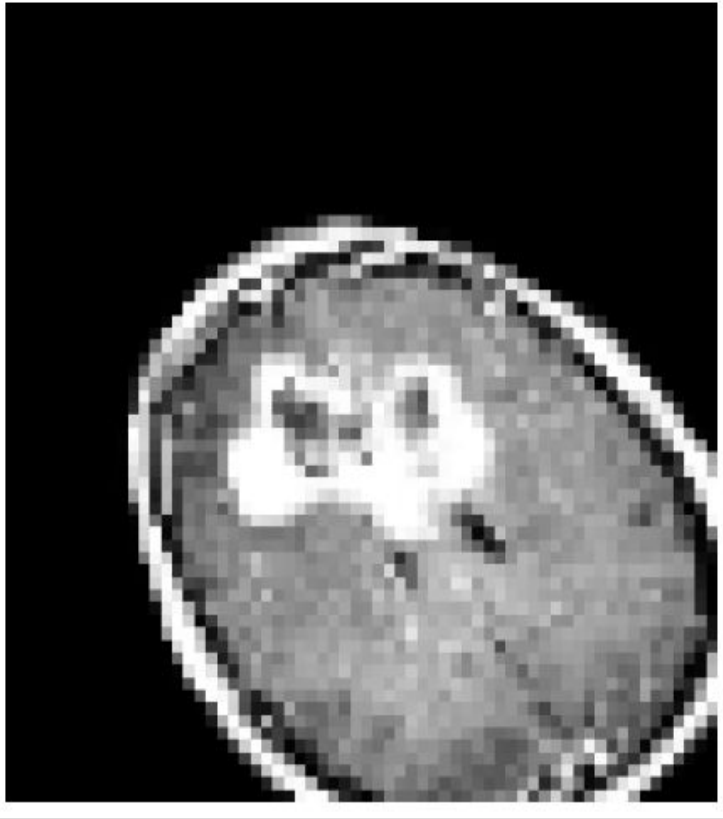
Normalization is a data transformation process that aligns data values to a common scale or distribution of values so that. Normalization includes adjusting the scale of values to a similar metric or adjusting the time scales to be able to compare like periods.[2]

Augmentation of data using Keras “ImageDataGenerator”

```
datagen = ImageDataGenerator( # Defining Paramters for the augmented data
    rotation_range=20,  #  Random rotate of images
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'  # Fill in new pixels that may emerge after a rotation or width/height shifts
)

X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], X_train.shape[2], 1))  # Add the channel dimension
datagen.fit(X_train)  # Now fit the data generator
#X_train.shape == num of images in the dataset
#X_train.shape [1] , X_train.shape[2] ==> height, width
#1 Keras requires (batch_size, height, width, channels). greyscale channel is 1


datagen.fit(X_train) # when data are under the proccess of training, augmented data will be generated
```

Sample of the Augmented Data





- Estimated Hours:

Data preprocessing: 5 hours

Traditional ML model implementation and training: 8 hours

CNN implementation: 8 hours

Under the process to be completed by 2/25/2024

- **Implementing Traditional Machine Learning Models**

- **Begin the Implementation of Convolutional Neural Networks “CNNs” for deep learning**

References:

1. https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection/data
2. https://c3.ai/glossary/data-science/normalization/#:~:text=Normalization%20is%20a%20data%20transformation,able%20to%20compare%20like%20periods.