MRI-Based Brain Tumor Detection Using Deep Convolutional Neural Networks

Humza Khan
College of Computing and Software Engineering
Kennesaw State University
Marietta, Georgia, USA
hkhan30@students.kennesaw.edu

Grant Williams
College of Computing and Software Engineering
Kennesaw State University
Marietta, Georgia, USA
gwilli97@students.kennesaw.edu

***Abstract*-We have created an object detection and classification model that is able to detect and label brain tumors from MRI scans. Normally, after a scan is taken it can take between one and two weeks [10] to get the results of the scan back. These weeks can be tense if you or a loved one is symptomatic of brain tumors. To reduce this waiting time drastically, we look towards machine learning for a solution. As the medical field progresses further and further towards a full embrace of technology, automation using machine learning is a natural progression. Some have performed similar object detection and classification tasks with very high rates of accuracy using k-means clustering and applying skull masking techniques, but while taking upwards of 40-50 seconds using their best-case algorithm. [11, 12] Our system does not require skull masking and is far faster than the methods listed. Our detection model can scan upwards of 10 images per second (or .1 seconds per image) on consumer-grade hardware. We are also able to achieve an average 83.5% accuracy and an average 93.6% precision on our test set. We can detect so quickly due to the use of a YOLOv3 object detection model. The YOLOv3 model has been used in a variety of applications from real time vehicle detection using unmanned aerial vehicles to detecting different growth stages of apples in orchards. [13]**

1. INTRODUCTION

The general way to define a tumor is that it symbolizes the abnormal growth of cells within a human body. Whereas, a mass or growth of abnormal cells in your brain is defined as a brain tumor. Radiology is a branch of medicine that uses imaging technology to diagnose lesions and any abrupt conditions that can alter the human body. Recently, it is widely used to monitor various diseases and helps in providing the appropriate treatment [1]. Main radiology assessments involve Magnetic Resonance Imaging (MRI). This non-invasive technique of acquiring images that are used to examine a specific structure of a human body. Therefore, MRI images of the head of a human body aids in the early discovery of intracranial growths and precise assessment of tumor confines.[2] This leads to a resolution that there is a need to develop such methods that could eventually lead to better and quick tumor detection techniques. This paper provides an overview of such techniques developed using machine learning that could eventually benefit the medical field and help overcome the issues faced when they use manual segmentation.

Many new methods are being proposed with this ever-evolving technology at our hands. [3] None of these techniques possess the necessary threshold of accuracy and/or robustness. So that brings us to a point where we define the sole purpose of this document to design and implement a machine learning process that could eventually lead to an automated brain MRI image segmentation/classification system.[9] This document attempts to design a technique where annotated data is used to train a model and produce a high margin of accuracy. By doing so, our model will be able to detect and classify the tumor based on its various features. It will eventually result in helping medical professionals, saving their precious time that is currently spent manually reviewing brain scans for tumor detection and classification.

Object detection is a task in machine learning that is used to identify the existence, position and several given characteristics in an image. This approach uses a single deep convolutional neural network that splits the input into a grid of cells and that cell directly predicts a bounding box and object classification.[10]

2. RESEARCH PROBLEM

Detection and classification of brain tumors from MRI scans using machine learning techniques involves several steps. First the necessary data must be gathered and annotated if necessary. The model is then trained on a portion of the dataset and then tested on a separate portion of the dataset to check for accuracy. The model must be able to take in MRI scans and output a marked-up version of the scan both locating the tumor as well as classifying the tumor, labeling the markup with a degree of certainty or confidence level.

3. PURPOSE OF THE STUDY

This article aims to provide technological advancement to the medical field as compared to the advancements in some other fields. To provide the right assistance and timely treatment it is essential to speed up the detection process for tumors as well as when it comes to identifying any particular disease or ailment. This particular model emphasizes the quick and accurate tumor detection procedure to help start the treatment process as fast as possible which could aid in helping the patients recover, or to stop any further degeneration of that particular muscle. Manual reading of the scans by medical personals to identify a particular issue slows down the process and may introduce an unwelcome margin of error which is not present with machine learning algorithms.

4. AUDIENCE

The intended audience for this application contains both the set of medical professionals whose lives could be improved by the application as well as the patients who may be given higher accuracy and/or faster results from an automated tumor detection machine instead of relying on a doctor to view the scan manually. MRI machine manufacturers could be a potential audience as well due to the ability of this system to be implemented in line with an existing machine to output both clean scans for a doctor to review as well as marked up scans with potential tumors or abnormalities marked by the model.

5. CONTRIBUTION

Some of the improvements of the model can contribute immensely to the accuracy of the results, some of them are as follows:

- Image enhancement
- Knowledge based pixel classifier construction
- Removal of redundant information
- Post-processing.

Changing the contrast of an image and further changing the color can change the whole outlook of an image which can help in an easy detection process.

Another approach would be to use small size object detection in a complex background, since the brain tumor is not always large and easily visible, therefore identifying a particular irregularity and using that particular part of the image instead of the whole image can also contribute to better results.

Noise blurring and rotating jitter can sometimes cause irregularities in image capture to tackle that particular issue can also help to achieve better results.

6. MOTIVATION

Due to the implications in the medical field that a model such as this may have, the motivation for constructing such an application is found in a desire to advance medical technology into automation to allow doctors to spend more of their time assisting patients and less time spent reviewing every scan. It is also possible that this detection application could advance beyond the abilities of current doctors and detect tumors earlier than the doctor would be able to see them. A member of our team is personally affected by a family member who has undergone multiple rounds of Chemotherapy and noninvasive surgery for brain tumor removal based on relatively small tumors that would likely be detectable with this application.

7. PAPER GOALS AND ORGANIZATION

The model discussed in this article can help detect a tumor accurately and at a very fast rate compared to doctors reading the scans to identify a brain tumor. The model would help the medical field to speed up the diagnosis and treatment procedure. By providing correct treatment at an early stage can help with further complications that could arise with the delay in the detection process.

Section 8 will cover works related to this project to provide background and differentiate our project from others. Section 9 will cover the techniques we used in our training, as well as some of the algorithms we use. Section 10 covers our proposed method including an overview of our object detection architecture alongside our methods for tagging, training, and detection. Section 11 covers our specific results and experiments in detail including our data. Section 12 focuses on our conclusion as well as potential future works that could be pursued with our topic.

8. RELATED WORKS

Here, we provide an overview of the literature works on several MRI brain tumor tissue classification procedures. Several approaches aimed to process multispectral MRI data, by generally applying supervised learning techniques, even though it requires a good chunk of data to make the system learn different discriminative features.

Meiyan Huang et.al proposed using LIPC (local independent projection-based classification) process to classify, the voxel of the head. Also using this technique, path feature is obtained therefore explicit regularization is not needed.[7]

Sergio Pereira et.al proposes computerized means for brain tumors identification and type labeling by utilizing MRI images of brain right at the time of scan and transfer to the system. On the contrary, NN (Neural Networks) and SVM (Support Vector Machine) being the commonly adopted methods lately as they offer better performance.[8]

Baljinder Singh et.al has initially proposed the course of pre-processing for noise elimination from the scans by employing fuzzy filter and a new mean shift based fuzzy c-means algorithm which requires less computing time and offers better segmentation display in contrast to traditional techniques. The above segmentation techniques have a mean field phrase in the traditional fuzzy c-means objective function. Since it's feasible for the mean shift to trace cluster centers quiet effortlessly and swiftly, all the methods can carry out effective diagnosis of the image area.[9]

Haojie Ma and several others also proposed an application for object detection. The main difference being that they used their YOLOv3 model to detect damaged

buildings caused by earthquakes. They used the technique using high spatial resolution remote sensing, they added another layer to the detection model to improved results. Causing the speed of the detection to increase by 5.21s and precision being improved by 5.24%. [23]

9. TECHNIQUES USED

The application is implemented using a YOLOv3 object detection model in Python using Tensorflow-GPU and Keras.[6] The development and training were done on a computer with the following specifications: Intel Core i5-6500, 16GB DDR3, NVIDIA GTX 1060 6GB, and an Intel SSD for the data drive. All the training was done on consumer grade hardware originally designed for gaming in an effort to display the relatively low cost associated with a potential implementation of this project in the large scale.

Our primary dataset contained images inside a MATLAB container file. All the images were extracted from their containers using imfuse and then saving the files to png format for ease of object detection as well as tagging the locations of tumors. Image tagging was done over the course of several days using Microsoft's Visual Object Tagging Tool (VOTT). The tags were saved in CSV format so they could be easily converted to a format that the model could learn from. The libraries and code that were built from contained a python script to convert the CSV to a YOLO-trainable format. Default YOLO weights were then downloaded and converted to be used to train the model.

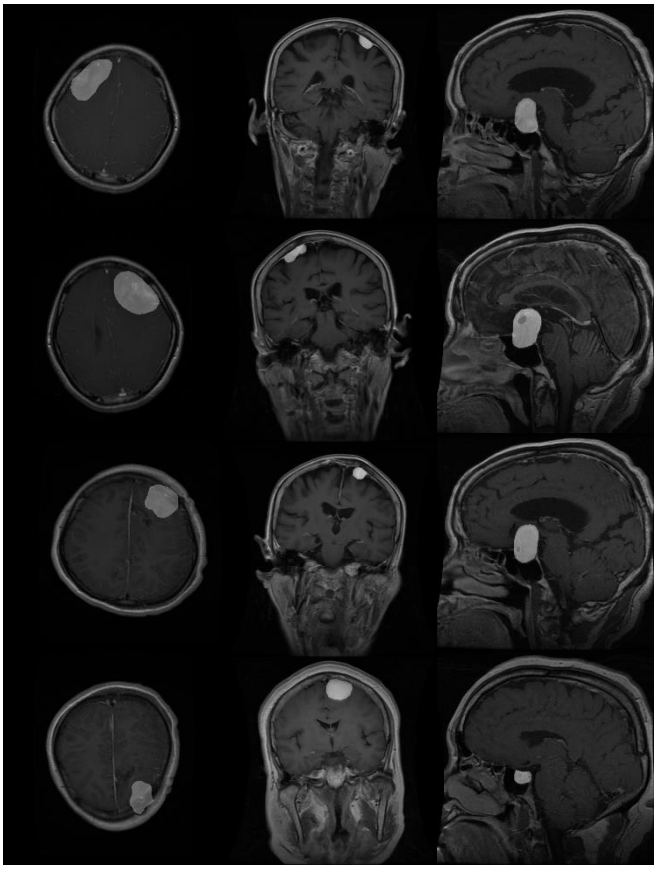

Figure 1: Dataset Example showing Glioma in the left column, Meningioma in the right column, and Pituitary tumors in the right column

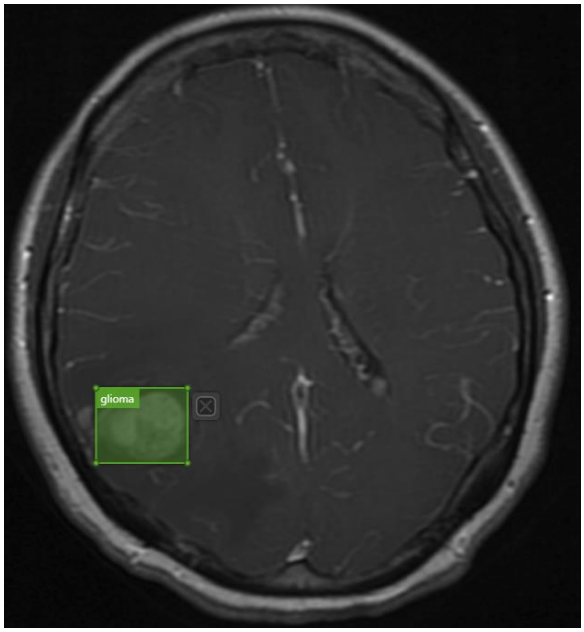

Figure 2: Using the VOTT

Training the model took 24+ hours and yielded a satisfactory result training in about 1000 epochs with a batch size of 32. As the model had to train on nearly 1500 images, there was not much tweaking done to the number of epochs or the batch size as those optimizations could have taken weeks to arrive at. Our model is compiled using the custom "yolo_loss lambda layer" loss function with the "Adam" optimization function. For all training, the "value_loss" variable was monitored to ensure that if we approached an optimal value of loss, we did not waste time training more than was necessary.

The YOLOv3 object detection model uses a YOLO algorithm. The YOLO algorithm acronym stands for you only look once as the image itself is only processed a single time. The image is applied to a convolutional neural network that can not only detect the presence of an object in an image but is able to locate and label the object coordinates as well. The activation function used is a Leaky ReLU function. The optimizer used is an Adam optimizer with an initial learning rate of 1e^-3, or .001, which is adjusted through the training as the model nears an optimal value of loss and accuracy. We use the standard YOLOv3 loss function as it is well respected. The loss function is covered in greater detail in section 10.

After training, the trained object detection weights are saved to a file which can then be used for object detection. The final step is to run the detection script which loads the weights and then uses the object detector to annotate images from the test set with a categorized label for the type of tumor it believes there is and assign a confidence level to that tumor. We chose to only have our detector annotate the image if it has greater than or equal to .4 confidence that there is a tumor there. Such a choice is made because in the medical field there should be a threshold for automation that would require a doctor's further review. The idea of a future implementation would be to flag images below a certain confidence threshold

(in this application: .4) and then allow a doctor to further review any scans with a confidence in a certain threshold of uncertainty (perhaps .1-.4).

From the original source code used, the primary modifications included: using a different dataset, modifying the input and output paths, and creating a custom script to analyze the results for accuracy. A large portion of time was spent annotating the dataset as the dataset that we used came without annotations and our training set ended up containing over 1500 hand annotated images to ensure optimal training accuracy. While the data were categorized into different types of tumors, the images themselves did not have an annotation for the location of the tumor that we could use as input to our YOLOv3 model. We annotated and trained on about 1500 images taking several days to annotate with the correct tumor label.

	A	B	C	D	E	F
1	image	xmin	ymin	xmax	ymax	label
2	g_1904.pn	137.1429	167.342	235.2208	299.2208	glioma
3	g_1841.pn	283.4286	181.1948	338.2857	210.0087	glioma
4	g_1908.pn	159.3074	198.9264	223.5844	270.961	glioma
5	g_1842.pn	282.8745	85.88745	355.4632	172.8831	glioma

Figure 3: A few rows from our annotations CSV file

10. PROPOSED METHOD

This model is divided into three major steps when it comes to the proposed methodology for this article.

A. Tagging

The first step towards achieving desired results was to mark up the images that we found for the brain tumors. 3064 T1-weighted contrast-enhanced images from 233 individual patients, with 3 different kinds of brain tumors including Meningioma, Glioma, and Pituitary tumor were collected and stored accordingly. To train the model we had to provide it identified and tagged images for it to learn and provide accurate results and to perform image tagging Microsoft's Visual Object Tagging tool (VOTT) was used. The model was trained with over 1500 marked up tumor images.

B. Training

To train this model YOLOv3 was used with the dataset that we assembled. To train the model pre-trained dark net weights were acquired and further converted to the format that the designed model can work with which was the YOLO format, with the help of a predesigned python program this step was conducted with great success. This process was the most time consuming, depending on the GPU. As recommended, an NVIDIA graphics card is the only supported GPU to be used for training because TensorFlow relies on CUDA for deep learning applications which is not available on AMD cards and is only available on NVIDIA cards.

C. Testing

The last process was to check for the accuracy of the model. After completing the training process of the model, the next step was to check for the results produced by the suggested machine learning system. To accomplish this process, we provided some sample images of brain tumors for the model to predict and label the results accordingly. After the completion, another python program was created to visualize the predicted results to get the exact estimates of the results that were generated in this process.

D. Architecture

Our model is built on the highly popular YOLOv3 object detection method. Below is a basic layout of the architecture behind how the object detector functions.

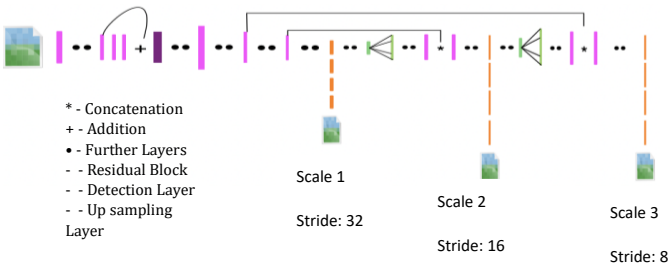

Figure 4: Display of YOLO v3 Network Architecture

11. EXPERIMENTS AND RESULTS

The model was tested on both tumor-free MRI scans as well as scans containing the three types of tumors it was trained on. We were able to achieve an average precision of .9361 and an average recall of .8257. It is probable that of the tumors which were not false negatives that the object detection model did not label the tumors due to not meeting the necessary .4 confidence threshold for labeling.

Training was performed over 1000 epochs and using 1320 samples with a batch size of 32 and a validation set of 151 images. The loss function we use is the standard YOLOv3 loss function as follows:

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]$$
$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]$$
$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}(C_i-\hat{C}_i)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}(C_i-\hat{C}_i)^2$$
$$+\sum_{i=0}^{S^2}1_{i}^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2$$

Figure 5: YOLOv3 Loss Function [18]

We decided to use the standard YOLOv3 loss function as it is very well regarded and since it is the default, we saw no valid reason to use our own custom loss function, though the option is available.

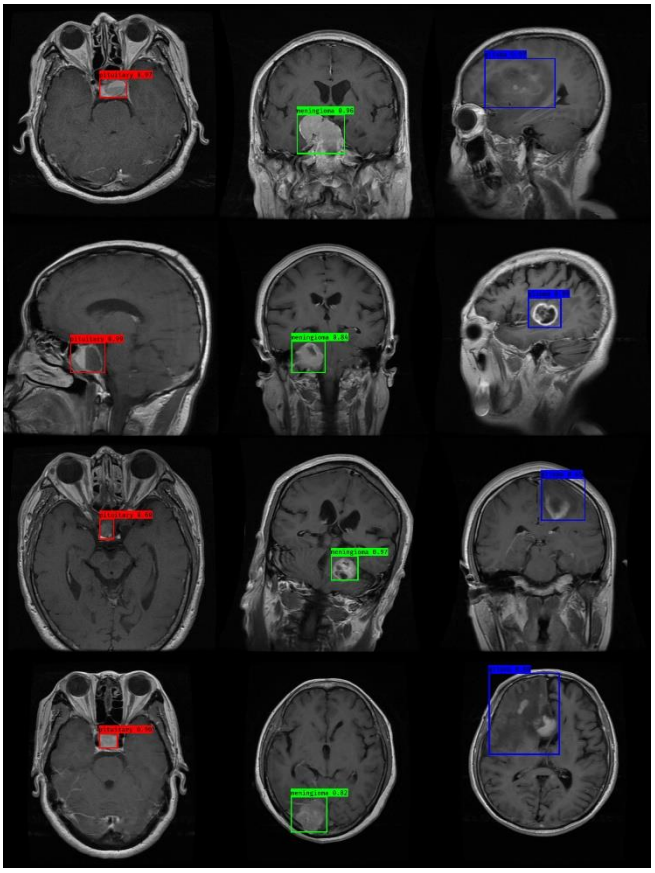

Figure 6: 12 sample tumors discovered and correctly labeled by our detector. From left to right: Pituitary, Meningioma, Glioma tumors.

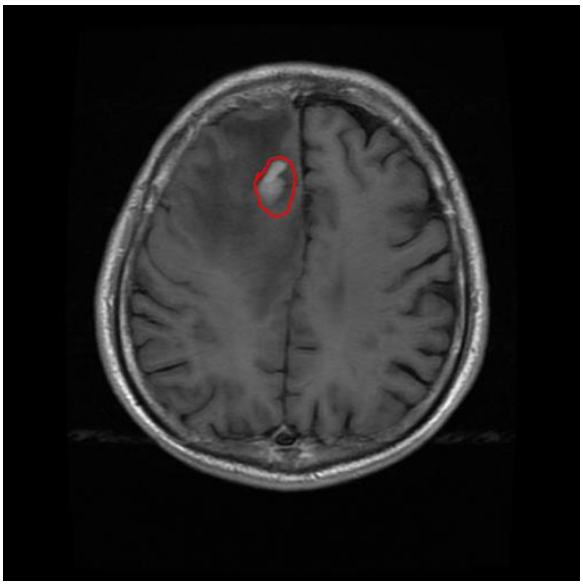

Figure 7: Glioma tumor that our detector did not detect, or label circled in red

We initially attempted to use a dataset that was supposed to contain no images of tumorous brains to ensure our model did not create a high number of false positives. Of this dataset that contained images that were not supposed to contain tumors, we encountered a nearly 16.3% false positive rate which seemed far too high to be acceptable. Upon further review of the data, most of the spots in the MRI which were labeled as tumors appear to be suspicious masses or abnormalities in the brain. There is not a doctor on our research team so we are not qualified to say that the scans are indeed tumorous, but many of the scans from the dataset which were not supposed to contain tumors appear to contain masses in the brain which closely resemble tumors from our training data. Such blemishes on the MRI scans could also just be defects in the image itself or some form of interference noise in the scan. Fortunately, we were able to procure a dataset containing scans that were indeed tumor free which is what allowed us to gather the following data.

As Fig. 8 depicts, our machine has good precision, recall, and F1 scores on average. We would like to see better results, but the results observed are acceptable for triggering a doctor's review of a patients MRI in this instance. Fig. 9 depicts very similar results as they are certainly acceptable to trigger a doctor's review, but we would have liked to see better sensitivity in particular. Fig. 10 is the final accuracy values for both the training and the test set. While the accuracy of the test set may appear low, the YOLOv3 model is not known for its accuracy but for its speed. Our model is faster than all the other implementations we researched by a great margin. As sensitive as it is to catch a brain tumor, our machine detecting a tumor would simply allow the doctor to know they should prioritize a scan containing a known tumor. A scan which our machine does not detect a tumor in would not be disqualified from being reviewed by a doctor. Fig. 11, 12, and 13 represent the primary metrics we traced as our model was being trained: Loss, Mean Squared Error (MSE), and Mean Prediction Error (MPE). The MSE is used in the graph to show the error in our training decreasing as our training progresses. MSE was used as our evaluation metric in our training because the YOLOv3 model does not use a standard accuracy value due to the nature of its activation. As a result, attempting to log its accuracy always results in a value of 0, so MSE must be used in order to get a metric of the running error in the system during training. MPE was utilized for the same reason as the MSE, to provide information about the accuracy of our model while the accuracy was not directly available.

	Precision	Recall	F1
Glioma	.9197	.7493	.8258
Meningioma	.9298	.9075	.9185
Pituitary	.9630	.8880	.9240
Weighted Average	.9361	.8257	.8762

Figure 8: Precision, Recall, and F1 Scores for Glioma, Meningioma, Pituitary, and the Weighted Average of the three tumor classes

	Sensitivity	Specificity
Glioma	.7493	.9457
Meningioma	.9075	.9766
Pituitary	.8880	.9789
Weighted Average	.8257	.9627

Figure 9: Sensitivity and Recall scores for Glioma, Meningioma, Pituitary, and the Weighted Average of the three tumor classes

	Accuracy
Test Set	.8353
Training Set	.9202

Figure 10: Accuracy for both the Test set and the Training set individually

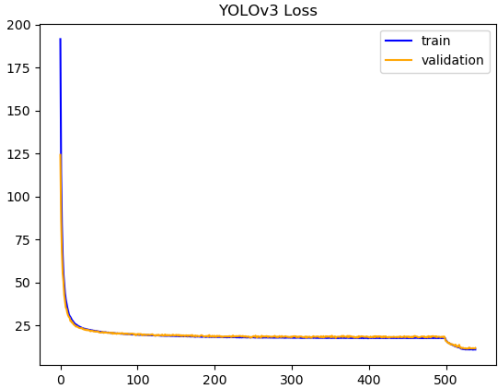

Figure 11: Graph of the loss for the training and validation sets over the 538 epochs of training

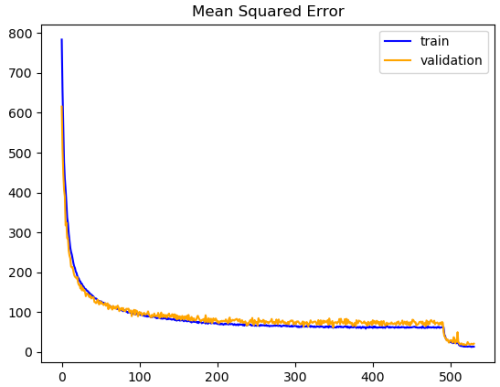

Figure 12: Graph of the Mean Squared Error for the training and validation sets over 538 epochs of training

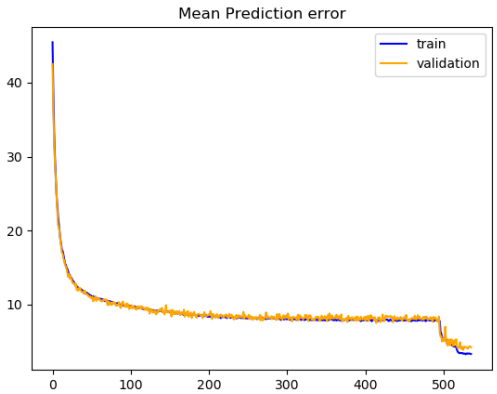

Figure 13: Graph of Mean Prediction Error for the training and validation set over 538 epochs of training

One of the main reasons to use this implementation was the benefit of the speed of this algorithm. While several other methods for automated tumor detection generated more accurate results, generally they took them more time to produce such results. On the other hand, the detection speed of our implementation is superior. Time frame for tumor detection is the most important, assuming a minimum reasonable threshold of accuracy is met. Early detection has proven to be instrumental in the survival rate of patients and our algorithm produces some of the fastest results observed. Having such speed allows for quicker response to the development of tumors as well as better informed patients and doctors.

12. CONCLUSIONS AND FUTURE WORKS

Accurate brain tumor detection is one of the fundamentals when it comes to providing a cure or the necessary treatment to any patient. The proposed technique is completely automated in identifying tumor images based on the YOLOv3 object detection model.[18] It stands unique against some other techniques used to perform the same task because of its accuracy. The experimental analysis was recorded for various datasets and showed a remarkable performance in detecting the appropriate tumors in an image provided as input, which has promising implications for treatment plans.

There are several aspects to further research on the topic, the first and foremost would be to further improve the model with some real-time data samples. Another could be carried out to accelerate the process to apply similar types of applications to perform such tasks in a real-time scenario in each medical field.

13. REFERENCES

[1] R. A. Novelline, "Squire's Fundamentals of Radiology" in, Harvard University Press, 2004.

[2] “Magnetic Resonance Imaging (MRI) – Head”, RadiologyInfo.org

[3] L. Rundo, C. Militello, A. Tangherloni, G. Russo, S. Vitabile, M . Gilardi, G. Mauri, “NeXt for neuro-radiosurgery: A fully automatic approach for necrosis extraction in brain tumor MRI using an unsupervised machine learning technique”, 2017.

[4] A.Farhadi, J. Redmon, “YOLOv3: An Incremental Improvement” arXiv:1804.02640, 2018.

[5] D. Weishaupt, B. Marincek, V. D. Koechli, "An Introduction to the Physics and Function of Magnetic Resonance Imaging" in, Springer, 2003.

[6] D. Garg, P. Goel, S. Pandya, A. Ganatra and K. Kotecha, "A Deep Learning Approach for Face Detection using YOLO," 2018 IEEE Punecon, Pune, India, 2018, pp. 1-4.

[7] M. Huang et al, “Brain Tumor Segmentation Based on Local Independent Projection based Classification”, IEEE Transactions on Biomedical Engineering, IEEE, (2013).

[8] S. Pereira et al, “Brain Tumor Segmentation using Convolutional Neural Networks in MRI Images”, IEEE Transactions on Medical Imaging, (2016).

[9] B. Singh, P. Aggarwal, “Detection of brain tumor using modified mean-shift based fuzzy c-mean segmentation from MRI Images”, © IEEE, 2017, p.p. 536-545.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You only look once: unified, real-time object detection” in arXiv.org, 2016.

[11] J.Brownlee, “How to perform object detection with YOLOv3 in Keras”, on machinelearningmystery.com, 2019.

[12] S. T. Blue and M. Brindha, "Edge detection based boundary box construction algorithm for improving the precision of object detection in YOLOv3," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-5.

[13] T. Weglinski, A. Fabijanska, “BRAIN TUMOR SEGMENTATION FROM MRI DATA SETS

[14] USING REGION GROWING APPROACH”, on MEMSTECH’2011, 2011.

[15] P. Du, X. Qu, T. Wei, C. Peng, X. Zhong, C. Chen “Research on small size Object Detection in complex background”, Chinese Automation Congress, 2018.

[16] C. Liu, Y. Tao, J. Liang, K. Li, Y. Chen,” Object detection based on YOLO Network”, in ITOEC, 2018.

[17] A. Muehlemann, “How to train your own YOLOv3 detector from scratch”, on insightdatascience.com, 2019.

[18] A. Kathuria, “What’s new in YOLO v3?”, www.towardsdatascience.com, 2018.

[19] “How it’s performed – MRI scan”, NHS UK, www.nhs.uk, 2018.

[20] S. Alam, M. Rahman, A. Hossain, K. Islam, M. Ahmed, K. Ahmed, B. Singh, S. Miah, “Automatic Human Brain Tumor Detection in MRI Image Using Template-Based K Means and Improved Fuzzy C Means Clustering Algorithm”, Big Data and cognitive computing, 2019.

[21] K. Rajawat, N. Gatsis, S.J. Kim, G. Giannakis, “Cross-layer design of coded multicast for Wireless Random Access Networks”, 2011.

[22] B. Benjidra, T. Khursheed, A. Koubaa, A. Ammar, K. Ouni, “Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3”, International Conference on Unmanned Vehicle Systems (UVS), Muscat, Oman, 2019.

[23] Ma, Haojie, et al. “Detection of Collapsed Buildings in Post-Earthquake Remote Sensing Images Based on the Improved YOLOv3.” Remote Sensing, vol. 12, no. 1, Jan. 2020, p. 44. EBSCOhost, doi:10.3390/rs12010044.

[24] A. Natarajan, S. Keumarasamy,“Efficient Segmentation of Brain Tumor Using FL-SNM with a Metaheuristic Approach to Optimization”

Dataset

OneDrive Dataset Link

Link Address:

https://kennesawedu-my.sharepoint.com/:u:/g/personal/gwilli97_students_kennesaw_edu/EZDaVpT-WLBFuf0Y3RNsWFsBlDZDXtWjvLNvYv1MqdrhUQ?e=sWpmXc

Source Code Video

Video Explanation of Source Code OneDrive Link

Link Address:

https://kennesawedu-my.sharepoint.com/:v:/g/personal/gwilli97_students_kennesaw_edu/ESMTzL5LBPVIhpY2FEl8IU8Bty_mE8zBWn3zGINZsfZ2aQ?e=UDQdp7

Source Code

Here the OneDrive link to the Source Code for our project

Link Address:

https://kennesawedu-my.sharepoint.com/:u:/g/personal/gwilli97_students_kennesaw_edu/EX21O8hHEi1Jjsl2dTlJZnIB6z2JQxpuQwWMhhUyc2IvlQ?e=bjJcCu