Using machine learning and deep learning approaches to detect Brain Tumors in MRI images

Authors: Juan Guerrero, Jiawei Liu, Niloufar Nouri

Abstract

Cancer is the second leading cause of death after cardiovascular disease. Of all types of cancer, brain cancer has the lowest survival rate. Brain tumors can come in different types, depending on their shape, texture and location. A proper diagnosis of the type of tumor allows doctors to make the right treatment choices to help save patients' lives. However, a recent study indicated that 12% of brain tumors had been misdiagnosed by pathologists, which means that the diagnostic accuracy of brain tumor detection is around 88%. In this paper, we propose an advanced methodology to determine the existence of brain tumors using machine learning and deep learning. Finally, we evaluated the proposed model on a Kaggle dataset containing 3064 MRI images as well as masks. We achieved the highest accuracy of 97%, exceeding the diagnostic accuracy of pathologists.

1. Introduction

Cancer is a disease that is becoming more prominent in the lives of individuals across the globe. Recent breakthroughs in science and technology have led to the production of machines that are able to scan any part of the human body. However, more often than not, a medical professional is needed in order to assess the size, position, and severity of these malignant cells. If not read by a medical professional, the consequences can be fatal for a patient carrying the disease. According to the American Brain Tumor Association, “There are over 120 types of brain cancer observed, according to the location of occurrence.”[1]. There is no question that it is imminent that a tumor must get assessed accurately and in due time. In this project we would like to tackle that problem by leveraging methods used in Data Science such as the creation of machine learning models that can learn and make predictions accurately.

In this exercise we will be using a Kaggle Data set containing 3929 Magnetic Resonance Images (MRI). Our approach will be to first perform some techniques used in Data Exploratory Analysis. Second, we will need to balance the binary classes, followed by the implementation of PCA to reduce the complexity of the images. Lastly, we will be training models and testing their performances using top of the line python libraries such as Scikitlearn and Keras. Ultimately we will showcase our top performing models: SVM and CNN.

2. Background

In order to understand the images used in this project, one must comprehend the background behind the generation of these images. According to *Brain Tumor Detection based on Machine Learning Algorithms,* An MRI scan "is a technique which depends on the measurement of magnetic field vectors that are generated after an appropriate excitation of string magnetic fields and radio frequency pulses in the nuclei of hydrogen atoms present in the water molecules of a patient's body."[2] Based on this idea, an image can be generated which can potentially show a particular mass and its properties that could identify it as either benign or malignant.

Other works have exposed their excitement for creating the finest model possible that makes accurate classifications in a timely manner. Natarajan et al .[2] proposed a brain tumor detection method for MRI images. The MRI brain images are first processed using a median filter, then segmentation and morphological operations are applied and then finally, the tumor region is obtained using image subtraction technique. This approach gives the exact shape of the tumor in an MRI image. Another work produced by Joshi et al. [3] proposed to create a system to detect and classify tumors in MRI images. The approach is as follows, first the tumor portion is extracted, then the texture features of the detected tumor using Gray Level Co-occurrence Matrix (GLCM) is produced and then classified using neuro-fuzzy classifier. From the examples proposed above, there is an overall trend which indicates that most brain tumor detection systems use texture, symmetry and intensity as features.

On the other hand, there have been numerous studies reporting the use of non-traditional machine learning methods such as those found in deep learning. In a report conducted by Othman and Ariffanan[5], "they propose a new system for brain tumor automatic diagnosis. The probabilistic Neural Network (PNN) provides a solution to the pattern classification problem." Through this approach, the scientists go through the following preprocessing steps, first the MRI images are converted to matrices by using MATLAB and then a PNN classification algorithm is

used to classify the images. The system yields an accuracy reading of more than 73%, and this accuracy can vary depending on the “smoothing factor” imposed. The features that they were able to extract from this experiment included a mean gray value which highlighted the area of interest, a standard deviation value of the gray area, the calculation of the area function, and the aspect ratio of the image. These features were among the most important that played a big role in the performance of the model. After finishing the experiment, the scientist realized that even though classification techniques are becoming more common nowadays, for this particular use case, there is no doubt that neural network classifications perform exceptionally well.

As seen in the previous examples from other related work, there is no question that accurate detection is imminent when detecting brain abnormalities. In this next study, which closely aligns with our experiment, they present a development of a new approach for automated diagnosis. “Based on the classification of Magnetic Resonance(MR) human brain images. 2D wavelet transform and spatial gray level dependence matrix is used for feature extraction.”[6] They also use a stratified K-fold cross validation to overcome the issue of overfitting and lastly, the use of hyperparameters is implemented to increase the overall power model. At the end this study was able to achieve a classification accuracy rate of exactly 95.6522% [6] via the hyper-parameter tuned Support Vector Machine.

3. Data

3.1. Data description and source

In this project we used brain Magnetic Resonance Images (MRI) images which contain 3929 images along with manual fluid-attenuated inversion recovery (FLAIR) abnormality segmentation masks. We collected images from the Kaggle dataset. The images were originally collected from The Cancer Imaging Archive and corresponded to patients included in The Cancer Genome Atlas (TCGA) lower-grade glioma collection. All images are provided in ‘.tif’ format with 3 channels per image. All images have 3 sequences, that is pre-contrast, FLAIR and post-contrast. For a few cases where post-contrast sequence or pre-contrast sequence is missing, the missing sequence is replaced with FLAIR sequence to make all images 3-channel. Masks are 1-channel images.

3.2. Exploratory data analysis and preprocessing

In the first step, we used mask images to generate labels for each image instance. That means, each image was assigned with a tumor or no-tumor label. We presented a few instances of images along with their masks in Figure 1. In this figure each row corresponds to one observation which includes the 3-chanelle image, the gray-scale converted image and its corresponding mask.

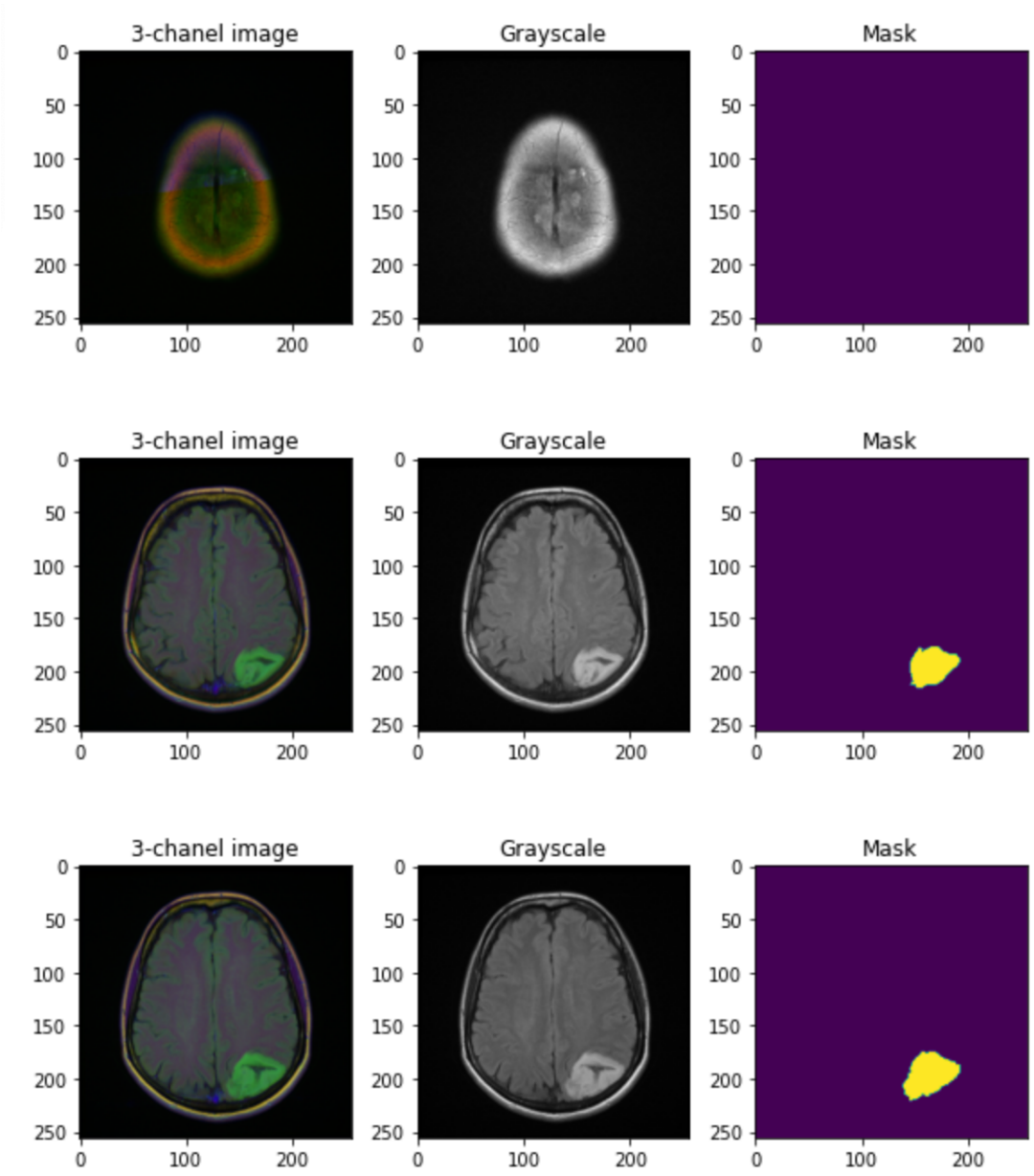

Figure 1. 3-chanel, gray-scale and mask images

The initial exploratory analysis revealed that classes (tumor/no-tumor) are imbalanced. We found that 2556 images do not have any tumor, and 1373 images have a tumor as presented in Figure 2. We addressed this issue by oversampling techniques which will be discussed in more detail in the method section.

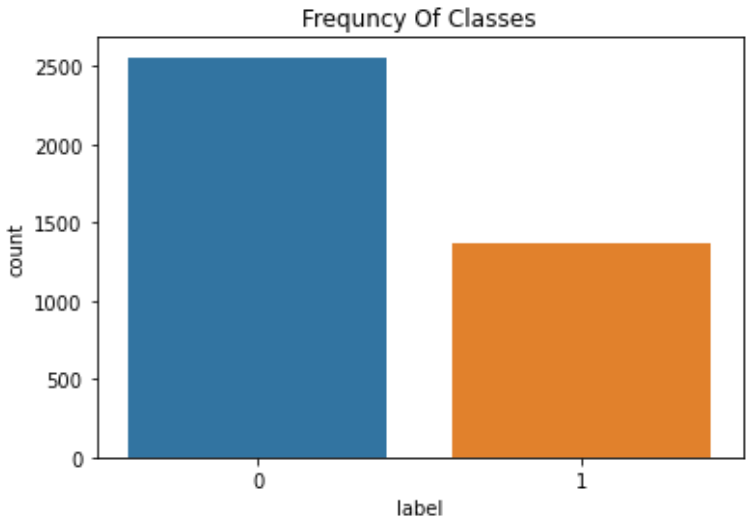

Figure 2. The distribution of classes in image dataset

Due to the memory insufficiency, we used gray-scale images for machine learning part of the project. However we used all three channels in deep learning models. As shown in Figure 3 we verified that all gray-scale images have 256 by 256 pixels.

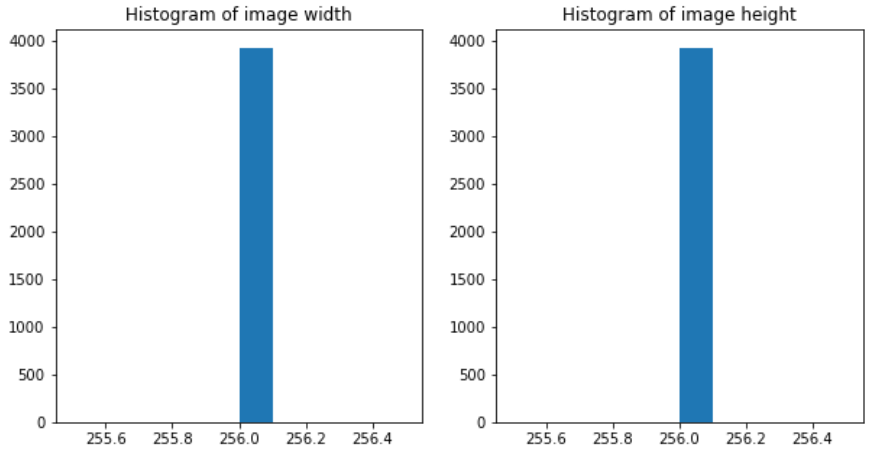

Figure 3. The distribution of pixel values in image dataset

4. Methods

4.1. Data Augmentation (oversampling)

As discussed earlier, the image dataset suffers from class imbalance. The class imbalance problem is a common issue which can affect machine learning (classification) performance due to having a disproportionate number of class instances. To resolve this issue, we generate new images of class 1 (with tumor) by transforming the existing tumor-positive images and their corresponding masks. Transform operations include Flip, rotate, changing the contrast and brightness. To perform these transformations, we used Albumentations python library which is a computer vision package useful for image processing, segmentation and classification. The package contains powerful methods to implement a wide variety of image transformation. Figure 4 shows an example transformed image:

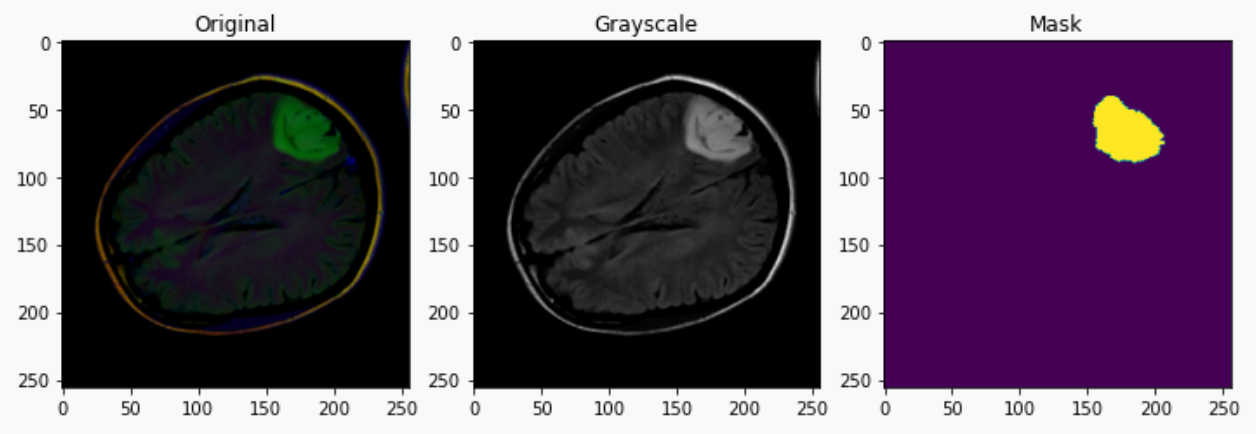

Figure 4- The transformed image and mask

After balancing the image data, we achieved a total of 5302 images in which the frequency of class 0 and 1 was approximately equal. (2746 images in class 0, and 2700 images in class 1). The following barchart represents the frequency of each class after data augmentation.

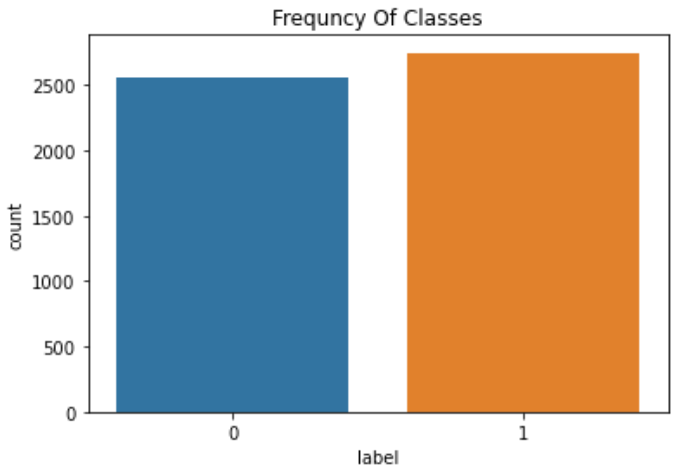

Figure 5- The frequency of classes after data augmentation

4.2. Feature selection

As discussed in the data section the number of features to be used for classification algorithm was 65,536. We tried running multiple classification algorithms using the entire features, but it took so long. In order to optimize the computational performance, we applied two feature selection techniques including Principal Component Analysis and Random Forest feature importance. For PCA analysis, we determined the number of components needed to explain 90% of variance. Results showed that using 470 components we can achieve 90% of variality. Figure 6 shows the variation of explained variance with number of components. We also presented the PCA-reconstructed version (using 470 components) for one of the images along with the actual image in Figure 7.

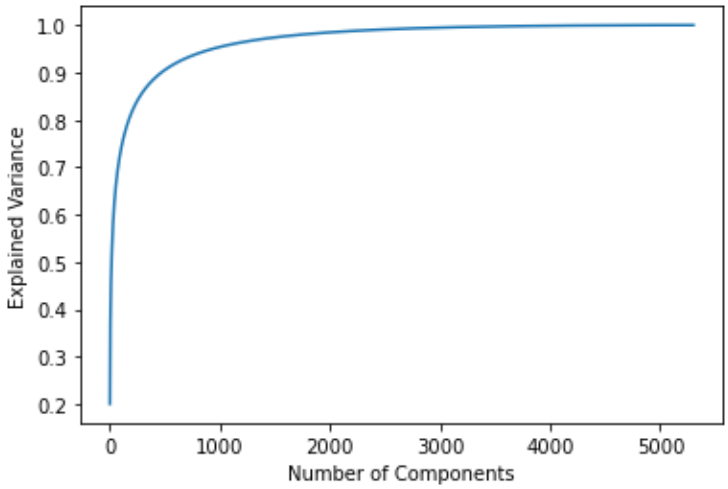

Figure 6. The variation of explained variance with number of components

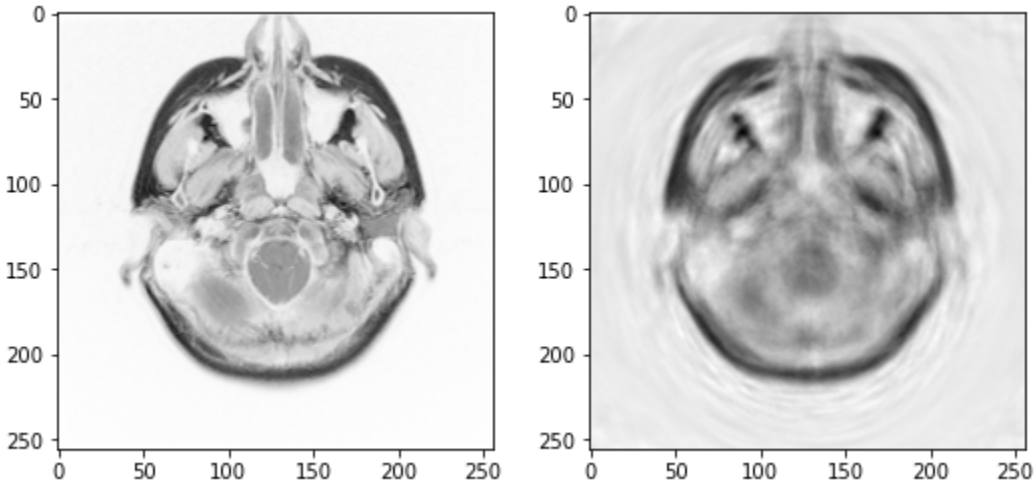

Figure 7. An example of actual image along with the pca-reconstructed image using 470 components (i.e. 90% explained variance)

We also used Random forest feature importance to find the top features which could achieve high accuracy. We found that 1000 features from Random Forest could achieve the same accuracy as PCA features would do. The details of classification results will be further explained in the next two sections.

4.3. Machine learning: Classification

To classify brain images, we started with using full-features data to train the model. However, using more than 65000 features resulted in memory crash and slow computation time. We found that reducing features with PCA or Random Forest feature importance would achieve an accuracy score close to the full-feature model. Therefore, we used PCA features to train classification algorithms. We splitted the whole dataset into three sets (Train, Validation and Hold-out set). The validation set was used to evaluate classification performances in different algorithms and choose the two best ones. At this step, we tried naive bayes classifier, logistic regression, K-nearest neighbors, linear support vector machine, polynomial and RBF support vector machine, gradient boosting, decision tree, random forest, quadratic discriminant analysis and stochastic gradient descent classifier.

After training and validating all models, we picked two best ones, and used the Hold-out set to obtain evaluation metrics including accuracy score, classification report (Precision, Recall and F1-score) and AUC score. The hold-out set was never seen before in the validation phase where we implement multiple classification techniques. The reason for this step is to check if the two best models still achieve high classification accuracy using the unseen data. After choosing the best two models, we also ran a 10-fold cross-validation to check their performance in a cross-validation framework.

Additionally we were interested to see if ensemble methods would achieve better accuracy, so we ran a voting classifier on three top models. The results of evaluation metrics will be presented in section 5, i.e. evaluation.

4.4. Deep learning: Classification

Furthermore, we would like to explore an alternative solution using deep learning as opposed to the classical machine learning approach. To begin with, since we were dealing with an image dataset, we designed and implemented a convolutional neural network (CNN) [7] to investigate its performance. To reduce complexity and increase

generality, we decided to develop a six-layer CNN model that includes both input and output layers. As we all know, there is no universal answer to determine the number of filters for the convolutional and dense layers. After some experiments, we decided to develop a CNN model with the following architecture. To avoid the overfitting problem, we also added dropout layers as well as batch normalization layers to the structure of our CNN model.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 32)      896

 activation (Activation)     (None, 254, 254, 32)      0

 batch_normalization (BatchN  (None, 254, 254, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 127, 127, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 127, 127, 32)      9248

 activation_1 (Activation)   (None, 127, 127, 32)      0

 batch_normalization_1 (Batc  (None, 127, 127, 32)     128
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 63, 63, 64)        18496

 activation_2 (Activation)   (None, 63, 63, 64)        0

 batch_normalization_2 (Batc  (None, 63, 63, 64)       256
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 64)       0
 2D)

 flatten (Flatten)           (None, 61504)             0

 dense (Dense)               (None, 256)               15745280

 activation_3 (Activation)   (None, 256)               0

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 128)               32896

 activation_4 (Activation)   (None, 128)               0

 dense_2 (Dense)             (None, 1)                 129

 activation_5 (Activation)   (None, 1)                 0

=================================================================
Total params: 15,807,457
Trainable params: 15,807,201
Non-trainable params: 256
_________________________________________________________________
```

Figure 8. The structure table of the CNN model

With this structure, we developed and trained three different models with three different datasets. First, to further understand whether the compressed images influenced the performance of the machine learning models, we trained a CNN model with images compressed by PCA for comparison. Second, we trained a CNN model with grayscale images. Lastly, we trained a CNN model with multi-channel images. For the model trained by multi-channel image, we got an impressive metric score in terms of accuracy, precision and recall, compared to our machine learning models and the other two CNN models. However, we are still concerned about whether our CNN model performed well on feature extraction. Therefore, we decided to use the transfer learning approach where we can construct a new model with the pre-train VGG16 model [8]. For the VGG16 model, we trained it with ImageNet [7] and added 3 dense layers as top layers to it. We anticipated this transfer learning model was outperforming our other CNN models. By integrating the pre-train model, the accuracy of our new model improved from 92% to 95%. In addition, both precision and recall have been improved. However, in this model, the recall value was lower than the precision value. Since we were working on a medical problem, we were supposed to concentrate more on recall rather than precision. Therefore, we determined to train an image separation model as our supplementary model to better understand MRI images. As we all know, the image segmentation model is comparing each pixel between the image and the ground truth mask, which means it is more sensitive to tumor detection. With respect to the structure of our separation model, we chose U-Net [9] because of its excellent performance on biomedical images, especially MRI images. In this project, we added two dropout layers to the structure of U-Net to avoid overfitting.

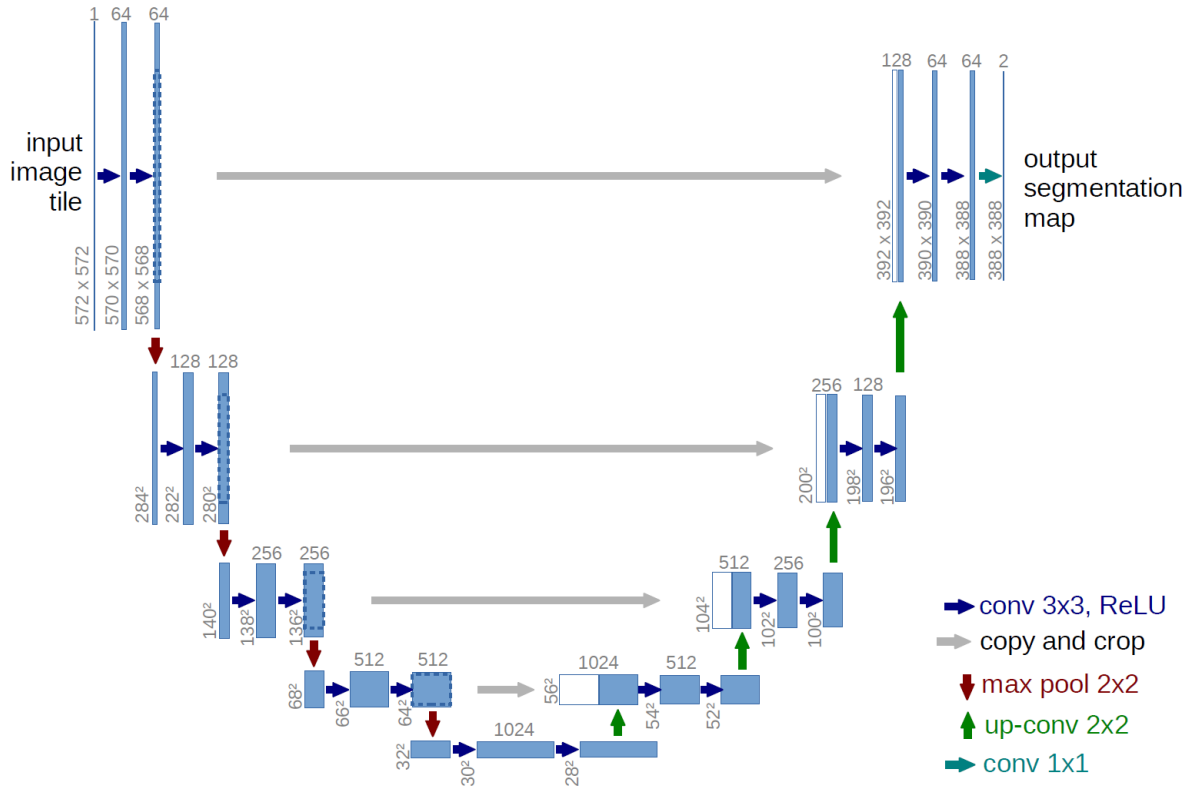

Figure 9. The structure of U-Net

In conclusion, to combine the advantages of each model, we decided to create a voting classifier that combines three neural network models including the CNN model trained with multichannel MRI images, the transfer learning model with a pre-trained VGG16 model, and a U-Net separation model. We were looking for the best metric score on this voting classifier.

5. Evaluation

5.1 Evaluation of ML part

For this evaluation, let us look at all of the models used which were trained on the MRI images. We will begin with the precision which indicates the quality of a positive prediction made by a model.

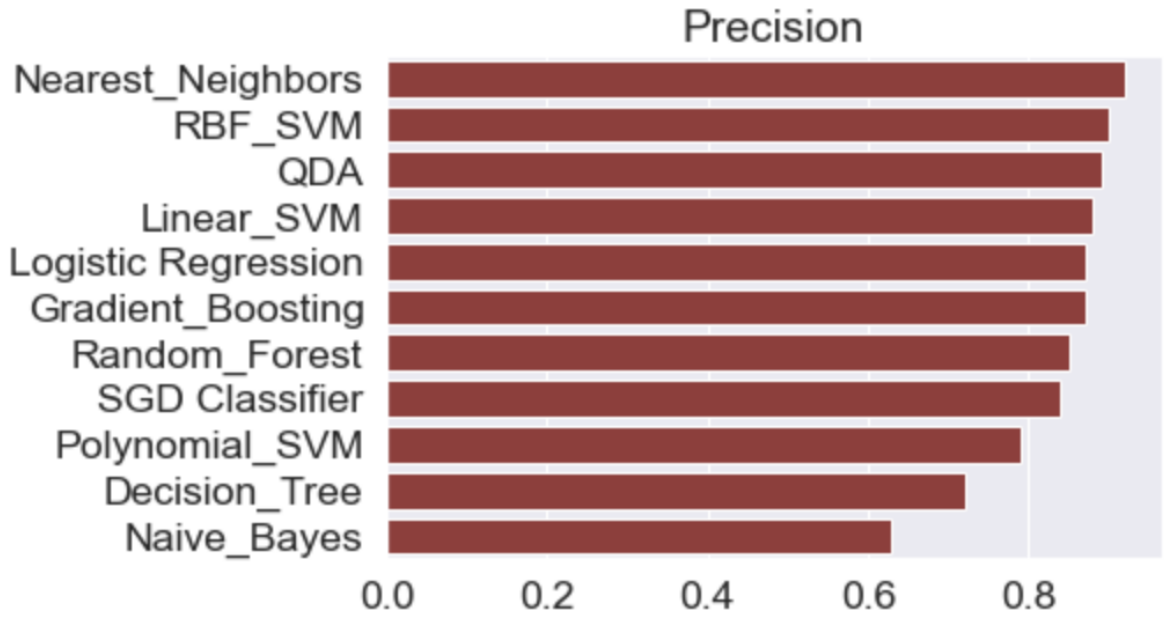

Figure 10. Precision of All Models

As seen above we can conclude that although most models performed relatively well, the KNN classifier surpassed the precision metric across all instances of the classifiers.

Since KNN accuracy is very close to the SVM (with RBF kernel) and it's higher than linear SVM, it indicates that data is not easily separable using the linear decision planes.

Next, let us look at the accuracy which describes how well the model performs across all classes. This is mostly important when the classes are of equal importance.

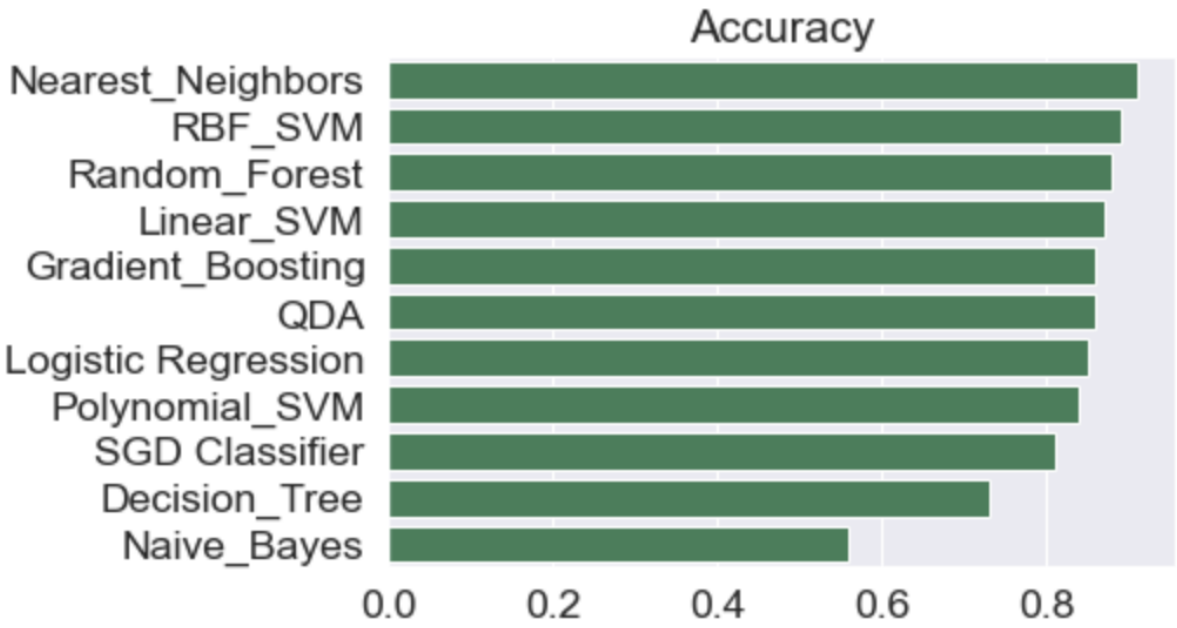

Figure 11. Accuracy score of All Models

As shown in the figure the KNN model once again performs exceptionally well when compared to the other classifiers. However, the RBF Support Vector Machine and Random Forest classifiers also yielded exceptional values which are more important in this experiment. A high accuracy value is most relevant when the classes are balanced. Next we will look at the recall.

Figure 12. Recall of Models

Through the recall, one can analyze how many of the true positives were recalled, in other words, how many of the correct hits were found. This time around, the Polynomial_SVM performs the best followed by Random Forest classifier. Lastly, we can analyze the AUC ROC score.

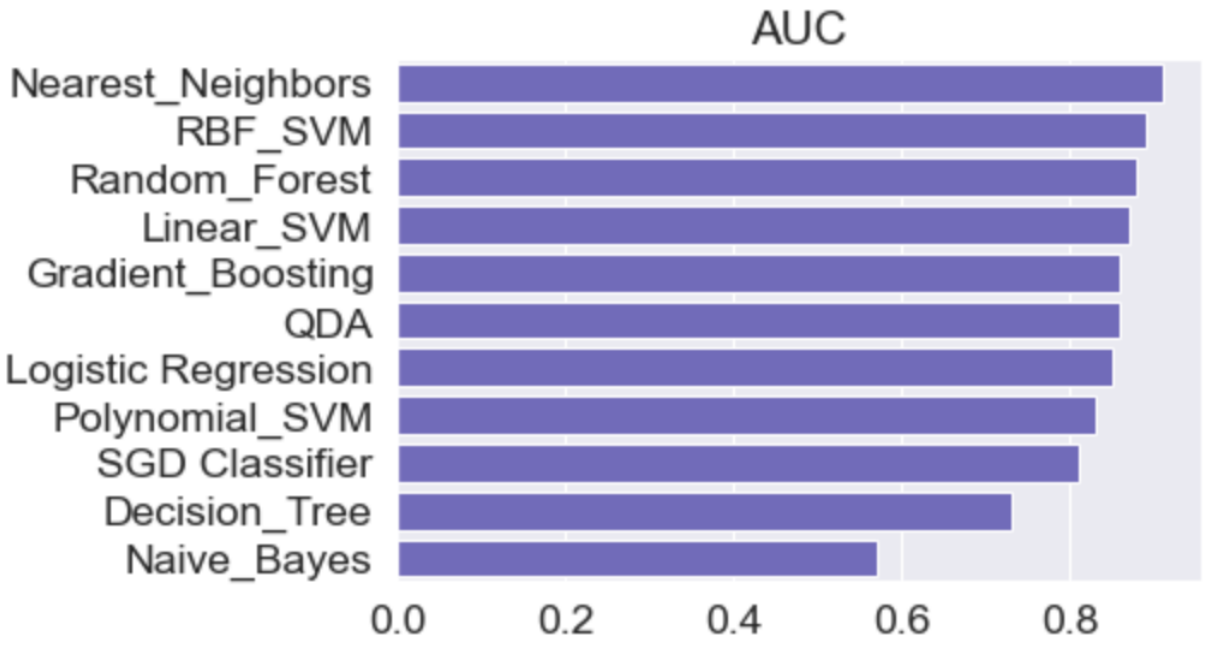

Figure 13. AUC of Models

This graph is able to show us the ability of a classifier to distinguish between classes. The higher the score the better the performance of a model to pick between positive and negative categories. From these results we decided to pick the two best performing models, KNN and SVM RBF, and generate an ROC AUC curve as shown below.

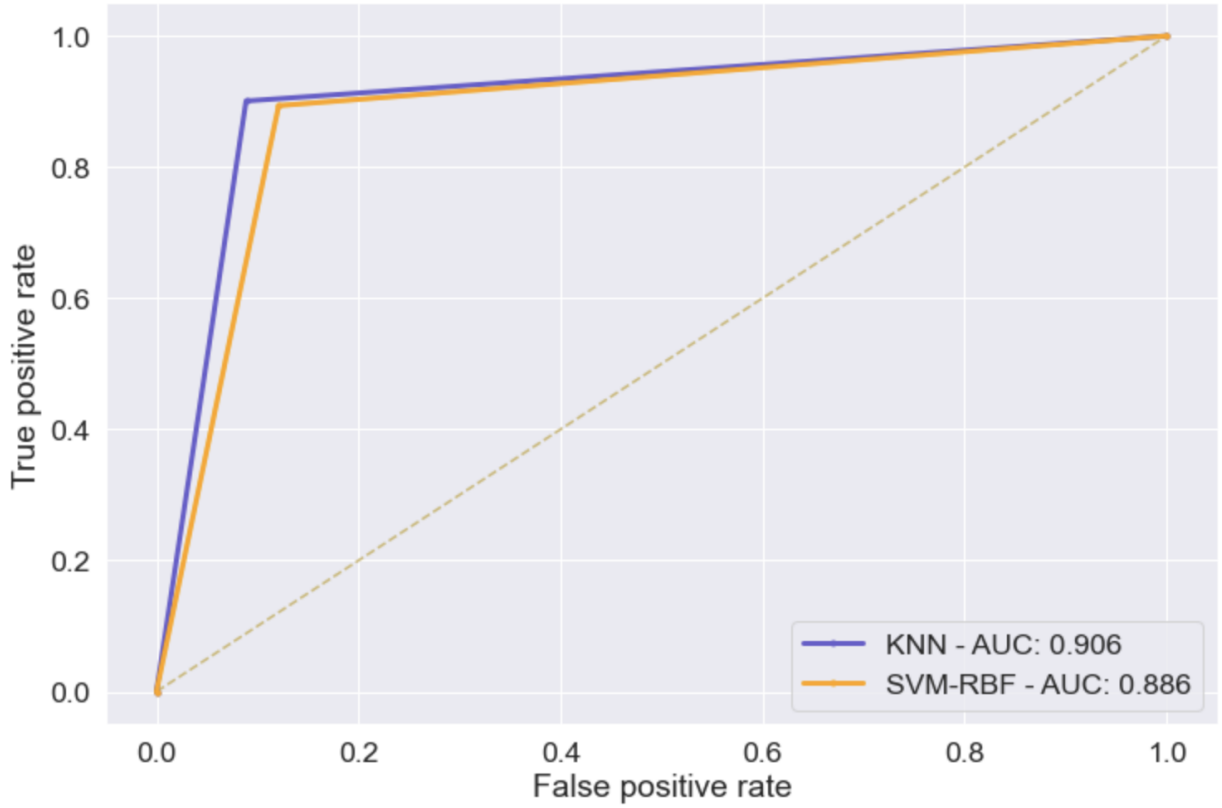

Figure 14. ROC of KNN and SVM-RBF

To further this investigation we conducted a 10-fold cross validation and compare the results with the values seen previously.

```
ROC AUC: 0.95 +/- 0.01 [Support Vector Machine (RBF)]
ROC AUC: 0.95 +/- 0.01 [Random Forest Classifier]
ROC AUC: 0.95 +/- 0.01 [KNN]
```

Figure. 15-fold-cross validation metrics

As seen above, we can conclude that using a 10 fold cross validation, which further divides the data into training and validation sets and samples without replacement, yields even better metrics than the approach used previously. Lastly, we picked the best performing models and created a voting classifier, which in theory combines the models and uses a hard voting approach to ultimately return the best prediction. The results are as follows

Classifier	Accuracy	Precision	Recall	AUC
Voting Classifier	0.92	0.93	0.92	0.94

These results support the theory that combining the classifiers does increase the performance of accurate predictions. Thus with these results, we can conclude that we can surpass our initial benchmark of 88% accuracy that was initially stated in this experiment.

5.2 Deep learning: Evaluation metric

When training the CNN models, we not only added dropout layers as well as batch normalization layers, but also applied an early stopping technique to stop the training process while the validation loss increased. Hence, our CNN models were able to be trained with enough epochs. We will take the loss-epochs plot of the CNN model trained by multi-channel MRI images as an example. As shown in the figure below, the CNN model is close to convergence after 15 calendar hours and the validation accuracy line looks good without sharply jumping.

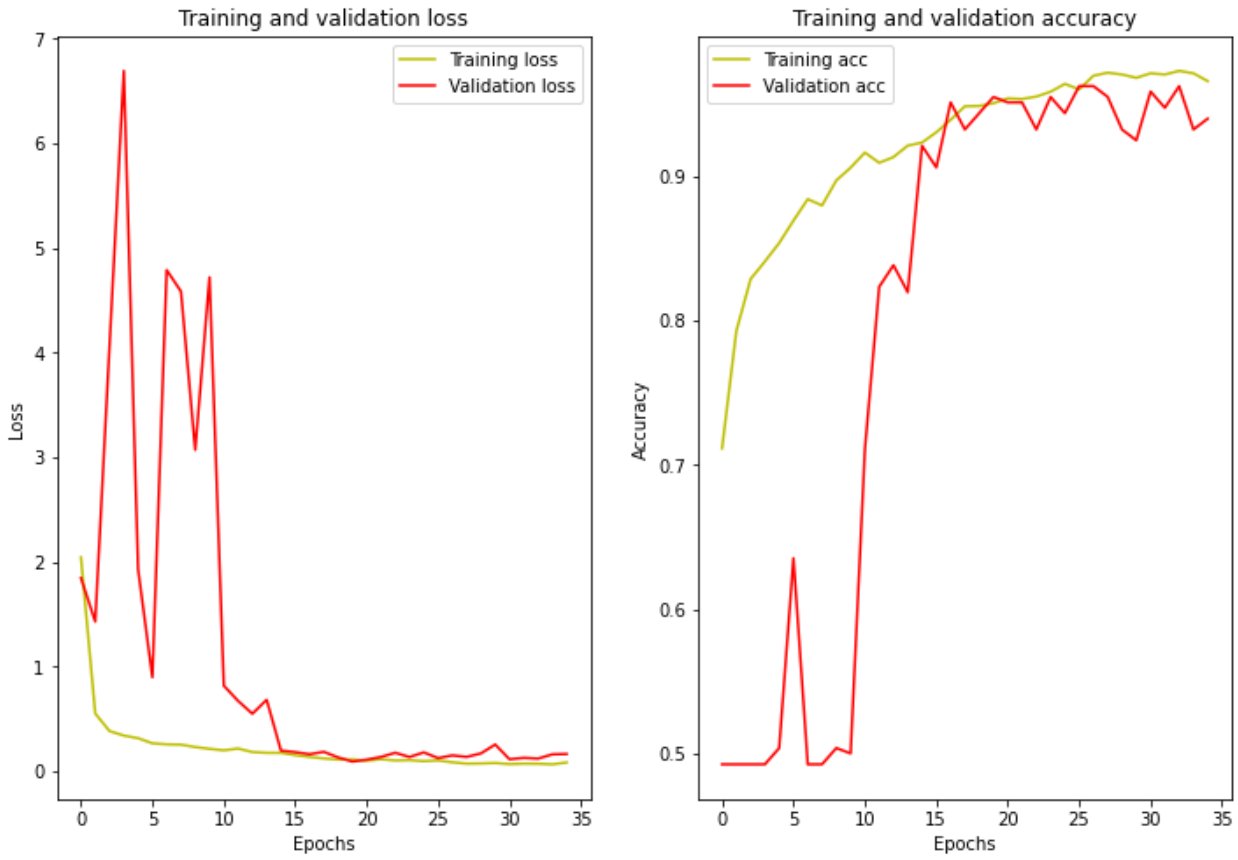

Figure 16. Loss-Epochs plot

Regarding the evaluation of neural network classification models, we will study the performance of our three CNN models and the transfer learning model with pre-trained VGG16 model. In this study, we mainly focus on four metric scores, namely accuracy, precision, recall and AUC. From the table below, we can see that the worst model is trained from grayscale images that have been compressed by PCA. An interesting finding is that our CNN model structure does not benefit from multi-channel MRI images, which is also a good indication that this structure does not do well in terms of feature extraction. Finally, as we expected, the transfer learning model with the pre-trained VGG16 model outperformed the other three CNN models. However, the recall value is relatively low compared to the accuracy, which is why we need to train a segmentation model as a supplementary model.

Classifier	Accuracy	Precision	Recall	AUC
CNN-Gray-PCA	0.9094	0.9071	0.9203	0.9292
CNN-Gray	0.9211	0.9412	0.9078	0.9943
CNN-Multi-Channels	0.9245	0.9275	0.9275	0.9838
VGG16	0.9585	0.9847	0.9348	0.9896

Figure 17. Performance metrics of neural network classifiers

As for the evaluation of the segmentation model, we will study its performance using the IoU score and the Dice coefficient. In this project, we get 79% of the IoU score and 88% of the Dice coefficient score. Furthermore, we customized the segmentation model as a binary classifier and compared its performance with a CNN model trained with multiple channels and a transfer learning model using a confusion matrix. In addition, we created a voting classifier that wraps all three models together and expects better performance. The four confusion matrices are listed below. As we expected, the voting classifier has the best performance. In addition, the recall has improved from 93% to 96% due to the combination of the segmentation classifier.

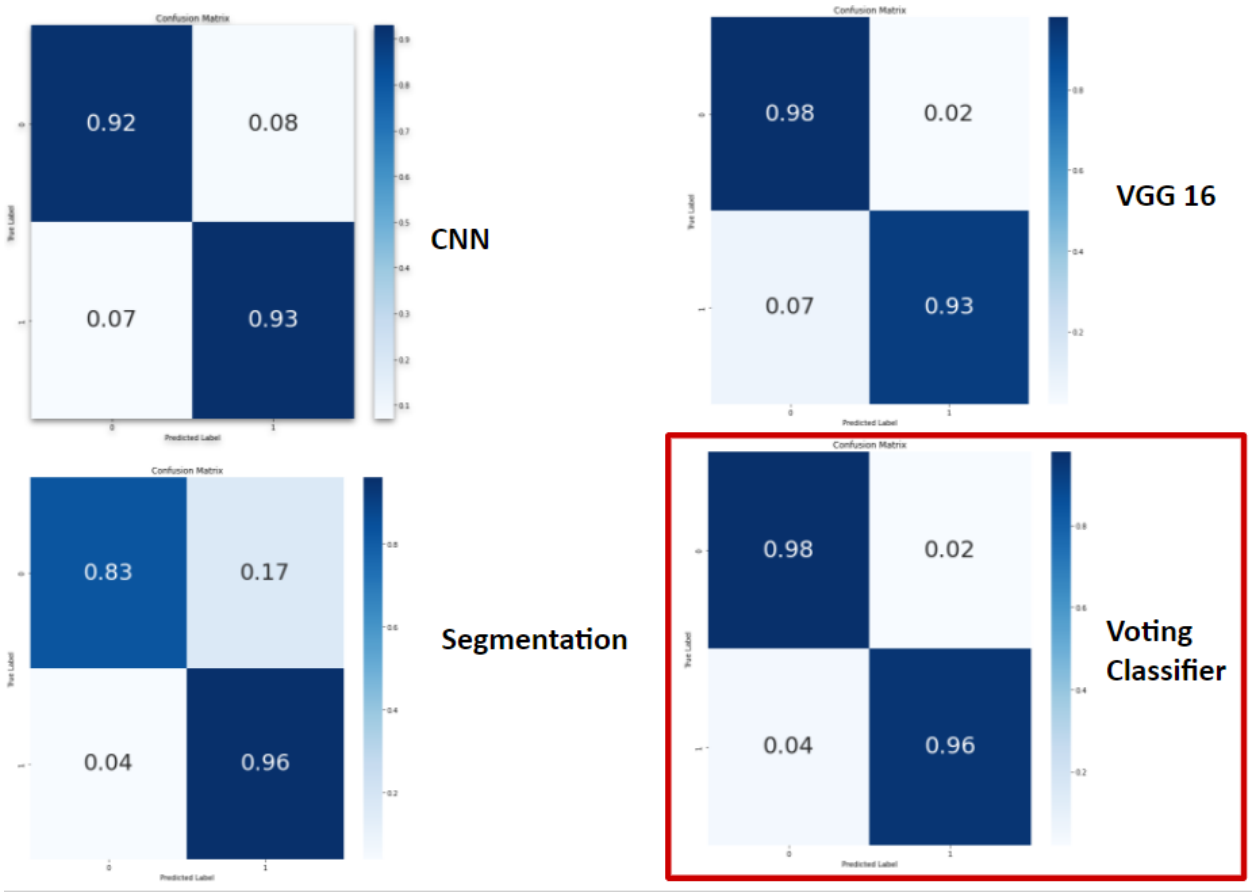

Figure 18. Confusion Matrices for Deep Learning

Finally, we come up with a stronger voting classifier with a 97% of accuracy score, 98% of precision score, and 96% of recall score

6. Conclusion

It is evident that the accurate detection of a brain tumor plays an important role in the life expectancy of a person who develops this disease. With the help of machine learning models, we are able to accurately identify and detect the area where this mass is present. Following a supervised learning approach in both classical and deep learning methods, we were able to build an exceptional model that runs and predicts accurately and faster than your average healthcare practitioner. The highest accuracy achieved using classical machine learning models was 0.92 (AUC=0.94). The metrics for the best performed model among deep learning/segmentation methods is shown below:

Classifier	Accuracy	Precision	Recall	F1-Score
Voting Classifier	0.97	0.98	0.96	0.97

According to a recent study, it indicated that 12% of brain tumors had been misdiagnosed by pathologists. In addition, even NYU has similar error rates of 12-14% among its patients [10]. Across the board, these values beat our initial benchmark that we used to gauge the models. In conclusion, the metrics for our optimal model indicate the high potential and usability of the proposed solution. Moreover, this experiment proves the importance of machine learning in the healthcare field and demonstrates that these data science practices can be utilized for the greater good of humanity.

7. Contribution

In terms of contribution, Juan and Niloufar mostly worked on EDA, feature selection and Machine Learning algorithms; Jiawei mostly worked on data augmentation, deep learning and segmentation. However, we met every other day to brainstorm about next steps and to share our thoughts about the results as we were moving forward. For the report, Niloufar and Juan worked on Background, EDA, Methods and Machine Learning part; Jiawei worked on Abstract, Deep learning, Segmentation and Conclusion.

8. Bibliography

1. V. Panca and Z. Rustam , "Application of machine learning on brain cancer multiclass classification", AIP Conference Proceedings 1862, 030133 (2017) https://doi.org/10.1063/1.4991237

2. Sharma, Komal, Akwinder Kaur, and Shruti Gujral. "Brain tumor detection based on machine learning algorithms." International Journal of Computer Applications 103.1 (2014): 7-11.

3. Al-Ayyoub, Mahmoud, et al. "Machine learning approach for brain tumor detection." Proceedings of the 3rd international conference on information and communication systems. 2012.

4. Dipali M. Joshi, N. K. Rana, V. M. Misra, " Classification of Brain Cancer Using Artificial Neural Network" , IEEE International Conference on Electronic Computer Technology ,ICECT ,2010.

5. D. F. Specht. Probabilistic neural networks. Neural Networks, 3(1):109–118, 1990.

6. A. Kharrat, M. B. Halima and M. Ben Ayed, "MRI brain tumor classification using Support Vector Machines and meta-heuristic method," 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA), 2015, pp. 446-451, doi: 10.1109/ISDA.2015.7489271.

7. ImageNet Classification with Deep Convolutional ... - List of Proceedings. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

8. Simonyan, Karen, and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” ArXiv.org, 10 Apr. 2015, https://arxiv.org/abs/1409.1556.

9. Ronneberger, Olaf, et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” ArXiv.org, 18 May 2015, https://arxiv.org/abs/1505.04597.

10. Savage, Neil. “How Ai Is Improving Cancer Diagnostics.” Nature News, Nature Publishing Group, 25 Mar. 2020, https://www.nature.com/articles/d41586-020-00847-2#:~:text=In%20the%20initial%20study%2C%20the,%E2%80%9314%25%20among%20its%20patients.