Team Members

1. Krista Radecke (GT ID: 903736778)
2. Sandra Kim (GT ID: 902745258)
3. Sean Lee (GT ID: 903648883)

Project Title: Brain Tumor Image Recognition

Background

According to GE healthcare, over 90% of healthcare data comes from medical imaging and more than 97% of images captured are not analyzed[2]. There are not enough radiologists in developing countries to diagnose all images efficiently and accurately. Even in developed nations, waiting times for image analysis can be over 30 days, depending on resources or availability of physicians who can accurately diagnose patients[2]. Another advantage of using AI for image classification is a higher accuracy of classification when compared to pathologist diagnosis. The Google powered AI for detecting breast cancer metastasis has a 99% accuracy compared to human pathologists, who can miss the diagnosis 62% of the time[2]. The lack of resources and lower accuracy of a human pathologist can cause a significant delay in the start of treatment. When the treatment is delayed due to lack of a diagnosis, there are significantly higher costs involved with misdiagnosis than there are for over-diagnosis. For example, risks such as higher medical bills, lower life expectancy, and reduction of quality of life[3]. Having a means to minimize these risks would be extremely beneficial to the communities and overall wellbeing of patients.

Creating a one-size-fits-all AI model for MRI diagnosis can be challenging because MRI images can come in assorted sizes, different positioning of the head, with varying levels of contrast and background noise. If all images fed into the model are regularized, the accuracy of the model could potentially increase when seeing a fully new data set, thus expanding the scope of the model's usability. We can utilize suggestions given by high performers of the Large Scale Visual Recognition Challenge to preprocess all images and possibly increase model performance[1].

Problem statement

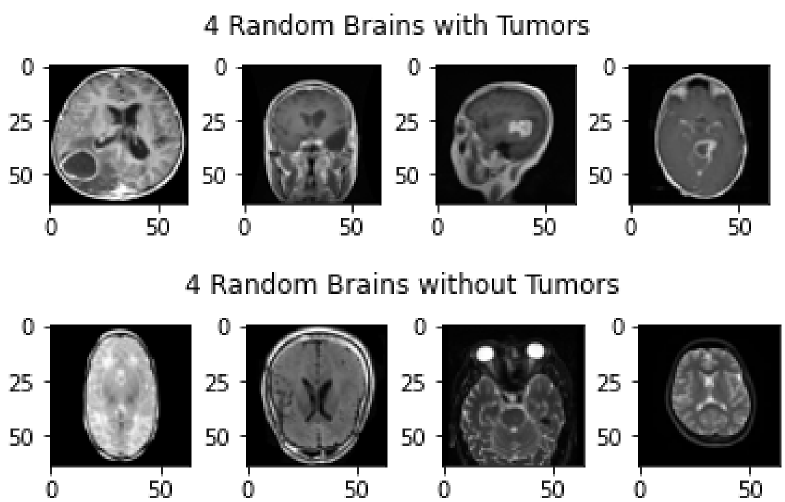
Identifying patients with brain tumors can be complicated, and with increasing demands on medical providers, there may be missed opportunities to diagnose patients. If diagnosed early enough, managing tumors through lifestyle changes or earlier invasive surgery can lead to a longer, healthier life for patients. Our project is to develop a process for regularizing images and detecting tumors, thus expediting the diagnosis process for a quicker medical response and relieving the resource burden in developing countries.

Dataset

Our project uses 2 datasets: the first dataset will be used for training and validation. The second dataset will serve as a testing dataset to ensure that our model fits appropriately and is not overfitted to the first dataset.

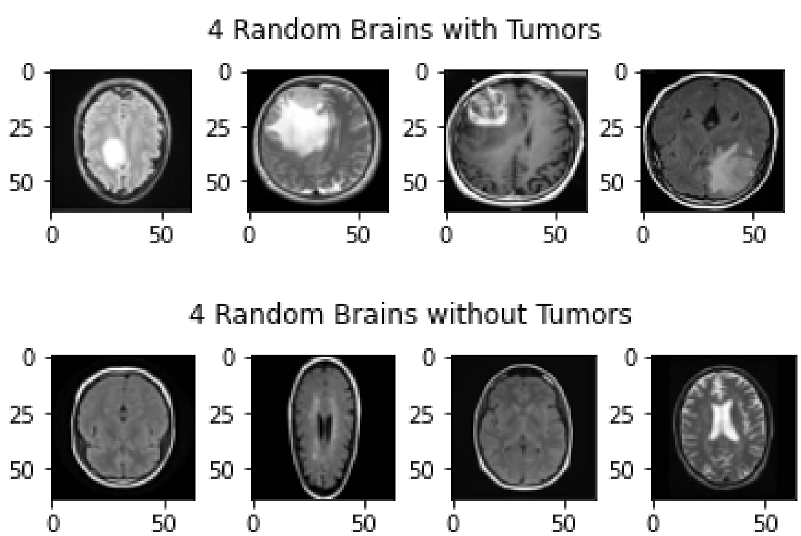
https://www.kaggle.com/preetviradiya/brian-tumor-dataset

- 2513 brain tumor images, 2087 healthy brain images. Split the dataset into 80% training and 20% validation images. Here is an example of the data of both healthy brains and brains with tumors:



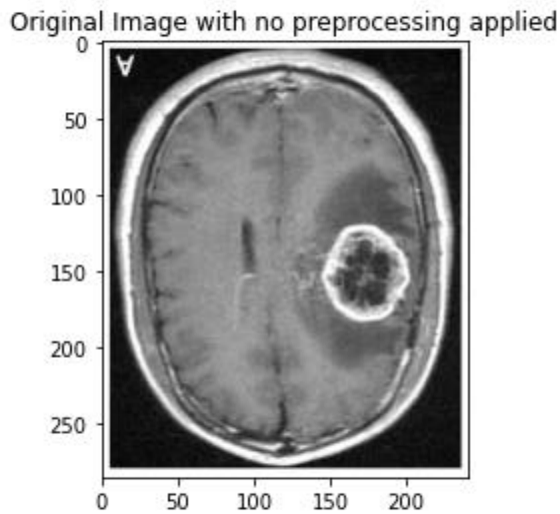
https://www.kaggle.com/ahmedhamada0/brain-tumor-detection

- 3501 train images, 161 test images, 202 validation images. This will be the test dataset that we evaluate our models on.



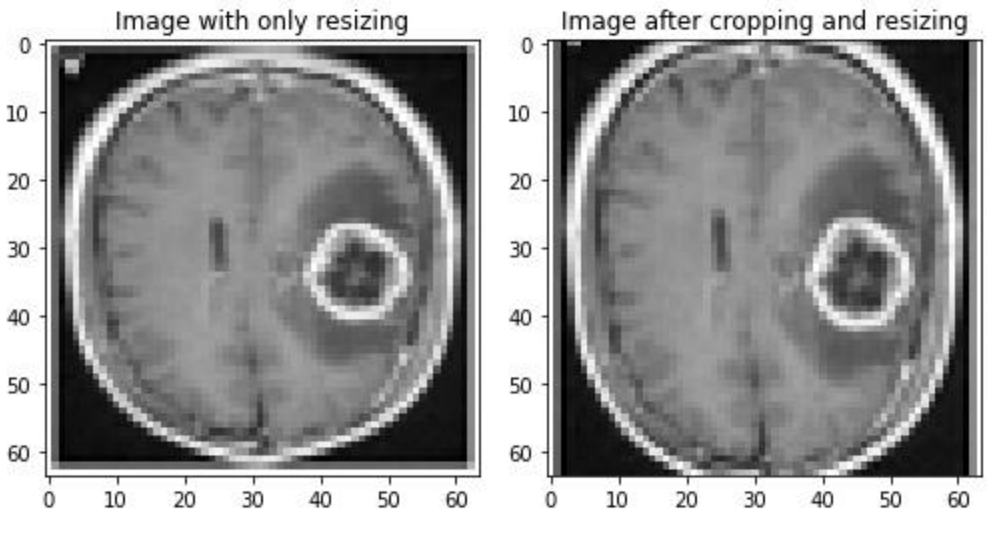
Methodology

There were several image preprocessing techniques that we discovered from the Large Scale Visual Recognition Challenge that we wanted to experiment with. We demonstrated all of the techniques on the same sample MRI of a brain with a tumor. The reference image with no transformations applied is as follows:



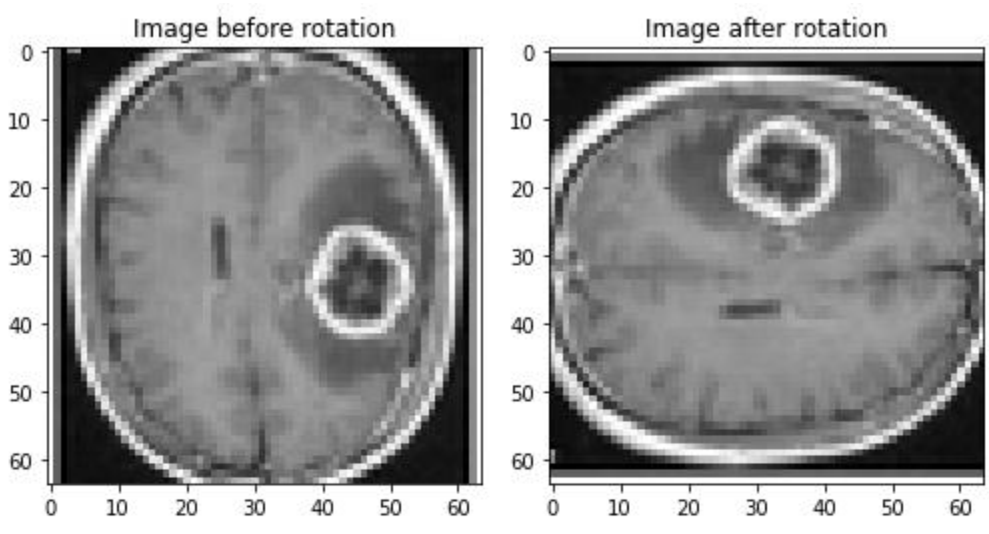
Resizing and cropping

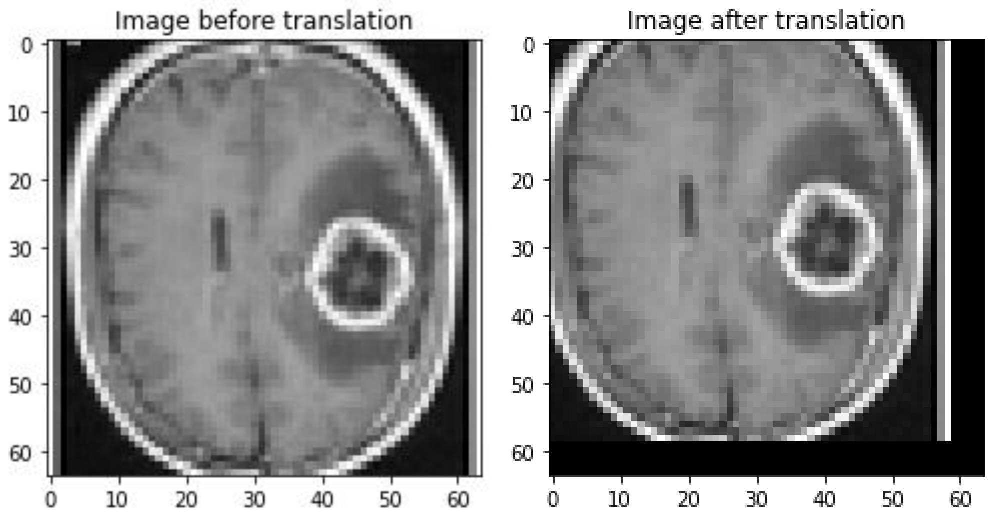
The first technique was for cropping and resizing the images. All our models required the images to be the same size so we could create a data matrix. Our dataset contained images of different shapes and sizes, some were rectangular, and the others were square. If we used the resize function on rectangles, the images would be compressed into a square which would not keep the aspect ratio. The technique recommended resizing the smallest side of the rectangle to the size we wanted, keeping the aspect ratio. We would then crop an equal number of pixels from either side of the image on the longer side to keep the subject of the image centered. An example of the transformation can be seen below. The resulting image is a square, but it is more representative of the original image.



Rotation and Translation

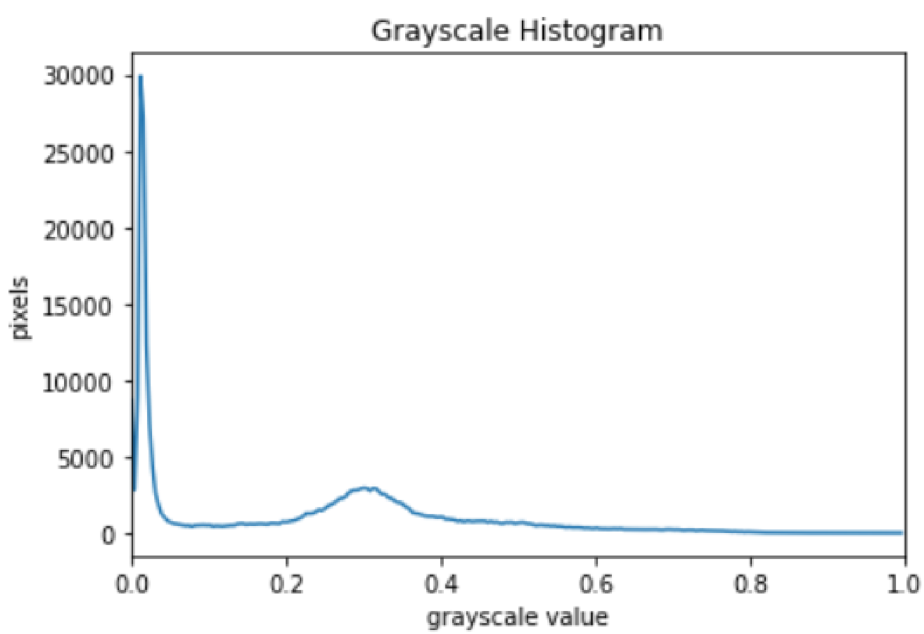
The next technique we explored was randomly rotating and translating some images. We initially planned on randomly rotating and translating 20% of the images from each data set to introduce variation. The idea was that the model would adapt to identify tumors regardless of the image's orientation. When we implemented these transformations, the accuracy of our models decreased by around ~5% for both datasets. We made the decision to exclude rotation and translation, because we only wanted to include preprocessing which would increase the performance of our models. These kinds of random transformations could be useful in model creation if there is a larger data set available, but our models were too affected by rotation and translation.



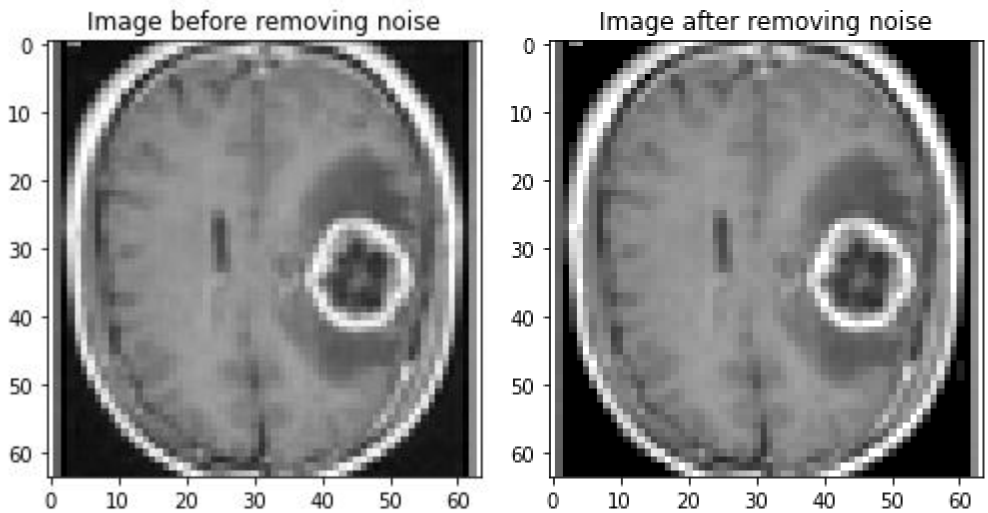


Removing noise

For improved consistency and to potentially have the tumor stand out more in the image, we implemented a noise reduction function to both blacken and whiten the images. An image grayscale ranges from 0 to 1, black and white respectively. “Grayscale Histogram” shown below exemplifies the typical grayscale image for our dataset, with most of the pixels in the black region. After evaluating the images and predictions to find the highest matching accuracies, we set thresholds of 0.08 for black and 0.92 for whites. Any image within the range 0-0.08 were turned complete black 0 and any image within the range 0.92-1.0 were turned complete white 1.



This function was not made to change the image visually, but just reduce the variation of pixels which should be plain white or black. The images below look the same before and after the implementation, but the first 10 rows and columns of the image array displayed following the images show the performance of the function.

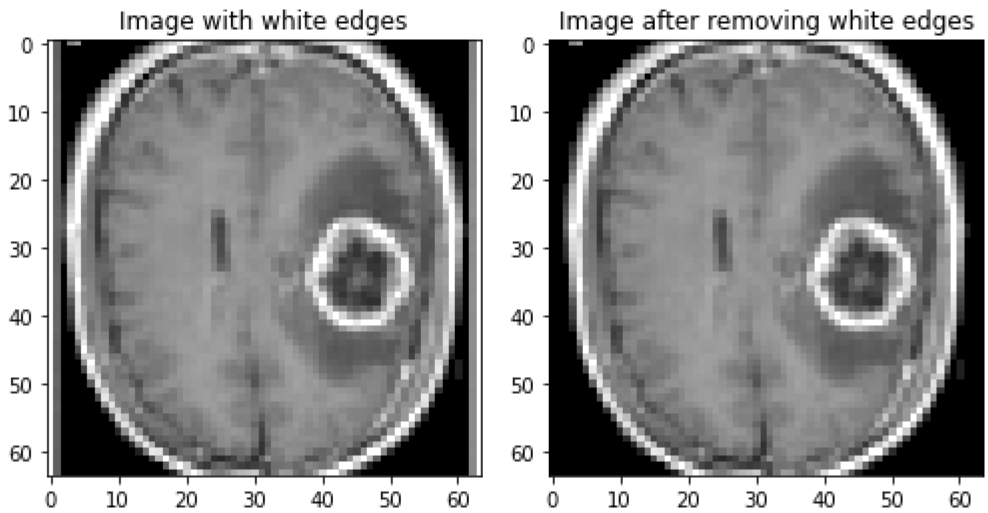



```
Image array before removing the noise
[[249 118  20  26  23  22  25   2 138 255]
 [174  36  25  22  23  19  24   2 138 255]
 [ 47  24  27  20  23  21  24   5 138 255]
 [ 21  25  24  26  22  20  24   5 138 255]
 [ 29  25  19  20  24  21  26   4 138 255]
 [ 23  21  17  18  21  20  20   1 136 255]
 [ 18  18  19  19  17  17  21   2 138 255]
 [ 22  18  20  23  18  19  22   0 135 255]
 [ 17  22  23  18  20  19  26   2 136 255]
 [ 18  21  18  18  19  19  22   1 136 255]]
```

```
Image array after removing the noise
[[255 118   0   0   0   0   0   0 138 255]
 [174  36   0   0   0   0   0   0 138 255]
 [ 47   0   0   0   0   0   0   0 138 255]
 [  0   0   0   0   0   0   0   0 138 255]
 [  0   0   0   0   0   0   0   0 138 255]
 [  0   0   0   0   0   0   0   0 136 255]
 [  0   0   0   0   0   0   0   0 138 255]
 [  0   0   0   0   0   0   0   0 135 255]
 [  0   0   0   0   0   0   0   0 136 255]
 [  0   0   0   0   0   0   0   0 136 255]]
```

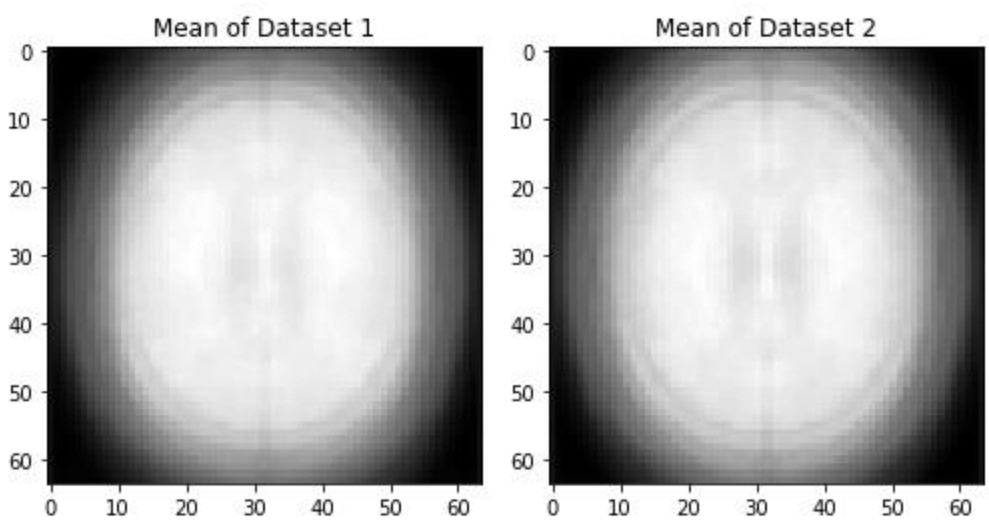
Removing White Edges

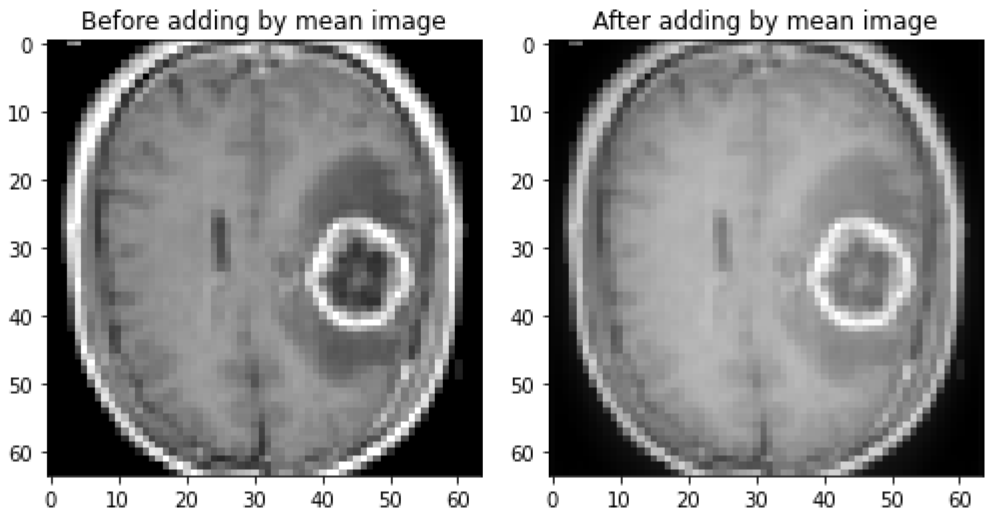
Some of the images in the dataset contained white/grey borders around the edges of the image. To make sure every image was in the same format, if the image had borders, they were removed.



Mean Image

To standardize, we took the mean image of each dataset and then added the mean to each image within its respective dataset. Mean images of Dataset 1 and Dataset 2 are shown below. Unfortunately, standardizing the images using the mean decreased prediction accuracy. Some models even had a reduction in accuracy by about 24%. Looking at the example images below of “Adding by mean image”, the function does appear to add more noise and reduce color contrast. Because of this, we decided to remove the add mean image function from our code.





Standardization

Since we removed the rotation and translation functions, adjusted for black and white thresholds, and removed adding the mean image to each image, we decided to evaluate our models with and without the standard scaling function provided by sklearn.preprocessing, StandardScaler(). This function standardizes features by removing the mean and scaling to unit variance.

Model Creation

After completing the preprocessing steps, we used the normalized images to train different classification models to identify “tumor” images vs “no tumor” images. Though each of these classification models have different algorithms for identification, we can make early inferences for which ones will perform better on the dataset. Because our experiment aims to classify the images as “tumor” vs “no tumor”, we inferred that the images were linearly separable. Thus, we suspected that a linear decision boundary would perform the best. To test this hypothesis, we considered both linear and non-linear decision boundaries and selected the model with the best performance. The models we ran for linear classification are logistic regression, Naïve Bayes, and SVM. For non-linear classification, we ran KNN, Neural Networks, and kernel SVM. We used 80% of the first dataset to train the models and 20% for validation, which includes hyperparameter tuning. We then used the full second dataset as a testing set to determine our models’ accuracy on an unseen dataset. We expected that the models would perform better on the classification set from the first data set.

Evaluation and Final Results

Once the models were trained and tuned, we evaluated the performance of the models using several different measurement techniques on the second dataset with both raw and normalized images. It is vital to report classification scores on both sets of images, so we can measure the feasibility of image preprocessing. For model scoring, we generated a confusion matrix and a classification report for each model. From the classification report, we used precision, accuracy, recall, and F-1 scores to compare the scores from different data sets. It is especially important that the recall score is high for any image classification that leads to a diagnosis, because we want to reduce the number of false negatives as much as possible. A false negative is equal to a misdiagnosis, which is costly to the patient in both medical expenses and health. A misdiagnosis would give the patient a false sense of security that they do not have a tumor but then come to find the tumor when it is in the later stages of cancer, which could be both a costly and potentially fatal mistake. Model performances are shown below.

Results for test set of data set 1

Classifier	**Preprocessing?**	**Accuracy**	**Recall**	**Precision**
Logistic Regression	With	0.96	0.96	0.96
	Without	0.96	0.96	0.96
Naïve Bayes	With	0.59	0.60	0.61
	Without	0.59	0.60	0.61
KNN	With	0.92	0.92	0.92
	Without	0.91	0.92	0.91
Linear SVM	With	0.66	0.65	0.66
	Without	0.79	0.79	0.79
Neural Network	With	0.92	0.92	0.92
	Without	0.75	0.76	0.77
Kernel SVM	With	0.96	0.95	0.96
	Without	0.95	0.95	0.95

Results for Data set 2

Classifier	**Preprocessing?**	**Accuracy**	**Recall**	**Precision**
Logistic Regression	With	0.99	0.99	0.99
	Without	0.99	0.99	1.00
Naïve Bayes	With	0.75	0.75	0.75
	Without	0.75	0.75	0.75
KNN	With	0.96	0.96	0.96
	Without	0.95	0.96	0.96
Linear SVM	With	0.59	0.59	0.59
	Without	0.74	0.74	0.74
Neural Network	With	0.96	0.96	0.96
	Without	0.82	0.82	0.84
Kernel SVM	With	0.99	0.99	0.99
	Without	0.98	0.98	0.98

Initially, we expected that a linear classifier such as logistic regression, Naïve Bayes, and linear SVM would work the best due to the binary nature of the problem. Based on the overall results, it seems that the nonlinear classifiers performed much better on average. However, logistic regression had the best results out of all the models. Additionally, we expected preprocessing steps to increase the model performance. For linear models, preprocessing either had no effect or decreased the model performance. On the other hand, preprocessing increased the model performance for nonlinear models.

After evaluating the results of the models, we recommend Logistic Regression without preprocessing to be the best model for classification of "tumor" images vs "no tumor" images.

Conclusion

Based on our results, we were successfully able to build a tumor recognition model for preprocessing and tumor identification. We had multiple image preprocessing steps which included resizing and cropping, removing noise, removing white edges, and standardization. The 6 models: Logistic Regression, Naïve Bayes, KNN, Linear SVM, Neural Network, and Kernel SVM were all trained to 80% and validated to 20% of the first data set. For nonlinear models, the prediction accuracies increased with the preprocessing. For linear models, the prediction accuracies decreased with preprocessing. Using the same trained models, they were all applied to the entirety of the second dataset, which yielded even better results than the first set. To have our models trained on one dataset and then have the prediction accuracies further improve in a completely different dataset proves that our preprocessing and tumor recognition was a success. Based on the results of this project, we would suggest using a logistic regression model without preprocessed images to classify MRI images. The model had a very high accuracy and recall on data it has not seen before, therefore it would likely perform similarly on real world data. Also, the model had less than a 1% false negative rate, meaning that we would mostly avoid the occurrence of a missed diagnosis. Physicians, medical professionals, and 3rd world countries with limited resources can hopefully use this model to accurately predict tumors from images that are inputted. Hopefully this will lighten the medical resource burden and better the lives of patients that it is intended to serve.

Team Member Contribution

1. Krista Radecke
   a. Created the code framework for downloading images, applying preprocessing steps, running models, and displaying images for reports.
   b. Responsible for resizing, cropping, and removing white edges.
   c. Ran Logistic Regression and Naïve Bayes.
   d. Contributed to writing reports.
2. Sandra Kim
   a. For preprocessing steps, responsible for rotation, translation, and standardization.
   b. In model creation, responsible for Neural Network and Kernel SVM.
   c. Also contributed to writing the project proposal and final report.
3. Sean Lee
   a. Created preprocessing steps for noise reduction and add mean image functions.
   b. In model creation, responsible for KNN and linear SVM.
   c. Contributed drafting, writing, and review of project proposal and final report.

References

(1) Brownlee, J. (2019, July 5). *Best practices for preparing and augmenting image data for cnns*. Machine Learning Mastery. Retrieved March 24, 2022, from https://machinelearningmastery.com/best-practices-for-preparing-and-augmenting-image-data-for-convolutional-neural-networks/

(2) Colangelo, M., & Kaminskiy, D. (2019). *Ai in medical imaging may make the biggest impact in healthcare*. HealthManagement. Retrieved March 24, 2022, from https://healthmanagement.org/c/hospital/issuearticle/ai-in-medical-imaging-may-make-the-biggest-impact-in-healthcare#:~:text=Using%20AI%20will%20reduce%20delays,medical%20images%20are%20not%20analysed

(3) World Health Organization. (n.d.). *Promoting cancer early diagnosis*. World Health Organization. Retrieved March 24, 2022, from https://www.who.int/activities/promoting-cancer-early-diagnosis#:~:text=Early%20diagnosis%20of%20cancer%20focuses,and%20higher%20costs%20of%20care