Brain Tumor Classification
With Autoencoder

Atahan Çaldır[*], **Musa Berkay Kocabaşoğlu**[*], **and Kerem Ersan**[*]
Department of Computer Science, Özyeğin University, Istanbul, Turkey
Email: {atahan.caldir, musa.kocabasoglu, kerem.ersan}@ozu.edu.tr

***Abstract*—The report contains the final project details of the CS 454 (Introduction to Machine Learning) course supervised by Prof. Ethem Alpaydın from Ozyegin University. The final project objective is to classify brain tumors given the various brain MRI images using VGG16 Convolutional Neural Network (CNN). To extend the problem with the knowledge that is learned in the course, an autoencoder logic is also implemented into the system to decrease the training time of the model.**

***Index Terms*—machine learning, deep learning, classification, brain tumors, autoencoder**

I. INTRODUCTION

A brain tumor is a collection, or mass, of abnormal cells in the brain. The skull, which encloses the brain, is very rigid and any growth inside such a restricted space can cause problems. Brain tumors can be cancerous (malignant) or noncancerous (benign). When benign or malignant tumors grow, they can cause the pressure inside the skull to increase. This can cause brain damage, and it can be life-threatening.

A Brain tumor is considered one of the most aggressive diseases, among children and adults. Brain tumors account for 85% to 90% of all primary *Central Nervous System(CNS)* tumors. Every year, around 11,700 people are diagnosed with brain tumors. The 5-year survival rate for people with a cancerous brain or CNS tumor is approximately 34% for men and 36% for women.

Brain Tumors are classified as; *Benign Tumors, Malignant Tumors, Pituitary Tumors*, etc. Proper treatment, planning, and accurate diagnostics should be implemented to improve the life expectancy of the patients. The best technique to detect brain tumors is Magnetic Resonance Imaging (MRI). A huge amount of image data is generated through the scans. These images are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties. Application of automated classification techniques using Machine Learning(ML) and Artificial Intelligence(AI)has consistently shown higher accuracy than manual classification. Hence, proposing a system performing detection and classification by using Deep Learning Algorithms using Convolution Neural Networks (CNN), Artificial Neural Networks (ANN), and transfer learning (TL) would be helpful to doctors all around the world.

Early detection and classification of brain tumors is an important research domain in the field of medical imaging and accordingly helps in selecting the most convenient treatment method to save patients' life therefore

In this study, a VGG16 Convolutional Neural Network (CNN) model is implemented to classify the brain tumors given the MRI images of different brains with and without tumors. To accelerate the training process by learning the representations of the compressed images, an autoencoder is integrated into the system, and its performance is measured by comparing the accuracy and training time of the models that use normal images, and images compressed with the autoencoder. The aim of this implementation is to minimize both the loss and the training time.

Section II explains the utilized data, details of the VGG16, the autoencoder structure, and the packages that are used in the software. The problem statement and the proposed solution can be found in sections III and V. For the literature analysis about the medical usage of machine learning, section VI should be read. Finally, the conclusion of the overall project is in section VII.

II. BACKGROUND

A. *Data*

In the project, *Brain Tumor MRI Dataset*[1] is fetched from Kaggle for the implementation. The data contains 7022 images of human brain MRI images which are classified into 4 classes and are split into train and test parts by default.

The labels of the classes are as follows;

- *Glioma:* A Glioma is a type of tumor that starts in the glial cells of the brain or the spine. Gliomas comprise about 30 percent of all brain tumors and central nervous system tumors, and 80 percent of all malignant brain tumors [1].
- *Meningioma:* A Meningioma is a primary central nervous system (CNS) tumor. This means it begins in the brain or spinal cord. Overall, meningiomas are the most common type of primary brain tumor [2].
- *Pituitary:* Pituitary tumors are unusual growths that develop in the pituitary gland. This gland is an organ about the size of a pea. It's located behind the nose at the base of the brain. Some of these tumors cause the pituitary gland to make too much of certain hormones that control important body functions [3].
- *No Tumor:* This label represents there is **no tumor!**

---

[*] Equal contribution.

[1] Brain Tumor MRI Dataset, Kaggle, 2020

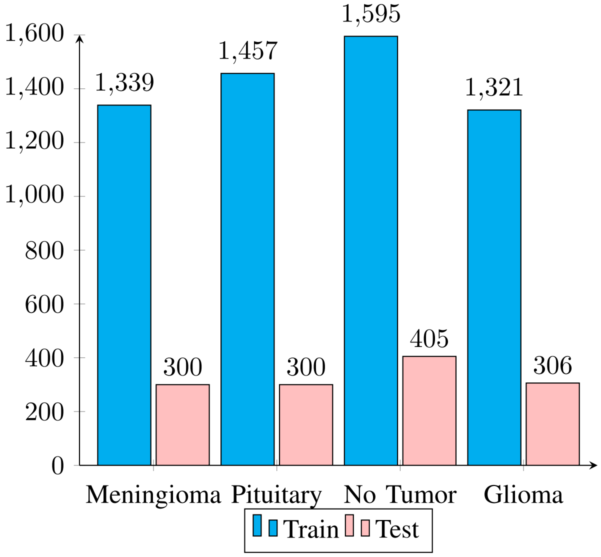


Fig. 1: Data distributions by counts

Fig. 1 shows the distribution of the data over the train and test sets. The distribution of both the train and test data is close to uniform distribution. The balanced structure of the data prevents the model from overfitting and provides close chances to each class label to be predicted.

In terms of the train/test split of the data, the distribution of the class labels are close to uniform distribution just like in the general overview of the data (Appendix 1). The train/test split ratio is 80:20 as shown in the Fig. 2.

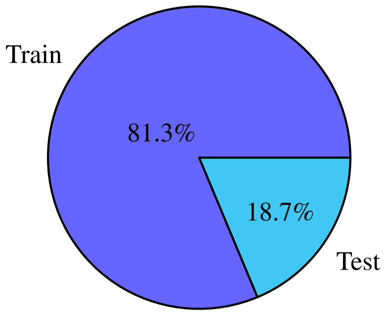


Fig. 2: Train/Test Split Percentage of the Data

The MRI images have different perspectives of the brain that are the top view, left/right side views, back view, and front view of the head. An example of the MRI images and corresponding tumor types are given in Fig. 3. The performance of the model with The realistic and diverse side of this data

B. Algorithms

*1) **VGG16**:* VGG16 is a convolutional neural network model trained on the ImageNet dataset. The ”VGG” in VGG16 stands for ”Visual Geometry Group”, which is the research group at Oxford University where the model was developed. The ”16” refers to the number of layers in the model.

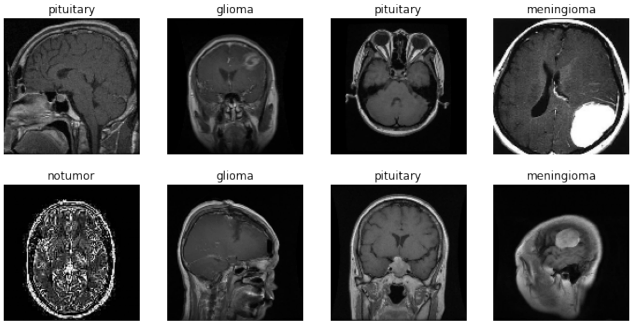


Fig. 3: Brain MRI images

The model was introduced in the 2014 paper [4]. It was trained on the ImageNet dataset, which is a large dataset of images labeled with 1000 different object categories.

VGG16 is a feedforward network that consists of 16 convolutional layers and 3 fully-connected layers. It was one of the first very deep convolutional networks, achieving record-breaking results on the ImageNet classification task. The model architecture is simple and straightforward, and it has been widely used as a benchmark for image classification tasks.

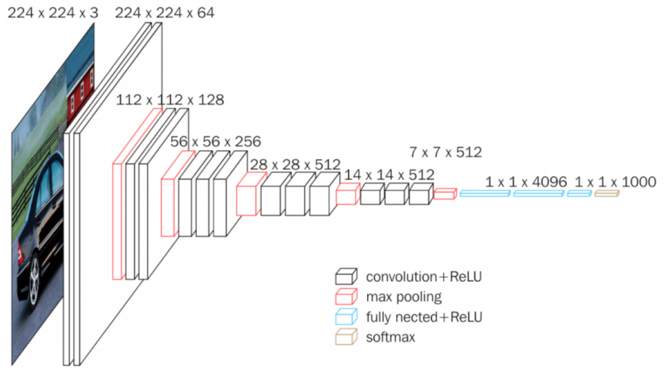


Fig. 4: VGG16 Architecture

*2) **Autoencoder**:* An autoencoder is a component of a neural network that is responsible for reducing the input data into a lower-dimensional representation, or encoding. This encoding is typically used to compress the input data so that it can be more easily processed by the rest of the network.

The autoencoder architectures vary widely, but they typically involve a combination of linear and non-linear transformations of the input data. For example, an autoencoder might consist of a series of convolutional layers that extract features from the input data, followed by one or more fully-connected layers that reduce the dimensionality of the feature maps. Fig. 5 shows a general structure of an autoencoder.

One common application of autoencoders is in the field of natural language processing, where they are used to convert words or sentences into fixed-length vector representations. These vector representations can then be used as input to other neural networks models, such as language translation models or text classification models.

Other applications of autoencoders include image compression, anomaly detection, and generative modeling.

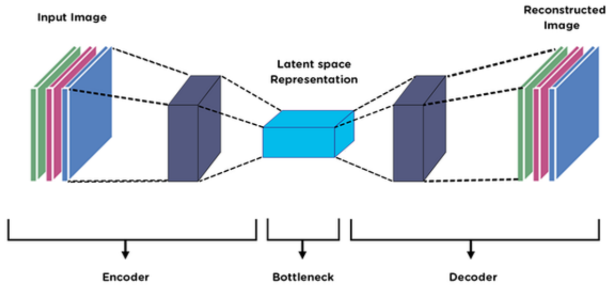


Fig. 5: Autoencoder Architecture

C. Packages

*1) **Scikit-learn**:* Scikit-learn is a powerful open-source library for machine learning and data science in Python. It offers a wide range of modeling and analyzing data, including linear and logistic regression, support vector machines, decision trees, random forests, and many more. Scikit-learn is built on top of NumPy and SciPy, and integrates well with other scientific libraries such as Pandas and Matplotlib. In our project, it is used to implement k-Fold Cross Validation, and to measure the precision, recall, accuracy, and f1 scores of the classifications.

*2) **Numpy**:* Numpy is a fundamental library for scientific computing in Python, and is widely used in the field of data science and machine learning. It is used to work with arrays and matrices, and is designed to make it easy to perform mathematical and statistical calculations on large and complex datasets. Numpy is built on top of low-level C and Fortran libraries, and is optimized for performance, making it much faster than standard Python lists and arrays for many types of calculations. In the project, Numpy is used to perform array and matrix calculations of the images, and to prepare data to appropriate format for the VGG16 model.

*3) **Matplotlib**:* Matplotlib is a Python library that is widely used for data science and machine learning for creating static, animated, and interactive plots and visualizations. It is built on top of the NumPy library and integrates well with other scientific libraries such as Pandas and Scikit-learn. In the project, Matplotlib is used for both data analysis and visualization of results. This library is particularly helpful when working with complex datasets, as it allows to create various types of plots, including line plots, scatter plots, histograms, box plots, and heatmaps.

*4) **TensorFlow**:* TensorFlow is an open-source library for multiple machine learning, deep learning and other statistical and predictive tasks. The framework includes sets of both high-level and low-level APIs and can work with either CPUs or GPUs (for higher performance).

*5) **Keras**:* Keras is a high-level deep learning API developed by Google that contains various built-in utilities to import and use with Python easily. It also allows users to design their own neural network structures with ready-to-use neural layers, cost functions, activation functions, etc. It works on top of the TensorFlow library and is capable of using TensorFlow's utilities. Keras is used to implement the VGG16 model and the autoencoder.

III. PROBLEM STATEMENT

Image classification is important for medical diseases because it can assist doctors and medical researchers in identifying and diagnosing a wide range of conditions. Medical images such as X-rays, CT scans, and MRI scans are often used to detect and diagnose diseases and injuries, but the sheer volume of images can be overwhelming for human experts to analyze, especially when looking for specific features or patterns.

By using image classification algorithms, medical images can be analyzed quickly and accurately, allowing doctors and researchers to identify potential cases of a disease or condition, track the progression of a condition over time, and monitor the effectiveness of treatments. Additionally, image classification can be used in computer-aided diagnosis (CAD) systems to assist radiologists and other medical professionals in interpreting medical images and making more accurate diagnoses.

Furthermore, image classification can be trained to detect early signs of certain diseases, and can be a powerful tool in detecting diseases or abnormalities that are difficult for human experts to spot. Automated classification can also be applied to pathological images, with the aim of detecting and diagnosing cancer in an early stage, giving a patient a better chance of survival.

Also, being able to classify medical images in an automatic way can help to reduce human errors that could result in misdiagnosis and therefore contribute to making medical care more efficient and accurate.

In this project, 7022 images of human brain MRI images are going to be used for training a model which classifies the images according to their class names, and the performance of the model will be tested to show how accurately the model performs.

Additionally, the aim is to extract some important features from the images using an autoencoder. The images will be given to the CNN in two different forms; I) normal form, II) output of the autoencoder form. The comparison of the two cases will show how can autoencoders be utilized in such type of a study, and what are its pros and cons.

Finally, trained models will be evaluated by looking at some important metrics such as accuracy score, and training time.

IV. SOLUTION APPROACHES

VGG16 model is used as the deep learning model to categorize brain tumors that are glioma, meningioma, pituitary, and no tumor. The structure of our deep learning model consists of 7 parts. After loading the dataset which is already splitted training and test subsets, every image in the dataset is resized to 128x128 size with 3 channels and normalized.

K-fold cross-validation technique is used for making hyperparameter tuning in VGG16 model. It is a model validation method that can be used to tune the hyperparameters of a VGG16 model, such as the number of epochs, batch size, optimizer, and learning rate. We used k=5 fold and 24 combinations of number of epochs, batch size, optimizer, learning rate for finding best hyperparameters. In every combination, all fold's average accuracy and loss took important role for selecting the best model. So, best model's epoch, batch size, optimizer and learning rate are respectively achieved 8, 30, Adam and 0.0001. The model uses *sparse categorical crossentropy* as the loss function. Then, best model is evaluated with test dataset.

Moreover, accuracy, precision, recall, and the F1-score are used metrics to evaluate the performance of a model on a test dataset. Accuracy calculated as the ratio of correct predictions to the total number of predictions (1). Precision, also known as positive predictive value, measures the proportion of true positive predictions out of all positive predictions made by the model (2). Recall, sensitivity or true positive rate, measures the proportion of positive samples that were correctly identified by the model (3). F1-Score is a measure that combines precision and recall (4).

$$Accuracy = \frac{\textit{Number of correct predictions}}{\textit{Total number of predictions}} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Moreover, dimension reduction technique is used to reduce the complexity of data by removing redundant or irrelevant information. Autoencoding, which is a type of neural network architecture used for unsupervised learning, is implemented for dimension reduction operation. The designed autoencoder architecture uses convolutional layers and pooling layers to perform dimensionality reduction, also known as encoding, and uses transposed convolutional layers and upsampling layers to perform decoding, to reconstruct an image from the lower-dimensional representation. Every image in the datasets encoded with this architecture for the second case of the classifications to be comparable with normal image input case. The autoencoder designed is shown in Fig. 6.

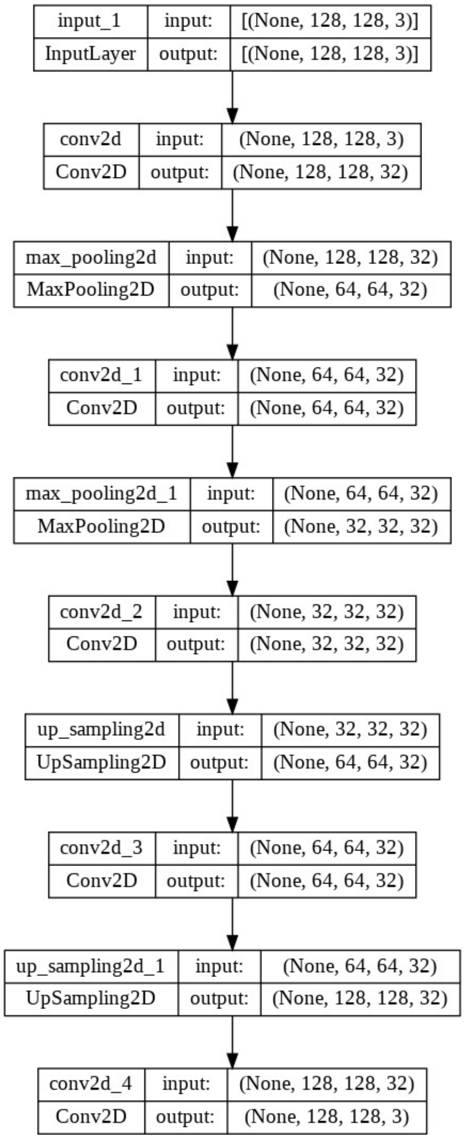


Fig. 6: Designed Autoencoder Architecture

Encoded images is trained in VGG16 model to specify brain tumors. Then, encoded images evaluated with performance metrics on a encoded test dataset. Finally, only VGG16 model and VGG16 with autoencoder model are compared via looking execution time and confusion matrices.

V. EXPERIMENTS AND RESULTS

A. ***Experiment Environment***

For the collaboration of the team in the coding part, Google Colab[2] platform is used with free version. A Python notebook is created and authors are coded different parts of the project together. After the coding is done, training and test processes are done in the same Google Colab environment.

---

[2] Official website of Google Colab

TABLE I: Results comparison

	Normal Image Inputs			**Autoencoded Image Inpus**		
Label	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Glioma	1.00	0.83	0.91	0.91	0.98	0.95
Meningioma	0.82	0.99	0.90	0.98	0.91	0.94
Notumor	1.00	1.00	1.00	1.00	1.00	1.00
Pituitary	1.00	0.94	0.97	0.99	0.98	0.99
Macro Avg	0.95	0.94	0.94	0.97	0.97	0.97
Accuracy	0.95			0.97		

B. Results

The images that are passed into the autoencoder for 10 epochs got more blurry after taken from the decoder. The total processing time of the autoencoder for 10 epochs is $62s$. The input and output of the autoencoder model can be viewed in Appendix 4.

The best VGG16 model selected after the validation process (epochs: 8, batch size: 30, optimizer: Adam, learning rate: 0.0001) is trained with both normal image data without preprocessing, and the image data that is the output of the autoencoder. After training is finished in these two cases, the average *loss* and *sparse categorical accuracy* results are given in Table II.

TABLE II: Loss & Accuracy of the Training Processes

Input	**Average Loss**	**Average Sparse Categorical Accuracy**
Training with Normal Image Data	0.142	0.947
Training with Autoencoded Image Data	0.149	0.944

The model history for the both cases that contain loss and accuracy values can be found in Appendix 2 and 3.

The training time of the model with normal image inputs is $142.8s$ with the configurations of the regular version of the Google Colab computer. The training time is shorter in the autoencoded image inputs case expected, which is $90.4s$. Since the autoencoder's process time takes $62s$, the second case can be considered as $152.4s$. However, to prevent this, image data can be passed to an autoencoder when it is first arrived in real-life cases to be prepared to get passed in the VGG16 model when the time has come.

The performance comparison of the two cases with the metrics that are explained in Section IV is given in Table I. In the table, *macro average* value is averaging the unweighted mean per label in terms of the precision, recall, and f1-score. It can be clearly seen that, when the images are passed to autoencoder before being fed to the VGG16 model, the result gets better in terms of the accuracy and other metrics. In this study's case, normal accuracy, 0.95, is passed with the accuracy of 0.97.

VI. RELATED WORKS

Much work has been done on image classification on the Brain Tumor MRI Dataset. Some of these reached serious accuracy, while others remained low. The following table shows the accuracy values obtained by the previous studies on this dataset.

TABLE III: Previous Studies And Accuracy Scores

Pretrained Model	**Accuracy Score**
ResNet	0.947
VGG-16	0.950
Custom	0.972
Custom	0.968
VGG-16	0.957

As it can be seen from the table, although the data set is the same, accuracy score changes based on the network used for classification task. Furthermore, Although the same pretrained model (VGG-16) is used on the same data set, the accuracy scores might differ. There could be multiple reasons for this. Data augmentation techniques, hyper parameters selected for the model are other reasons that affect the accuracy score.

Additionally, image classification is highly used in the medical field. Below are some of the academic works which uses image classification in the medical field.

In the paper [5], This paper presents a new approach to deep learning, called synergic deep learning (SDL), that aims to address the challenges of intra-class variation and inter-class similarity in medical image classification. The method involves using multiple deep convolutional neural networks (DCNNs) at the same time and allowing them to learn from each other. The representations learned by each pair of DCNNs are concatenated and used as the input to a synergic network that predicts whether the images belong to the same class. If one DCNN makes a correct classification, but the other makes an error, this synergic error serves as a powerful force to update the model. The approach is trained end-to-end using both classification errors from the DCNNs and synergic errors from each pair of DCNNs. Results on several datasets indicate that the proposed SDL model outperforms the state-of-the-art in medical image classification tasks.

In the paper [6], a specialized Convolutional Neural Network (CNN) that utilizes a shallow convolution layer has been developed to identify interstitial lung disease (ILD) in lung image patches. As opposed to traditional feature descriptors that can be complex and specific to certain domains, our CNN framework can automatically learn the relevant image features from lung image patches that are optimal for classification. Additionally, this CNN architecture can be adapted for other medical image classification or texture classification tasks.

In the paper [7], authors have created and made publicly accessible a dataset of 7909 breast cancer histopathology images obtained from 82 patients. This dataset includes both benign and malignant images and the goal is to classify them into two classes using automated methods. This could serve as a valuable tool for clinicians in diagnosing breast cancer. They have also provided preliminary results using current image classification systems and found accuracy ranging from 80-85%, indicating that there is still room for improvement. Through making this dataset and evaluation protocol available to the scientific community, they aim to bring together researchers from both the medical and machine learning fields to work towards this important clinical application.

In the paper [8], authors investigate the use of convolutional neural networks (CNNs) for classifying pneumonia on a chest X-ray dataset. They evaluate three different techniques in our experiments: (1) a linear support vector machine classifier with local rotation and orientation-free features, (2) transfer learning on the VGG16 and InceptionV3 CNN models, and (3) a capsule network trained from scratch. All three techniques were tested with data augmentation as a preprocessing method. Their results indicate that data augmentation is an effective way to improve performance for all three algorithms. Additionally, they found that transfer learning is more effective for classification on small datasets compared to the support vector machine with ORB robust independent elementary features and capsule network.

In the paper [9], authors propose a novel consistency-based method that utilizes unlabeled data to improve medical image classification. They leverage a self-ensembling model and consistency targets to exploit unlabeled data by encouraging prediction consistency under perturbations. Additionally, they introduce a sample relation consistency (SRC) paradigm to consider the relationships among different samples, which reflects the way human diagnose medical images. SRC is superior to existing consistency-based methods that only enforce consistency of individual predictions, as it encourages the model to explore extra semantic information from unlabeled data by enforcing consistency of semantic relations among samples under perturbations. They conduct experiments to evaluate our method on two public benchmark datasets: skin lesion diagnosis with the ISIC 2018 challenge and thorax disease classification with the ChestX-ray14 dataset. Their method outperforms many state-of-the-art semi-supervised learning methods in both single-label and multi-label image classification scenarios.

VII. CONCLUSION

The study showed that there are significant benefits of using autoencoders in image processing projects, specifically medical projects. This is due to the fact that speed is usually the most critical part of a medical issue, and decreasing the workflow of the machine learning models without losing accuracy can save many people's lives. Since both the accuracy and the timing is achieved better using autoencoders, the conclusion can be considered successful.

ACKNOWLEDGEMENTS

Best regards to dear Prof. Alpaydın and Mahir Atmış for their endless love and curiosity to pure science, and the professional ability to reflect this talent to the students with high intensity.

REFERENCES

[1] Mamelak, A. N., & Jacoby, D. B. (2007). Targeted delivery of antitumoral therapy to glioma and other malignancies with synthetic chlorotoxin (TM-601). Expert opinion on drug delivery, 4(2), 175-186.

[2] Meningioma diagnosis and treatment. (2021). CANCER. Retrieved January 7, 2023 from https://www.cancer.gov/rare-brain-spine-tumor/tumors/meningioma

[3] Pituitary tumors. (n.d.). Hopkins medicine. Retrieved January 7, 2023 from https://www.hopkinsmedicine.org/health/conditions-and-diseases/pituitary-tumors

[4] Simonyan, K.,& Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[5] Zhang, J., Xie, Y., Wu, Q., Xia, Y. (2019). Medical image classification using synergic deep learning. Medical image analysis, 54, 10-19.

[6] Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014, December). Medical image classification with convolutional neural network. In 2014 13th international conference on control automation robotics vision (ICARCV) (pp. 844-848). IEEE.

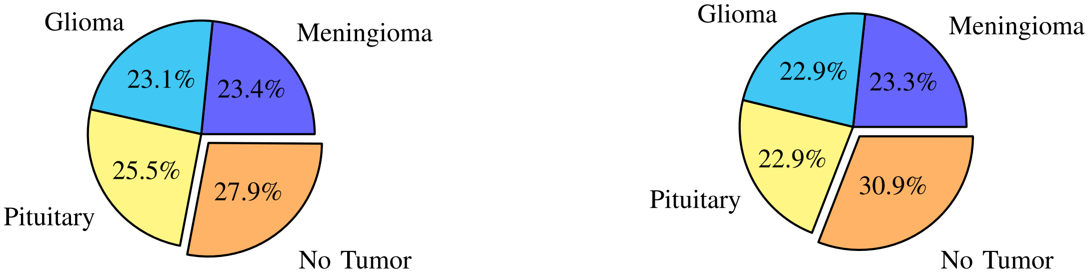
[7] Spanhol, F. A., Oliveira, L. S., Petitjean, C., Heutte, L. (2015). A dataset for breast cancer histopathological image classification. Ieee transactions on biomedical engineering, 63(7), 1455-1462.

[8] Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data, 6(1), 1-18.

[9] Liu, Q., Yu, L., Luo, L., Dou, Q., & Heng, P. A. (2020). Semi-supervised medical image classification with relation-driven self-ensembling model. IEEE transactions on medical imaging, 39(11), 3429-3440.

APPENDIX

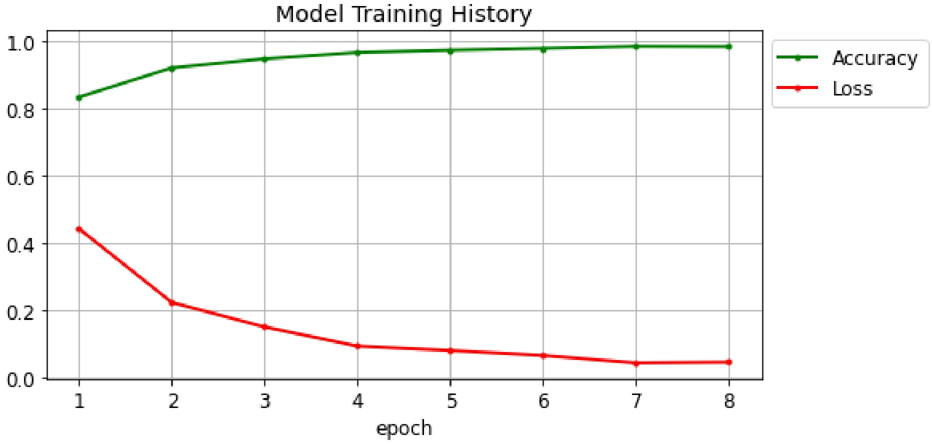
Appendix 1 - *Data distribution of the train and test datasets by percentages*



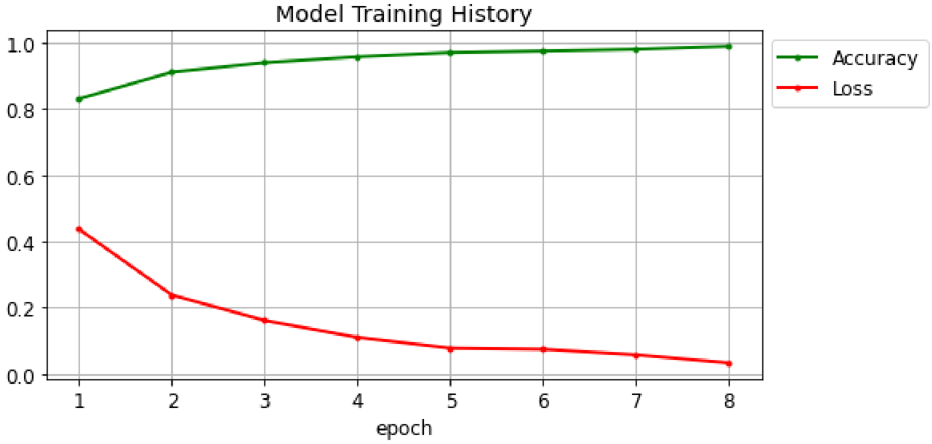
(a) Train Set

(b) Test Set

Appendix 2 - *Model training history of the normal image inputs*



Appendix 3 - *Model training history of the autoencoded image inputs*



Appendix 4 - *Autoencoder model’s input & corresponding output examples*