ISyE 6740
Classification Methods for Tumor Detection on MRI Images

Paul Telford, Hannah Pavlovich

April 27, 2024

Abstract

This paper looks at the Brain Tumor: Extracted Features for Brain Tumor [Ram25] dataset from kaggle and builds KNN, Support Vector Machine, Neural Network, Naive Bayes, Random Forest, and XGBoost models on the MRI images and different selections of the Extracted Features dataset, which interprets the images based on visual components described in this paper. The metrics were measured on accuracy and sensitivity, as the cost of missing a tumor is greater than the cost of finding a false tumor. Ultimately, we find that the Neural Network Model using Principal Component Selection on the Feature Set performed best on this dataset, while Naive Bayes performed the worst.

Contents

1 Introduction

In medicine, a brain **MRI (magnetic resonance imaging)** scan is a painless test that produces very clear images of the structures inside a patient's head — mainly, the brain. Healthcare providers use brain **MRIs** to evaluate, diagnose and monitor several types of medical conditions that affect the brain or other structures in the head. Currently, **MRI** is the most sensitive imaging test for the brain. To enhance image quality, a patient's brain is often injected with a ***contrast agent*** like **gadolinium;** a rare earth metal that alters the magnetic properties of nearby water molecules. This improves the **sensitivity** and **specificity** of the diagnostic images. The contrast material enhances the visibility of **tumors, inflammation, blood vessels, etc.**

In this project, we'll investigate the performance of some common **classification techniques** with features based on **first and second order statistics** for the detection of **brain tumors** using a data set comprised of features extracted from **MRI** images. The performance will be evaluated in terms of their performance **accuracy** and **sensitivity.**

We'll also apply the same classification models to the **MRI** image files and compare these results with the results obtained from the statistical **features**

2 Description of Data

The dataset is taken from kaggle: *Brain Tumor: Extracted features for brain tumor*[Boh20], which includes data from 1,644 **MRI** image scans. The .csv data file consists of five first-order (textural) feature columns, eight second-order feature columns, and the class (target) column.

It is important to note at this juncture that we did not pre-process the raw **MRI** images ourselves. This process would typically involve the following steps:

1. **Grayscale conversion -** RGB images contain unnecessary information that requires a lot of storage space

2. **Filtering -** for elimination of common noises like salt and pepper and speckle in grayscale images by applying filtering techniques like Gaussian, mean, and median filter.

3. **Feature Extraction -** shortening the number of resources needed to define a big group of data correctly. This is necessary because analysis of a huge number of variables requires more memory and computation time. The features that we are going to use for this exercise are typically extracted by way of a **Gray Level Co-occurrence Matrix (GLCM).** A **co-occurrence matrix** or **co-occurrence distribution** is defined over an image to be the distribution of co-occurrence values at a given offset. Meaning, values in the matrix represent the distance and angular spatial relationship over an image sub-region of specific size. Moreover, The **GLCM** calculates how often a pixel with gray-level (grayscale intensity or tone) value i occurs either horizontally, vertically, or diagonally to adjacent pixels with the value j. Once the **GLCM** is created, several statistics similar to the ones we'll use for this exercise can be derived using different formulas.

2.1 First-Order Features

Below is a description of the first-order feature columns contained in the raw data set.

	Mean	Variance	Standard Deviation	Skewness	Kurtosis
count	3762.0000	3762.0000	3762.0000	3762.0000	3762.0000
mean	9.4889	711.1011	25.1823	4.1027	24.3891
std	5.7280	467.4669	8.7735	2.5609	56.4347
min	0.0787	3.1456	1.7736	1.8860	3.9424
25%	4.9824	363.2255	19.0585	2.6202	7.2529
50%	8.4775	622.5804	24.9516	3.4222	12.3591
75%	13.2127	966.9543	31.0959	4.6517	22.6403
max	33.2400	2910.5819	53.9498	36.9313	1371.6401

Table 1: Description of First Order Features

- **Variance** is a measure of the histogram width which represents the deviation of gray levels from the mean.
- **Skewness** is a measure of the degree of histogram asymmetry around the mean, and
- **Kurtosis** is a measure of the histogram sharpness and is given by:

$$Kurtosis \ = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\{ \left[\frac{P_{i,j} - \sigma}{\sigma} \right]^4 \right\} - 3$$

  Here, $P_{i,j}$ is the pixel value at point (i, j) and m is the mean.
- **Standard Deviation** is the square root of the variance and, therefore, also measures deviation of gray levels from the mean.

2.2 Second-Order Features

Below is a description of the second-order feature columns contained in the raw data set.

	Entropy	Contrast	Energy	ASM	Homogeneity	Dissimilarity	Correlation	Coarseness
count	3762.0000	3762.0000	3762.0000	3762.0000	3762.0000	3762.0000	3762.0000	3762.0000
mean	0.0736	127.9615	0.2047	0.0586	0.4793	4.6985	0.9558	0.0000
std	0.0703	109.4996	0.1294	0.0583	0.1279	1.8502	0.0262	0.0000
min	0.0009	3.1947	0.0247	0.0006	0.1055	0.6811	0.5494	0.0000
25%	0.0069	72.1252	0.0696	0.0048	0.3650	3.4124	0.9471	0.0000
50%	0.0666	106.7374	0.2255	0.0508	0.5126	4.4824	0.9616	0.0000
75%	0.1133	161.0590	0.2989	0.0893	0.5756	5.7238	0.9714	0.0000
max	0.3945	3382.5742	0.5897	0.3477	0.8109	27.8278	0.9900	0.0000

Table 2: Description of Second-Order Features

- **Entropy -** the degree of uncertainty in a random variable is termed entropy.

$$Entropy \ = \sum_{i,j=0}^{n-1} P_{i,j} \log(P_{i,j})$$

- **Contrast -** splits the darkest and brightest area of an image. It is calculated using the following formula:

$$Contrast = \sum_{i,j=0}^{n-1} P_{i,j}(i-j)^2$$

- **Energy -** used in GLCM to calculate the total number of squared elements. Energy measures homogeneity. A high level of energy indicates that the image has excellent homogeneity or pixels of an image are very similar.

$$Energy = \sqrt{\sum_{i,j=0}^{n-1} P_{i,j}^2}$$

- **Angular Second Moment (ASM) -** represents the uniformity of distribution of gray level in the image.

$$ASM = \sum_{i,j=0}^{n-1} P_{i,j}^2$$

- **Homogeneity -** the quality or state of being homogeneous.

$$Homogeneity = \sum_{i,j=0}^{n-1} \frac{P_{i,j}}{1 \ + \ (1,j)^2}$$

- **Dissimilarity -** a linear measure of local variations in an image.

$$Dissimilarity = \sum_{i,j=0}^{n-1} P_{i,j}|i-j|$$

- **Correlation -** calculated as the correlation coefficient between -1 and +1.

$$Correlation = \sum_{i,j=0}^{n-1} P_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sigma^2}$$

- **Coarseness -** a measure of grey level contrast that is used to establish descriptors of relative smoothness.

$$Coarseness = \sum_{i,j=0}^{n-1} 1 - \frac{1}{1-s^2}, \quad s \ = \ standard \ deviation$$

2.3 Binary Response Variable

Below is a brief overview of the binary response variable (1 = Tumor, 0 = Non-Tumor)

class	count
0	2079
1	1683

3 Exploratory Data Analysis

Before proceeding any further, we must point out that any good **regression analysis** requires investigating the ***feature variables*** against the **four assumptions;** linearity/zero mean, constant variance, independence, and normality.

- **Linearity/Zero mean Assumption** means that the expected value of the errors is zero across all errors; this also implies that the linearity assumption holds.¡br¿
- **Constant Variance Assumption** means that it cannot be true that the model is more accurate for some parts of the population and less accurate for other parts. A violation of this assumption means that the estimates are not as efficient in estimating the true parameters, resulting in poorly calibrated confidence and prediction intervals
- **Independence Assumption** means that the response variables are independently drawn from the data-generating process. Violation of this assumption can lead to a misleading assessment of the strength of the regression.
- **Normality Assumption** means the error terms are normally distributed. If violated, hypothesis tests and confidence or prediction intervals can be misleading.

3.1 Correlation

Some graphical methods like a **correlation plot** and **histogram** can give us some idea as to whether the assumptions are being violated and point us in the direction of possibly **transforming** some of the ***feature variables*** or perhaps which **feature selection** techniques we may need to deploy to address possible issues like ***multicollinearity*** or ***inflated statistical significance.***

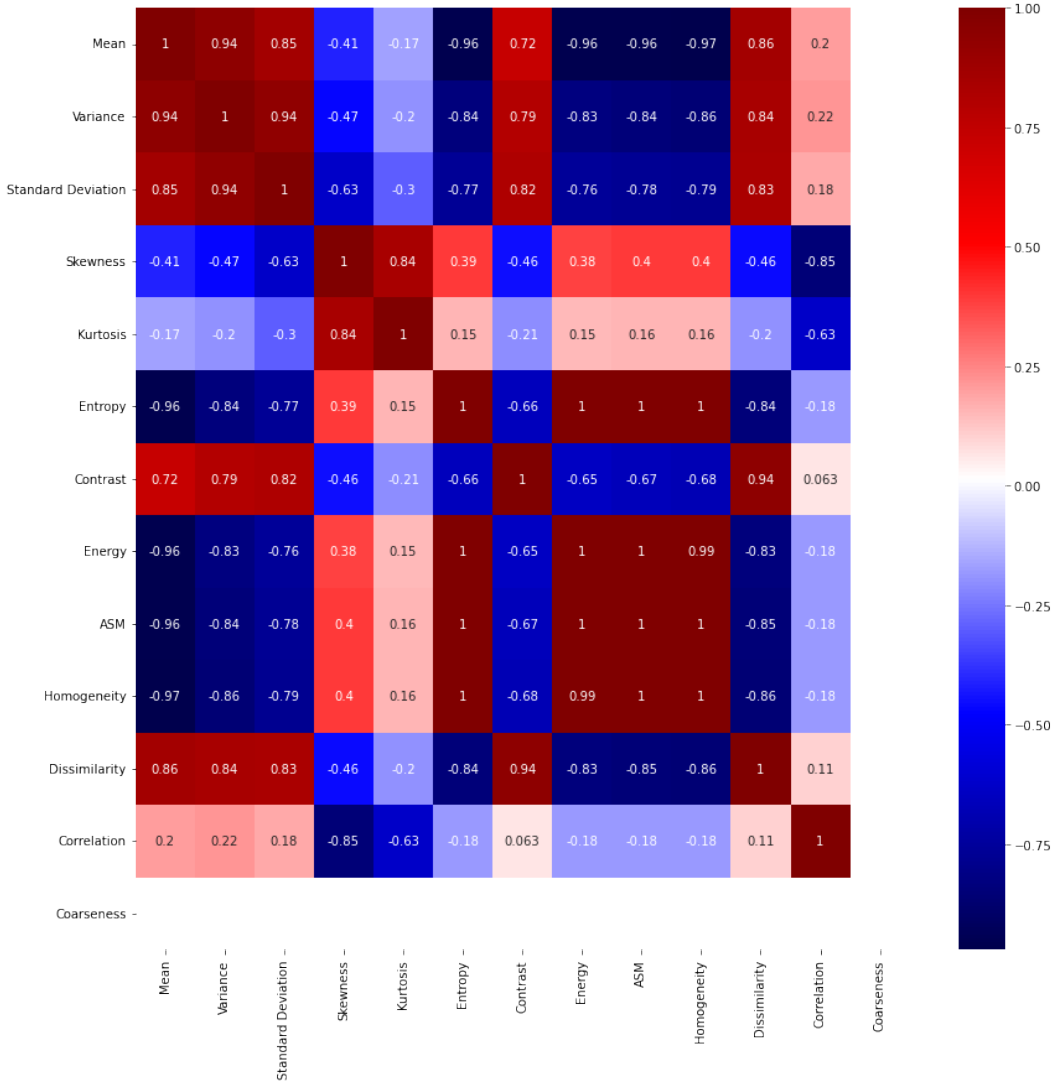

Figure 1: Correlation Matrix

Based on this correlation plot, (figure 1) we can begin speculating about multicollinearity. Multicollinearity is addressed in Variable Selection (section 4, mainly through Elastic Net variable selection

3.2 Histograms

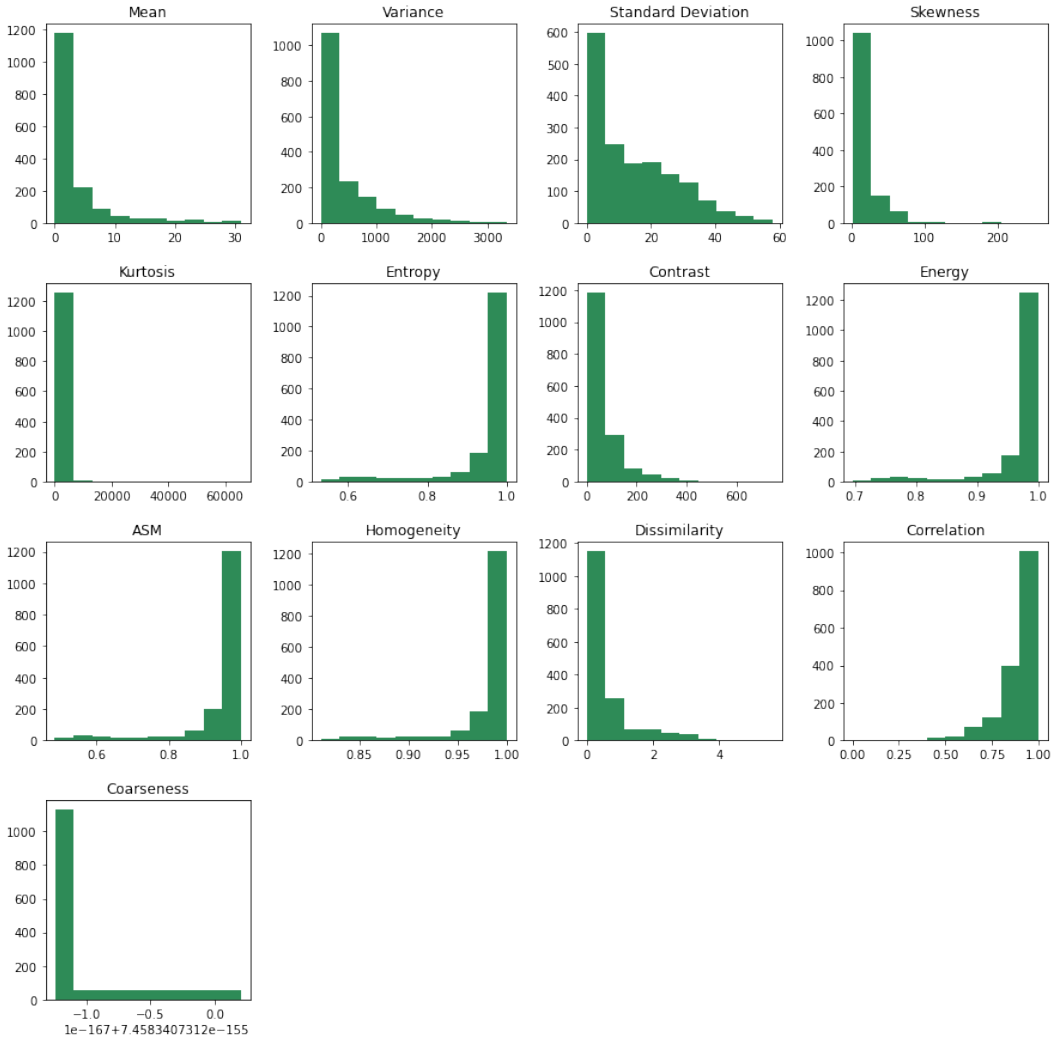

Figure 2: Histograms of variables

From the histograms above (figure 2), we see that there may be issues with the normality assumption. All variables have large skews to the left side for variables that correspond to low types of deviation: Mean, Variance, Skewness, etc, and to the right for variables where largest value 1 describes low deviation: Energy, Homogeneity, Entropy, etc.

These plots also show the difference in scales between the variables.

3.3 ISOMAP

Figure 3 below depicts an Isometric Mapping (ISOMAP) of the MRI images by way of Principal Components Analysis (PCA). This mapping shows how the images are related to each other as it demonstrates how the brain scans can be separated by size and darkness. While the right most portion may show tumors with the white spots in the middle, this ISOMAP is not reliable in determining if their is a tumor present or not.

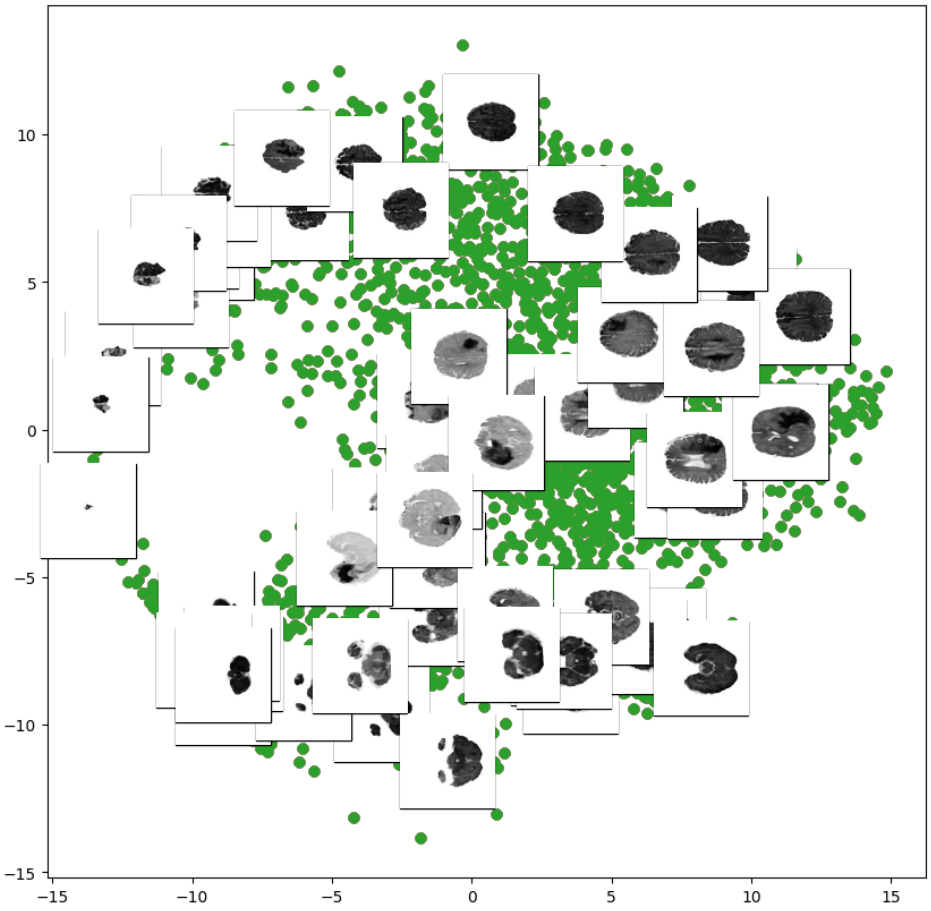

Figure 3: PCA Image Recognition

4 Variable Selection

Principal Components Analysis (PCA) and Elastic Net are performed to identify the most important amongst the 13 first and second order statistical features that were extracted from the raw image files [Ram25]. We performed these variable selection techniques to avoid some of the issues that would arise from violating the **four assumptions** by modeling all the variables together. In addition to building models with the most important variables identified through the PCA and Elastic Net selection processes, we'll also build models by utilizing the top four Principal Components. These top four Principal Components were found to explain as much as 91.32% of the variability in the statistical features data.

4.1 Image Files (img)

The dataset includes image files with a classification as 0: no tumor, 1: tumor. The images are reduced to 25% of their original size and then scaled to reduce computational time. The image size reduction was especially needed for tuning and training Neural Network models, which are traditionally expensive to run.

4.2 Raw image files with PCA (img-pca)

The PCA on the raw image files is seen in Figure 4. Per the image, 10 Principal Components are selected. While this is a large number of PC's, it explains less than 80% of the data.

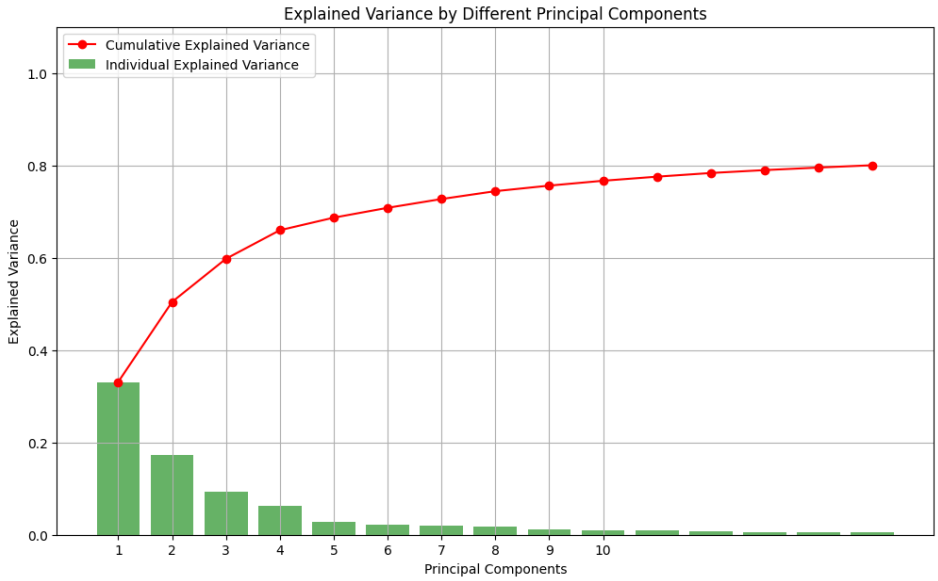

Figure 4: Explained Variance in PCA on Image Dataset

4.3 PCA and Component Selection (pca)

The Dataset distilled to features is next considered. PCA is run on the dataset, as described in Figures 5 and 6. From these images, the top four principal components are chosen to describe the dataset moving forward under this selection.

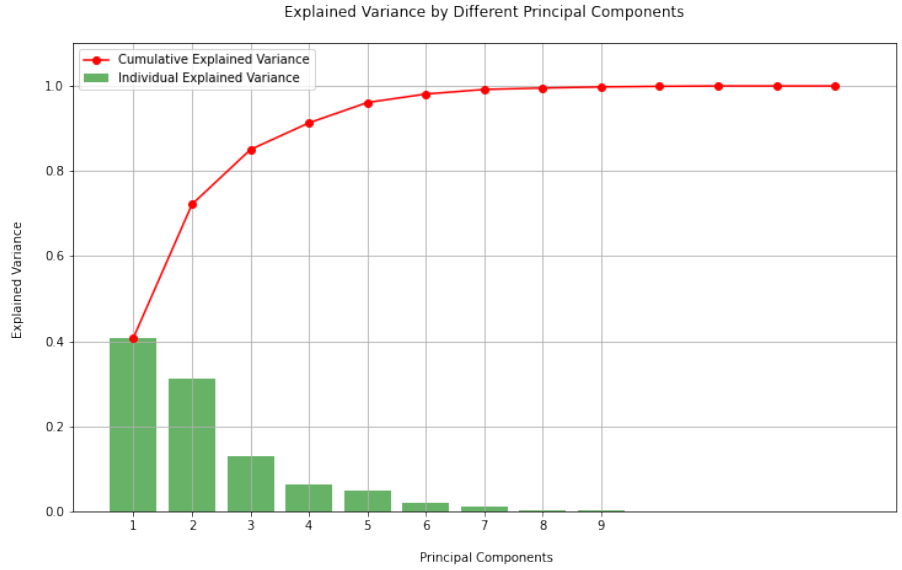

Figure 5: Features PCA Variance Explained

Figure 6: Total Variance Explained by First Four Principal Components

4.4 Feature Selection with PCA (pca-select)

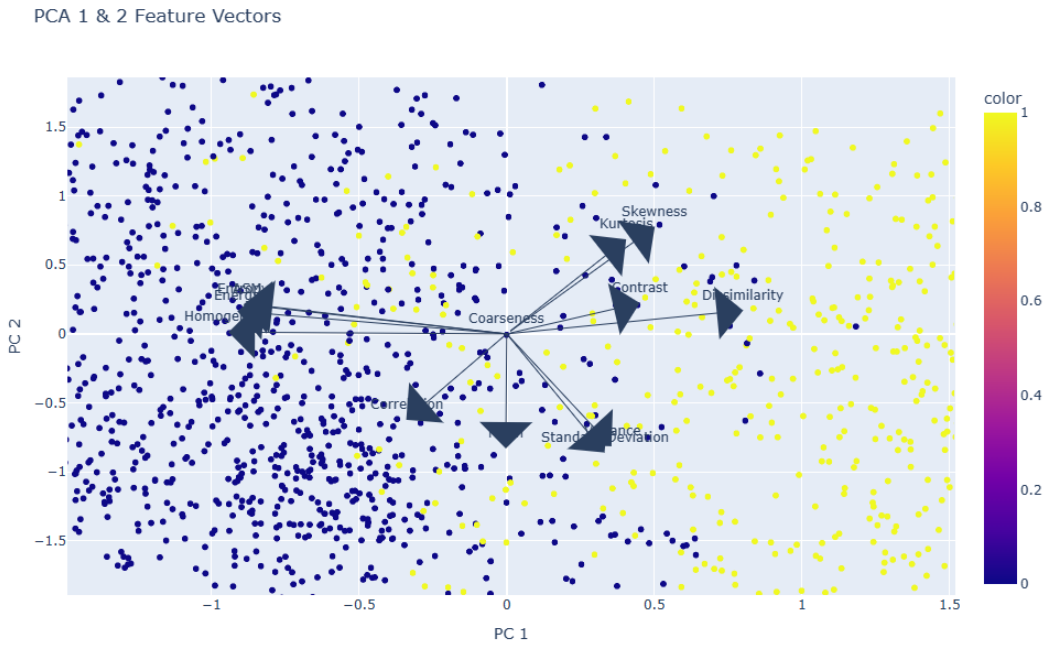

Figure 7: Feature Vectors for the First Two Principal Components

The **top four principal components** account for as much as 91.32% of the **explained variation.** In continuation of the Feature Dataset with PCA variable selection (section 4.4 Figure 7) displays the importance of the features by the length of their vector arrows. From this image and reading of the top features per component, the five features that will be selected for use in this analysis are: *Energy, Skewness, Homogeneity, Dissimilarity, and Mean* which contribute strongly to the first four principal components in addition to the second.

4.5 Feature Selection with Elastic Net (en-select)

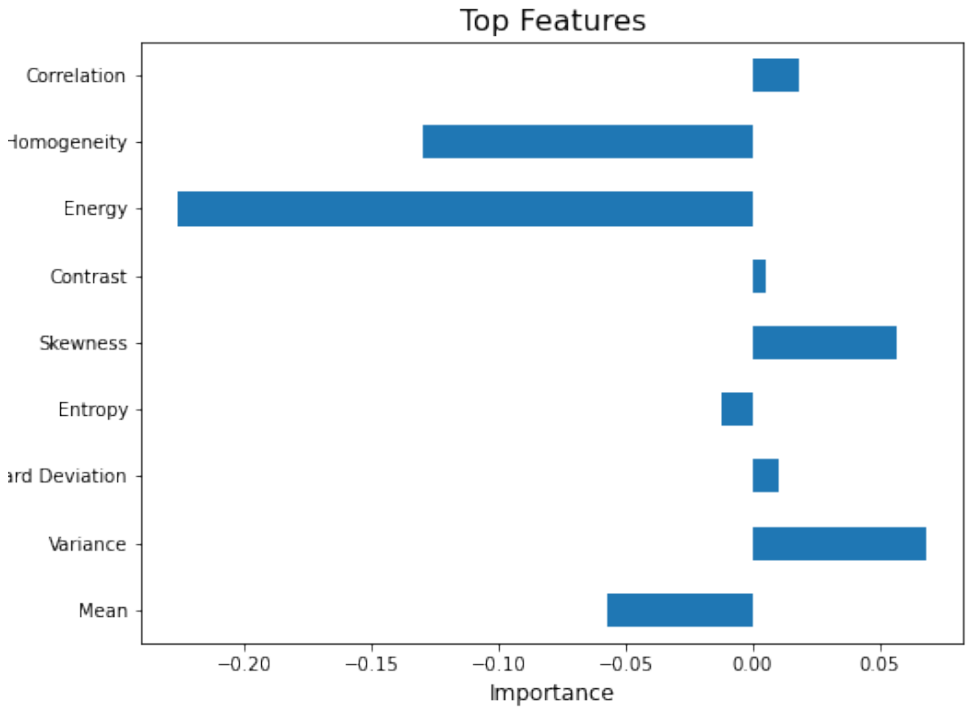

Figure 8: Importance of Feature Vectors per ElasticNet

The last feature selection is done using Elastic Net, which was performed using the sklearn package. We tuned the penalty constant (α) and the ElasticNet mixing ratio (l1_ratio) using a five-fold cross validation grid search. Using ROC and AUC as the scoring mechanism, we obtained the top five variables by their importance on the final model, see in Figure 8. Per this diagram, the variables chosen by this method are: *Homogeneity, Energy, Skewness, Variance, and Mean.*

5 Methodology

The goal of this project is to find the classification model that most accurately predicts whether an **MRI** image contains a tumor or not. The data provided, as described in Section 2, are statistical features of the images themselves, as described by Ramtekkar, Pandey, Pawar [Ram25]. Their paper selects the five "first-order" features to describe the model, while the remaining 8 are second-order features.

This project aims to train and tests models using these statistical features after performing variable selection techniques, and models using the images themselves. Each model is created with the selections as described in Section 4, and then split into 80/20 training and testing and cross validated 10 times to achieve the tuning results. The models are as follows:

5.1 K-Nearest Neighbors

K-Nearest Neighbors is a good classification method because of its ease of use. In general, it is not computational expensive, unless applied to high dimensional data. The datasets created in Section 4 reduce the number of features, allowing it to avoid overfitting and high complexity with KNN. **Five-fold cross validation** was performed on the datasets to determine optimal k for each set.

The results of KNN after tuning for each data selection is in Table 3: KNN Tuning Results. Images of the tuning are in the Appendix: Table 9.

model	k	testing score	training score
img	6	0.9349	0.9435
img-pca	8	0.9110	0.9083
pca	4	0.9960	0.9791
pca-select	2	0.9123	0.9358
en-select	24	0.9933	0.9751

Table 3: KNN Tuning Results

Models pca and en-select both have very high training scores, which could be due to overfitting. The testing scores for both are similarly high, though, which may suggest a good model rather than overfitting. In Results, Section 6, we use the k-values shown in Table 3

5.2 Support Vector Machines (SVM), Kernel and Linear

To tune the SVM models, **Five-fold cross validation** was performed to determine the best hyper-parameter for the datasets. Support Vector Machines are good models classification because of the malleable separator and ability to work with complex data. Using GridSearchCV, the models were tuned for both 'rbf' and 'linear'. All models except en-select performed better with Kernel than Linear, suggesting a non-linear model is more appropriate for this dataset. For this reason, the paper from this point on will only be using SVM-Kernel and not SVM-Linear.

The results are below in Table 4. Almost all of the models show overfitting with the excepts of img-pca and pca-select, suggesting that PCA is near-essential before building a Support Vector Machine. The images in Table **??** also demonstrate overfitting as C approaches 100. There is not consistency across the models in regar to the gamma parameter.

GridSearchCV() did not provide the best parameters for these models; instead the visualization must be used. In SVM, the higher value C may lead to overfitting, while a smaller C creates low variance/high bias. This is evident in the images, where when C reaches 100, almost all of the accuracies are very high. The only exception is the img model,

	C	gamma	kernel	testing score	training score
img	100	0.001	rbf	0.9501	1
img-pca	100	0.001	rbf	0.9022	0.9365
pca	10	0.1	rbf	0.9741	0.9888
pca-select	100	0.1	rbf	0.9567	0.9624
en-select	10	0.0001	linear	0.9805	0.9809

Table 4: Tuning Parameters for SVM

Using the table above and the images in Table **??** in the Appendix, the optimal parameters for img, pca, and en-select are changed. The training scores in all cases are high and could indicate over fitting. The new variables are below in Table 5.

	C	gamma	kernel	testing score	training score
img	1	0.001	rbf	0.9179	0.9519
img-pca	100	0.001	rbf	0.9022	0.9365
pca	10	0.0001	rbf	0.9649	0.9657
pca-select	100	0.1	rbf	0.9567	0.9624
en-select	10	0.0001	rbf	0.9589	0.9591

Table 5: Updated Tuning Parameters for SVM

5.3 Neural Networks

Neural Networks is a good classification model that is tolerant to noisy data and flexible. **Five-fold cross validation** was performed on the datasets to determine the most effective number of hidden layers, regularization parameter, and non-linear activation function. The MLPClassifier() function from sci-kit with GridSearchCV(). The validation is performed on all 5 datasets, with the tuning results in Table 6. Images of tuning parameters are in Table **??**

The table shows that the most common hidden layer size is (30, 20, 15), and the largest training score is for raw images. This combination may be a result of overfitting, as the mean test score is much less than that of the train score. Alternatively, with elastic net, the train and testing scores are equally high.

	activation	alpha	hidden layer sizes	mean test score	mean train score
img	relu	0.0001	(40, 35)	0.9495	0.9983
img-pca	relu	0.0778	(40, 35, 20)	0.9063	0.9457
pc	relu	0.0667	(30, 20, 15)	0.9784	0.9785
pca-select	relu	0.0334	(30, 20, 15)	0.9412	0.9437
en-select	relu	0.0223	(30, 20, 15)	0.9801	0.9814

Table 6: Neural Net Hyperparameter Tuning

5.4 Naive Bayes

We implemented the Gaussian Naive Bayes algorithm for classification where we assumed that the likelihood of the features to be Gaussian. The **Prior** probabilities of the classes parameter was left at its default value so that the priors were adjusted according to the data. The **variance smoothing** parameter was changed from its default value of $1 \times e^{-9}$ to $1 \times e^{-3}$. This parameter specifies the portion of the largest variance of all features that is added to variances for calculation stability. The performance results can be viewed in Table 6 and the confusion matrices are also available in the 8.2

5.5 Random Forest

The method to produce the RandomForest Model is different from the others because the Random Forest does not require variable selection. It is robust to multicollinearity, which is seen in the dataset, and thus an appropriate choice for testing. Thus, this model is performed on the raw image dataset and full feature set.

GridSearchCV() was performed using **Five Fold Cross Validation** on both datasets. The parameters are below in Table 7. Again, we see a model with very high mean train score and similarly high test score. Random Forest can be prone to overfitting, so the model with the highest train score is not necessarily the best. Moving forward in building the Random Forest model, **image dataset parameters: max_depth=40, n_estimators=200** and **feature dataset parameters: max_depth=40, n_estimators=200** are used for training the model.

	max depth	n estimators	mean test score	mean train score
img	8	200	0.9870	0.9958
features	9	400	0.9166	0.9869

Table 7: Random Forest Tuning Results

5.6 XGBoost

Gradient Boosting is a similarly robust model when multi-collinearity could be a problem in classification methods. Like Random Forest (section 5.5, variable selection is not needed on the XGBoost method, and the model is performed on the raw image dataset and full feature set.

Because XGBoost has many parameters, RandomizedSearchCV() is a better method for choosing the parameters. All of the scores have a high train score and high test score, as was also evident in the Random Forest model. The result with highest testing score and not highest training score is selected. Moving forward in building the Random Forest model, **image dataset parameters: max_depth=40, n_estimators=200** and **feature dataset parameters: max_depth=40, n_estimators=200** are used for training the model.

	subsample	max_depth	gamma	colsample_bytree	mean_test_score	mean_train_score
img	1	5	5	0.6	0.9136	0.9759
features	1	3	5	1	0.9840	0.9884

Table 8: XGBoost Results

6 Results

Model	Data Type		Accuracy	Senitivity
Neural Net	Images	img	0.8924	0.9713
		img-pca	0.9283	0.9045
	Extracted Features	pc	0.9920	0.9904
		pc-select	0.9655	0.9427
		en-select	0.9894	0.9904
SVMK	Images	img	0.9097	0.9140
		img-pca	0.9323	0.9140
	Extracted Features	pc	0.9907	0.9777
		pc-select	0.9655	0.9554
		en-select	0.9841	0.9618
Naive Bayes	Images	img	0.7649	0.6497
		img-pca	0.7477	0.4650
	Extracted Features	pc	0.9602	0.9235
		pc-select	0.6614	0.3201
		en-select	0.9562	0.9150
KNN	Images	img	0.9349	0.8917
		img-pca	0.9110	0.8471
	Extracted Features	pc	0.9960	0.9904
		pc-select	0.9124	0.8185
		en-select	0.9934	0.9873
Random Forest	Images		0.9283	0.9219
	Extracted Features		0.9870	0.9765
XGBoost	Images		0.9880	0.9735
	Extracted Features		0.9880	0.9735

Table 9: Metrics from All Models

The final metrics for all of the models and datasets are above in Table 9. The models performed differently depending on the dataset used in building and testing the model. In general, the images themselves did not perform as well as the features that came from the extracted features. While the NeuralNet and SVM-K models returned high sensitivities for the images, the accuracies were much lower, making this not an optimal dataset, either raw or with PCA.

The Neural Network Model had better sensitivity results than KNN, which is important to consider as the cost of not spotting a tumor is much larger than spotting a tumor that is not there. The least accurate was the Naive Bayes model, except with pc model. Naive Bayes is mainly used when the features are independent from one another. In this dataset, the features are highly correlated, which could explain its underperformance.

Amongst the selected features, the Elastic Net Method consistently performed better than the features selected through PCA. Elastic Net is used generally where multicollinearity is present, which could explain why this variable selection performed well.

Using the Top Four Principal Components also produced high metric for all models, especially in KNN and Neural Networks.

7 Conclusion and Findings

In this project both the raw image files and the extracted features, performed by Ramekka, Pandey and Pawar [Ram25], were used. Ultimately, the extracted features in general performed better than the images themselves. The image processing performed made for a better dataset. The dataset had numerical values for the shifting in images, taking away a layer of guessing the model has to do with the raw images.

The high scoring of the Principal Component Selection could indicate that each feature does not contribute in a significantly independent way, thus the creation of the component can better describe the dataset. With the high performance of the Elastic Net Selection confirms, we can infer that there is high multicollinearity in this model, as presented in EDA (section 3).

As a surprise, the CART methods did not outperform the other classification methods. Before running the models, the CART methods seemed to be the best option, as they could use all features, and, in general, are simple to run. The tuning did help with problems of overfitting. These methods may perform better on a larger dataset, so the full advantages of using Random Forest or XGBoost are not evident in this dataset.

The Naive Bayes model did not perform well on this dataset with the exception of the Elastic Net selection and Principal Component models. This leads us to infer that there is multicollinearity in the model, as Naive Bayes performs better when the features are independent. In the variable selection, the collinearity was reduced, which could have contributed to a better performance on the reduced dataset.

Ultimately, the Neural Network Model run on the Top Four Principal Components is the best selection to explain this model. While it is computationally expensive to run, the results are consistently better than the other models.

8 Appendix

8.1 Tuning Parameters

K-Nearest Neighbors

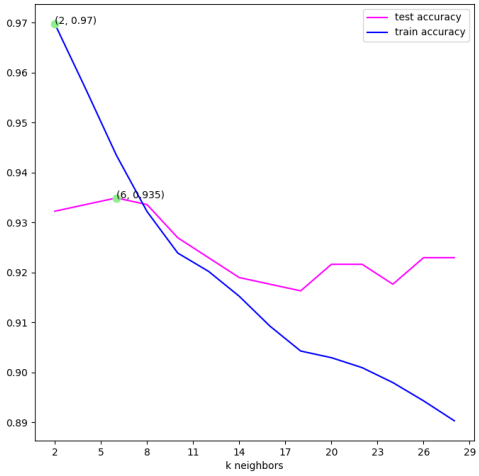

Figure 9: KNN Tuning - n_neighbors - img

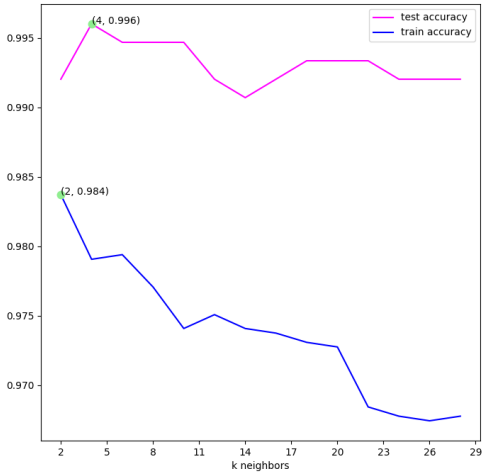

Figure 11: KNN Tuning - n_neighbors - pc

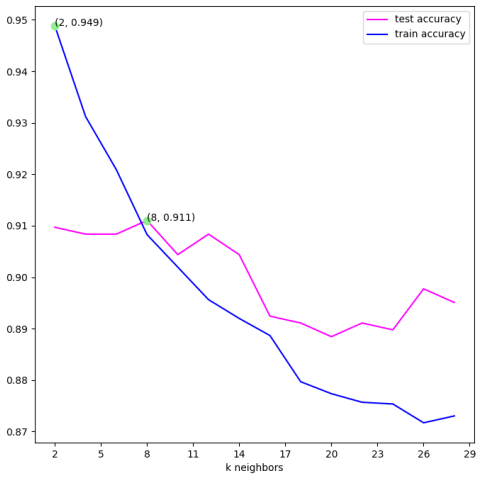

Figure 10: img-pca

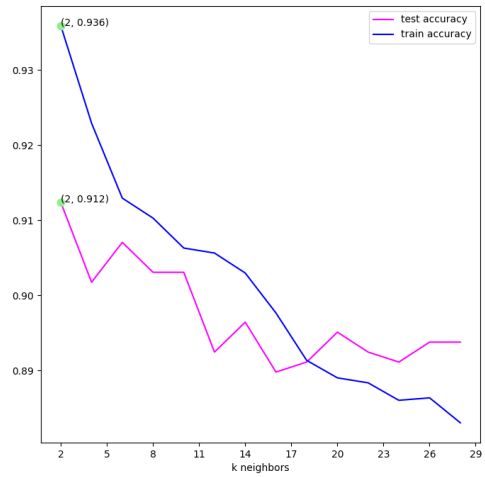

Figure 12: KNN Tuning - n_neighbors - pca-select

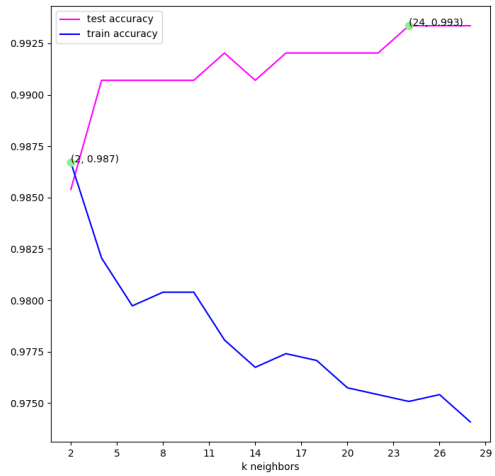

Figure 13: KNN Tuning - n_neighbors - en-select

Kernel - SVM

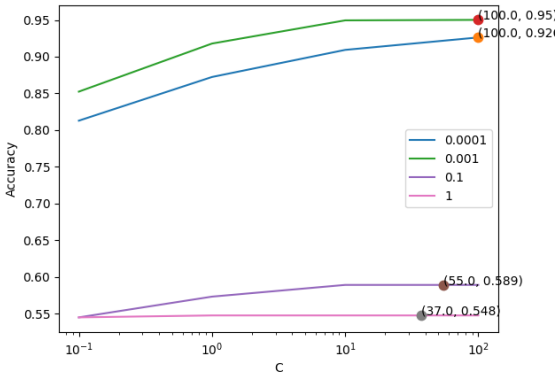

Figure 14: SVM Tuning - C, gamma - img

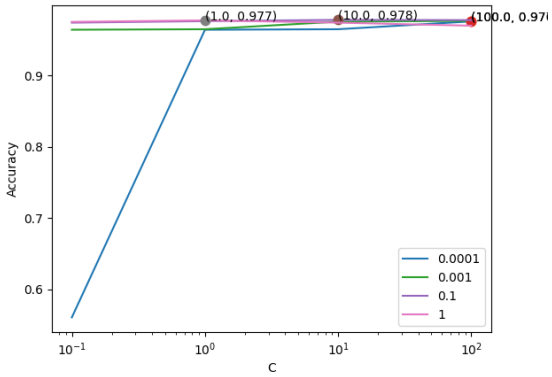

Figure 16: SVM Tuning - C, gamma - pc

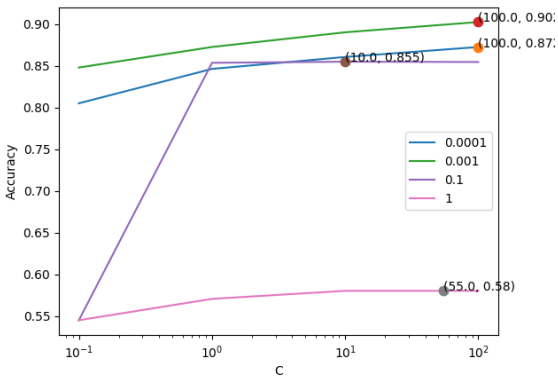

Figure 15: SVM Tuning - C, gamma - img-pca

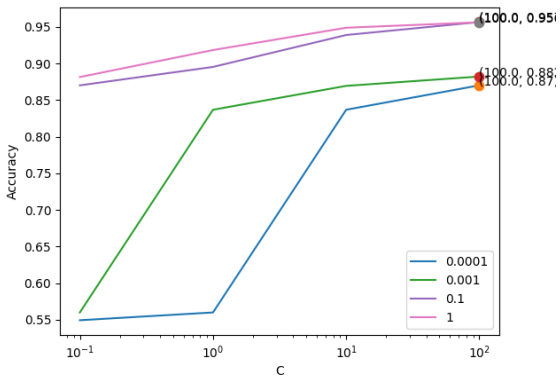

Figure 17: SVM Tuning - C, gamma - pca-select

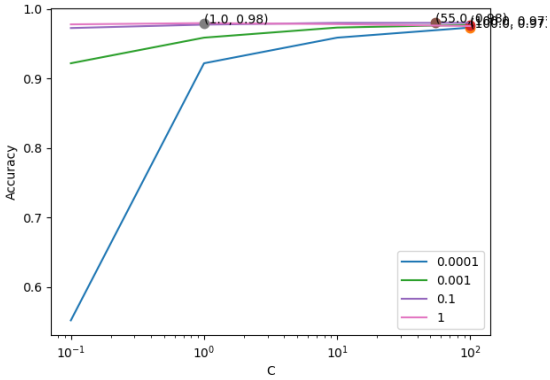

Figure 18: SVM Tuning - C, gamma - en-select

Neural Networks

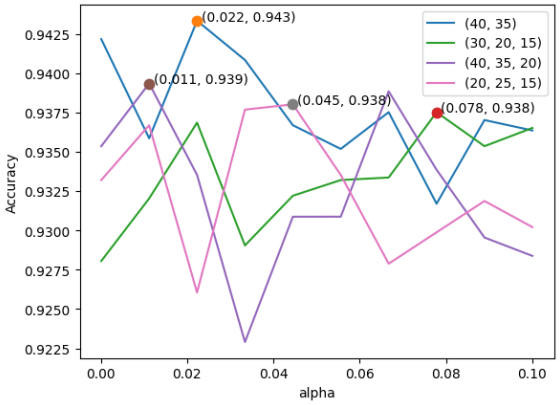

Figure 19: NN Tuning - alpha, layers - img

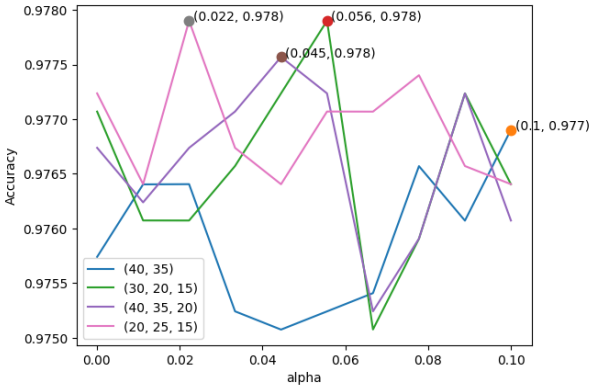

Figure 21: NN Tuning - alpha, layers - pc

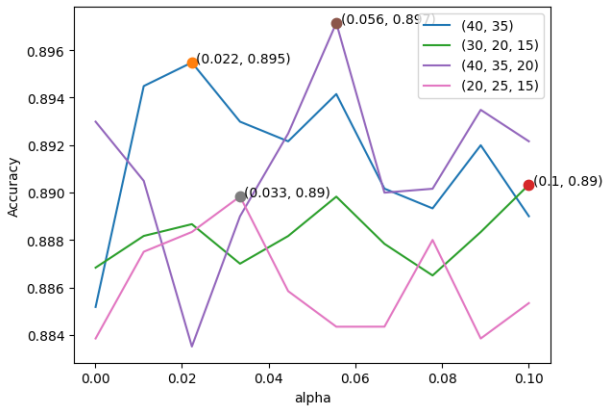

Figure 20: NN Tuning - alpha, layers - img-pca

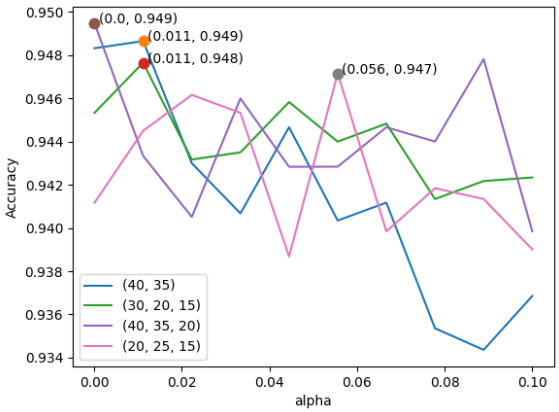

Figure 22: NN Tuning - alpha, layers - pca-select

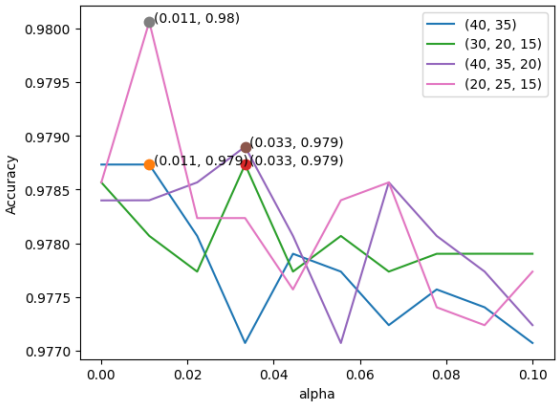

Figure 23: NN Tuning - alpha, layers - en-select

Random Forest

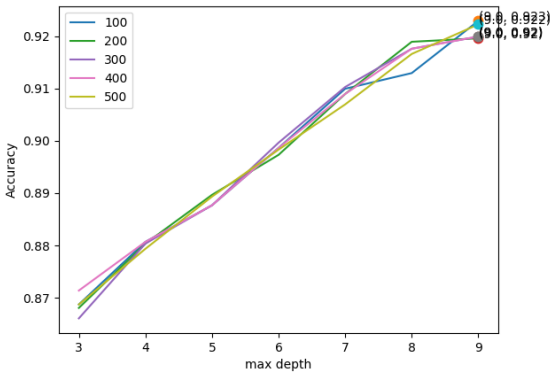

Figure 24: RF Tuning - max depth, ntrees - img

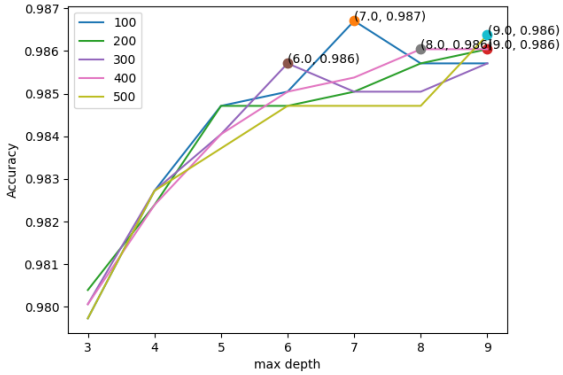

Figure 25: RF Tuning - max depth, ntrees - full

8.2 Confusion Matrices

K Nearest-Neighbors

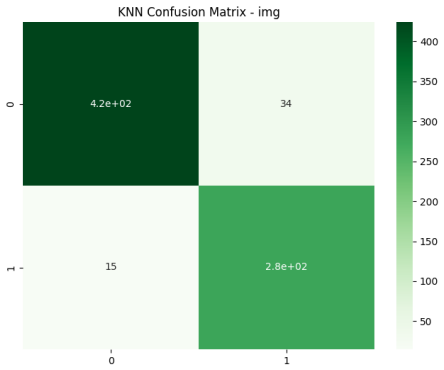

Figure 26: KNN Confusion Matrix - img

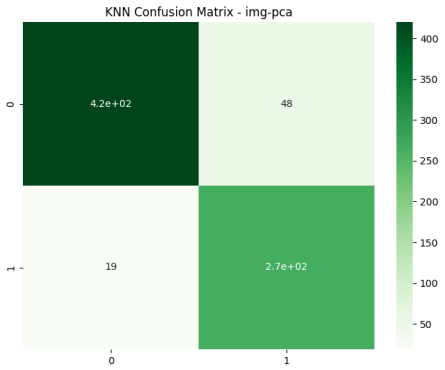

Figure 27: KNN Confusion Matrix - img-pca

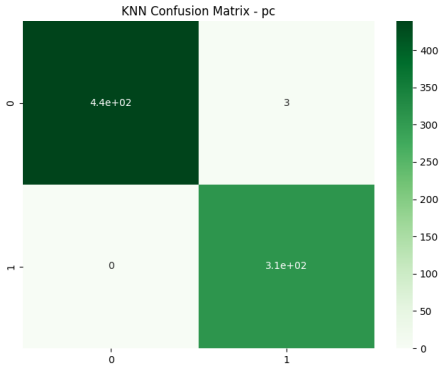

Figure 28: KNN Confusion Matrix - pc

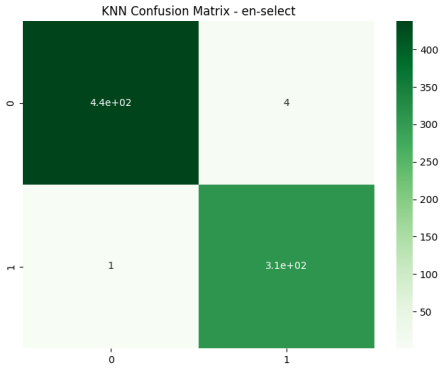

Figure 30: KNN Confusion Matrix - en-select

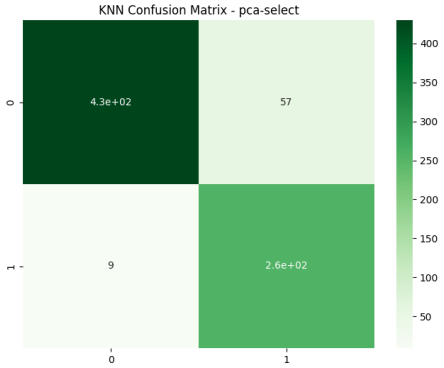

Figure 29: KNN Confusion Matrix - pca-select

Kernel - SVM

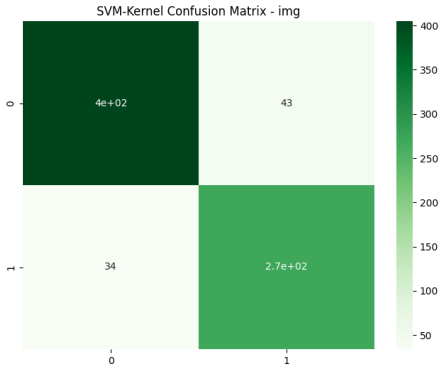

Figure 31: SVM-K Confusion Matrix - img

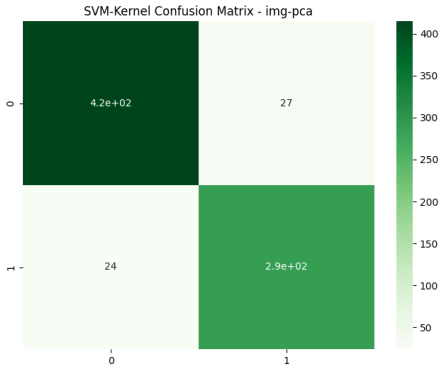

Figure 32: SVM-K Confusion Matrix - img-pca

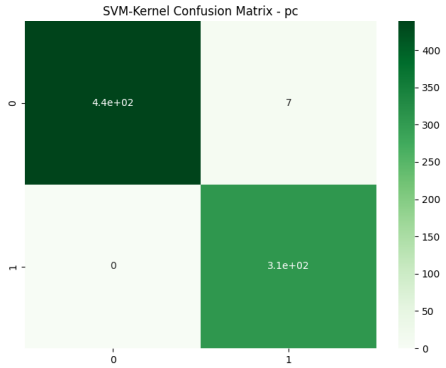

Figure 33: SVM-K Confusion Matrix - pc

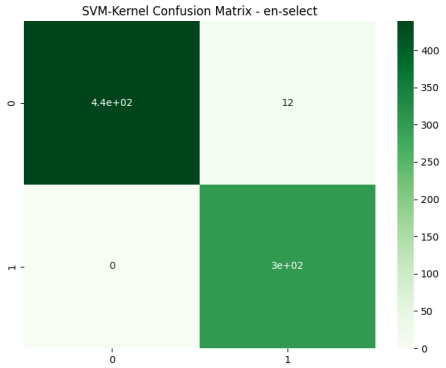

Figure 35: SVM-K Confusion Matrix - en-select

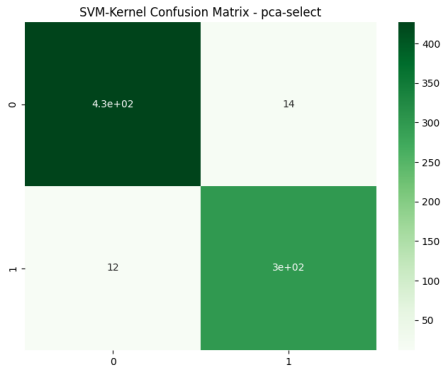

Figure 34: SVM-K Confusion Matrix - pca-select

Neural Networks

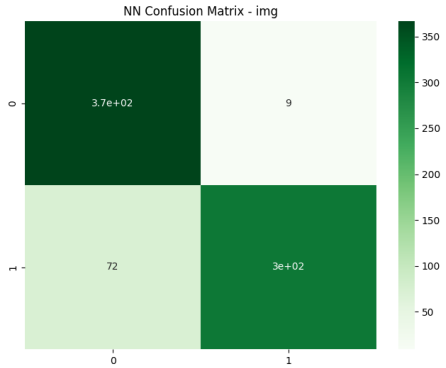

Figure 36: Neural Net Confusion Matrix - img

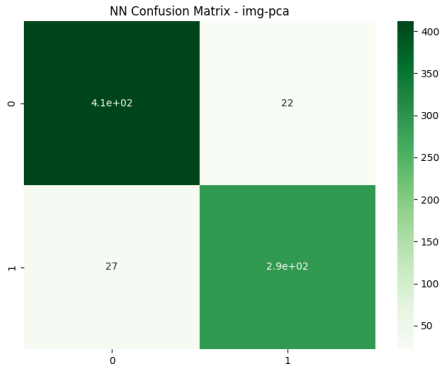

Figure 37: Neural Net Confusion Matrix - img-pca

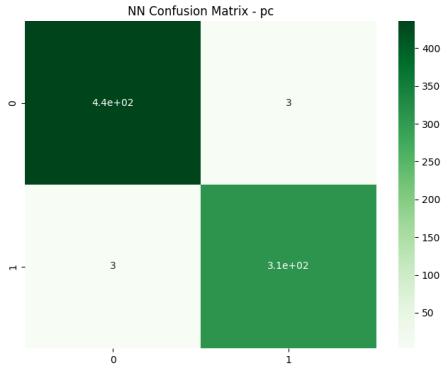

Figure 38: Neural Net Confusion Matrix - pc

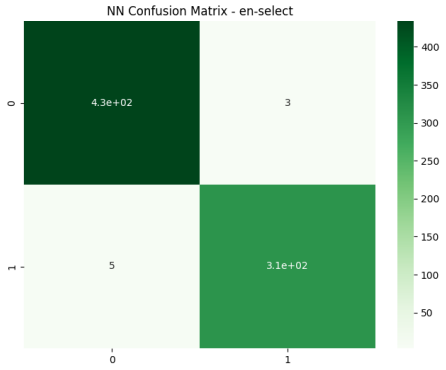

Figure 40: Neural Net Confusion Matrix - en-select

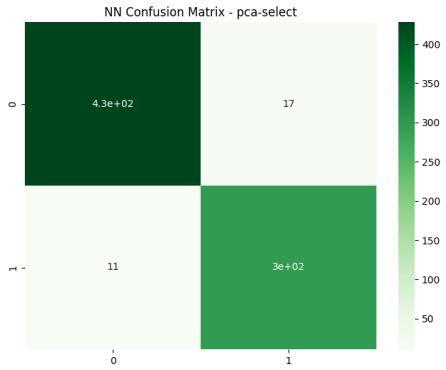

Figure 39: Neural Net Confusion Matrix - pca-select

Naive Bayes

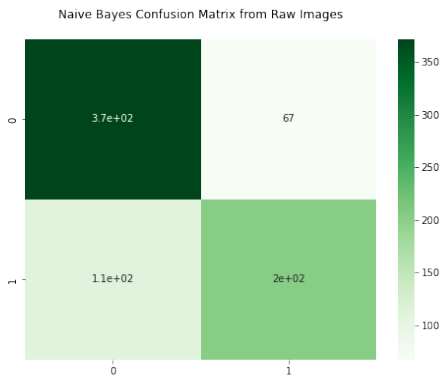

Figure 41: Naive Bayes Confusion Matrix - img

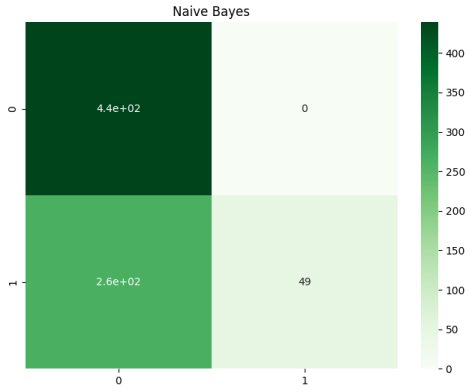

Figure 42: Naive Bayes Confusion Matrix - img-pca

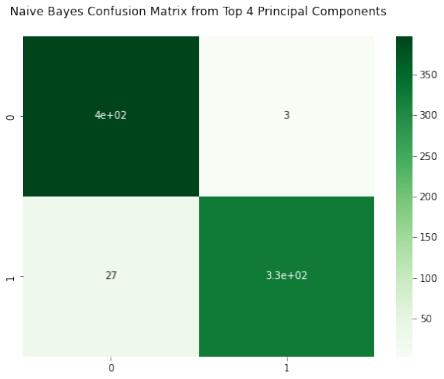

Figure 43: Naive Bayes Confusion Matrix - pc

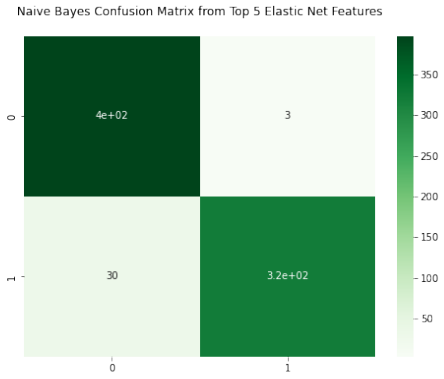

Figure 45: Naive Bayes Confusion Matrix - en-select

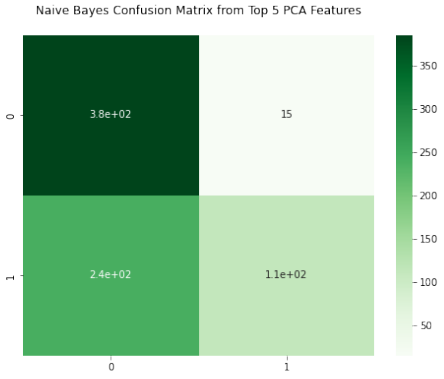

Figure 44: Naive Bayes Confusion Matrix - pca-select

Random Forest

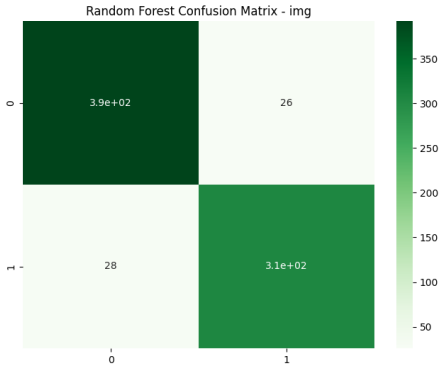

Figure 46: Random Forest Confusion Matrix - img

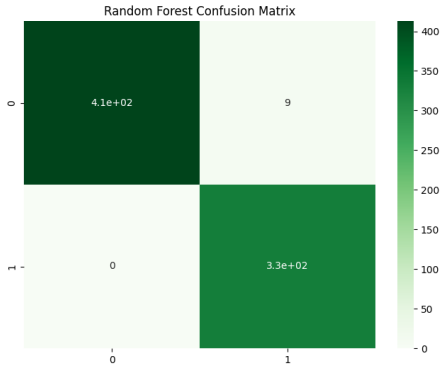

Figure 47: Random Forest Confusion Matrix

XGBoost

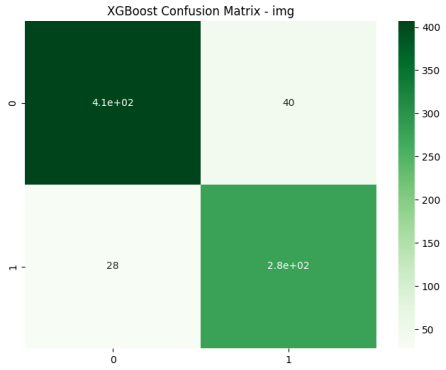

Figure 48: XGBoost Confusion Matrix - img

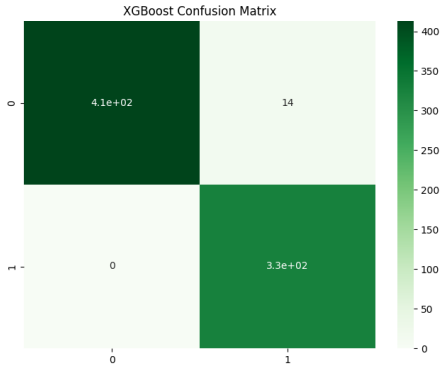

Figure 49: XGBoost Confusion Matrix

References

[Boh20] Jakesh Bohaju. Brain tumor, 2020.

[Ram25] A; Pawar MK Ramtekkar, PK; Pandey. Accurate detection of brain tumor using optimized feature selection based on deep learning techniques. *Multimed Tools Appl.*, 2023 Apr 25.