2021 International Conference on Intelligent Technologies (CONIT)
Karnataka, India. June 25-27, 2021

Deep Learning Based Brain Tumor Detection and Classification

Nadim Mahmud Dipu
Dept. of Electrical and
Computer Engineering
North South University
nadim.dipu@northsouth.edu

Sifatul Alam Shohan
Dept. of Electrical and
Computer Engineering
North South University
sifatul.shohan@northsouth.edu

K. M. A. Salam
Dept. of Electrical and
Computer Engineering
North South University
kazi.salam@northsouth.edu

2021 International Conference on Intelligent Technologies (CONIT) | 978-1-7281-8583-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CONIT51480.2021.9498384

***Abstract*—One of the most crucial tasks of neurologists and radiologists is early brain tumor detection. However, manually detecting and segmenting brain tumors from Magnetic Resonance Imaging (MRI) scans is challenging, and prone to errors. That is why an automated brain tumor detection system is required for early diagnosis of the disease. This paper proposes two deep learning based approaches for brain tumor detection and classification using the cutting-edge object detection framework YOLO (You Only Look Once) and the deep learning library FastAi, respectively. This study was done on a subset of the BRATS 2018 dataset that contained 1,992 Brain MRI scans. The YOLOv5 model achieved an accuracy of 85.95% and the FastAi classification model achieved an accuracy of 95.78%. These two models can be applied in real-time brain tumor detection for early diagnosis of brain cancer.**

***Index Terms*—Brain Tumor, Brain Tumor Detection, YOLO V5, Convolutional Neural Networks, FastAi, Brain Tumor Classification, Convolutional Network, Deep Learning, Tumor Segmentation, Neural Networks**

I. Introduction

Numerous studies have been conducted on medical image segmentation in recent years [1], [2], [3], [4], [5], [6], [7]. The same is true in the case of brain tumor detection [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Researchers have proposed various methods for detecting brain tumors in brain Magnetic Resonance Image scans [17], [18], [19]. Each of these methods has achieved varying degrees of accuracy.

Sapra et al. [6] proposed image segmentation techniques and a modified probabilistic neural network (PNN) model to carry out automatic brain tumor classification. Their method outperformed previous PNN based models and it achieved 100% accuracy in their classification task. Amin et al. [7] suggested differentiating cancerous and non-cancerous MRI of the brain tissue by going through a three-step process. Those steps include image processing, feature extraction, and image classification. Threshold segmentation and morphological operations were suggested by Zhang et al. [8] for brain tumor detection and segmentation. Alfonse et al. [9] proposed a slightly different approach in which Fast Fourier Transform is used for feature extraction, Minimal-Redundancy-Maximal-Relevance is used for feature reduction, and finally, the SVM was utilized for classification. By utilizing these three methods, they achieved an accuracy of 98.9%. A new study performed by Dong et al. [10] suggested a fully convolutional neural network-based system to solve the brain tumor detection and segmentation problem. Deepak et al. [11] proposed a classification system that uses deep transfer learning and pre-trained GoogLeNet[12] to extract features from MRI images. The experiment performs a patient-level five-fold cross-validation [13] process. Their proposed method outperforms all state-of-the-art records achieving 98.9% classification accuracy.

A novel convolutional neural network (CNN) [14] is proposed to classify multi-grade brain tumors [15]. They used deep learning techniques to segment tumor regions from MR images then use data augmentation to train the proposed model, which shows convincing performance compared to existing approaches. They used deep learning techniques to segment tumor regions from MR images then used data augmentation to train the proposed method, which shows convincing performance compared to other existing methods.

Abiwinanda et al. [16] proposed the most straightforward CNN architecture to classify different types of brain tumors. This simple architecture archives 98.51% training accuracy and 84.19% validation accuracy without any region-based prior segmentation.

Each of the studies that we have discussed thus far have certain drawbacks. One of the biggest disadvantage of utilizing the algorithms that have been used in these studies are computationally expensive. Not only do these methods require an exorbitant amount of processing power, they also need to be trained for an extended period. These models had to be trained on large datasets in order to gain satisfactory accuracy. Furthermore, all of these models have to be trained on previously annotated and preprocessed datasets and they will not work on custom annotated MRI scans.

In order to overcome these drawbacks and challenges, we have decided to study the performance of the YOLO object detection framework [20] and FastAi library [23] for the detection and classification of brain tumors, respectively. YOLO has become remarkably popular in the past few years as a neural network based framework for object detection. And numerous scientific papers have been published that propose YOLO based object detection models [24], [25], [26]. This architecture has not been extensively studied for detection of brain tumors. There has been only one paper published on brain tumor detection that proposed a YOLO V2 model that was trained on several brain tumor datasets [21]. The objective of this study is to build fast deep learning models for brain tumor classification and detection using the latest versions FastAi library and YOLOv5 framework to build an automated real time brain tumor diagnosis system.

II. Dataset

As mentioned before, we used the BRATS 2018 dataset for this study [22]. We received this dataset from the Medical Segmentation Decathlon challenge. This large annotated medical image dataset is made for the development and evaluation of detection and segmentation algorithms. It had ten different segmentation tasks, but we only used the brain tumor segmentation dataset, a part of the BRATS 2018 dataset. Figure 1 shows some sample MRI images from our dataset.

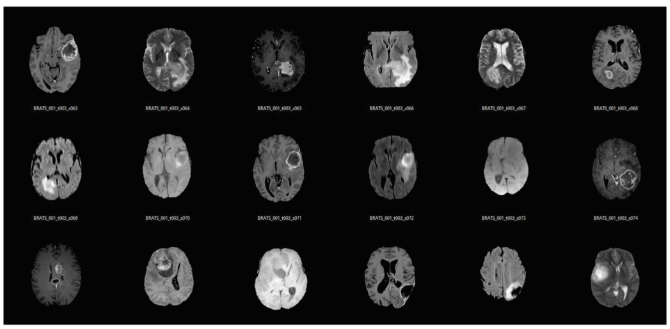

Fig. 1. A look at the images of the Brain MRI scans in the BRATS 2018 dataset.

This dataset consists of 498 NIfTI (Neuroimaging Informatics Technology Initiative) files, and each of the files contained 620 images of four MRI scans. Each scan had 155 slices. For the sake of simplification and for making it easy for us to train our segmentation and classification models, we took four images out of the 620 images to represent each MRI scan, and thus our dataset contained 1,992 MRI images. For the classification task, we split it into two files named ”Yes” and ”No,” depending on whether the scan contained a tumor or not. This can be observed in figure 2. As the figure shows each of the images have a class associated with it that represents whether the MRI scan contains a brain tumor or not.

III. Methodology

A. Classification

The classification of the MRI scans was done with the help of the FastAi V2 library. It is a library that's made explicitly for building deep learning-based applications. FastAi V2 was released in August of 2020, and it features sophisticated built-in deep learning models and methods. With the help of this library, we were able to build, train and test our classification model by writing only a few lines of code. At first, we created a training directory that contained two nested directories (Yes and No). Then, we populated each of the folders with images from our dataset based on whether or not they contained a

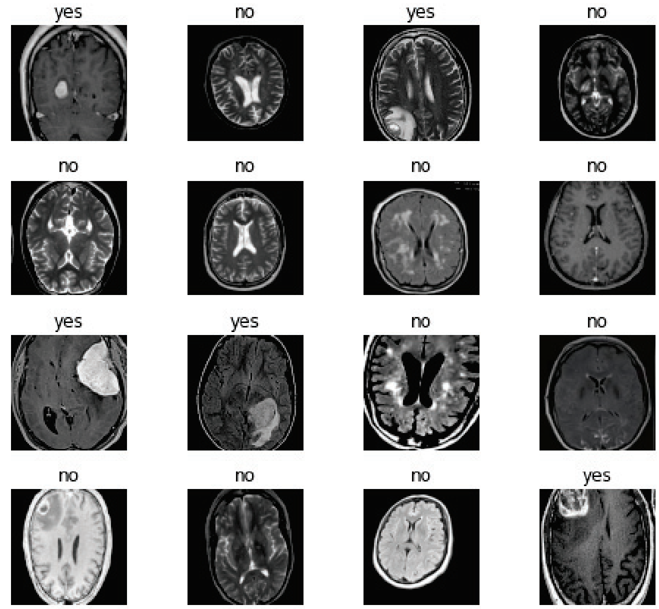

Fig. 2. Brain Tumor MRI images being split into two different classes (”Yes” and ”No”).

tumor or not. We then proceeded to build a testing directory whose purpose is to be used for validating how the model performs on unseen images. When the dataset was ready, we started building the model in Google Colab. The training data was split into training and validation using the RadomSplitter function. The images were then resized to be 224 x 224 using the RandomResizedCrop function. We built our object classification model using the cnn_learner class of the FastAi library. This model was based on resnet34, which is a neural network architecture. We had to train the model for 100 epochs until we received an accuracy score of more than 90%. Figure 3 shows the loss vs learning rate curve of the training process.

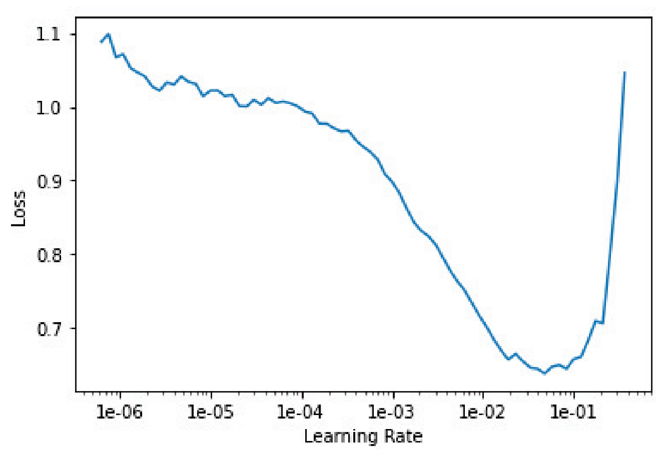

Fig. 3. Loss V learning rate graph of the fastAi CNN model

B. Object Detection Using YOLO V5

This task aimed to identify the presence of brain tumors in the MRI scans taken from the BRATS dataset. To automate this process, we had to train a custom object detection model to detect the relevant objects (brain tumors) in the images.

1) YOLOv5 Model: We built our object detection model using the latest version of YOLO, i.e., YOLO V5. This model is maintained using the Darknet framework, and it provides a single network that can be used for both object classification as well as prediction using bounding boxes. The overall architecture of the Darknet framework can be seen in figure 4. Architecturally speaking, YOLO V5 is quite similar to YOLO V4, the main difference being the fact that YOLOv5 is written in PyTorch. That makes version 5 of YOLO much faster and more lightweight. We used the YOLO V5 model that was trained using the benchmark COCO dataset. This model was further trained using our custom annotated MRI images. Unlike the other neural network-based frameworks used for object detection, YOLO does not have a complex pipeline and it is a single convolutional neural network. It features 24 convolutional layers that extract features from the images and two fully connected layers used for bounding box prediction. This network is implemented using the Darknet framework.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 4. Darknet 53 architecture

2) Annotating Training Images: In order to build our YOLOv5 object detector we required custom annotated images. We used the LabelImg software to annotate the BRATS MRI scan images with bounding boxes around the tumor areas. This software generated a text file for each of the images in the format compatible with the YOLO V5 model.

3) Setting Up The YOLOv5 Environment in Google Colab: We built and ran our custom object detector model in Google Colab as it's an excellent tool for building data science projects. At first, we cloned the YOLOv5 repository to our Colab notebook environment. And then, we installed all the necessary dependencies. This was done in order to set up the programming environment required for running object detection models. We ran our training on a GPU environment instead of the CPU environment. The reason for that is neural network-based models, especially those used for object detection, run significantly faster on GPUs. For this study, the Tesla P100 GPU was used that was provided by Google Colab. We then mounted our custom annotated dataset from drive to our Colab runtime.

4) Training The Custom YOLOv5 Object Detector: Before we could start training our model, we had to define some of the critical parameters. Those parameters were image size, batch size, number of epochs, the path to our data, model's configuration, the path for storing the weights generated by YOLO, etc. After setting the parameters, we ran the training command.

5) Evaluating the Performance of the YOLOv5 Object Detector: After the training was successfully completed, we proceeded to evaluate the performance of the training procedure. We visualized the validation metrics by using the utils library of python. These validation metrics can be seen in figure 5.

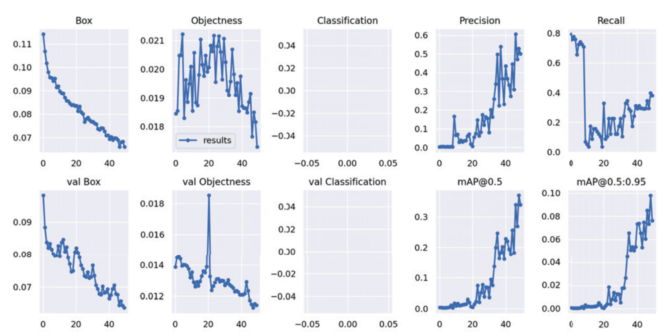

Fig. 5. A look at the different validation metrics used by YOLO.

While analyzing our results in a more comprehensible way, we used metrics such as accuracy, precision, recall, mean average precision, F1 score, etc. Here are the formulas used to determine these metrics:

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{1}$$

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

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

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (5)$$

Here,

- TP = True Positive (The number of images that are accurately detected to be positive)
- FP = False Positive (The number of images predicted to be positive but actually were negative)
- TN = True Negative (The number of images that were correctly predicted to be negative)
- FN = False Negative (The number of images that were incorrectly predicted to be negative)
- n = The number of classes
- AP = The average precision of a class

6) Running the Detector on Test Images: Now that we had a fully trained model on our hands, it was time for us to run the model on the test set to determine how accurate the model was at detecting brain tumors on MRI images. After the training was completed, the weights generated during the training process had been stored in the weights folder. We specified the path to the weights file, and we also specified the location of the test set. After that, we ran the command for running the detector on each of the test images. This model ran at a swift pace on the Tesla P100 GPU as, on average, it only took 7 ms for processing each of the images. In total, it took the detector 3.486 seconds to detect brain tumors on the test set comprised of 498 images.

IV. Experimental Results

A. Classification

We trained the FastAi based CNN model for up to 100 epochs for the classification task. Its performance was satisfactory, and it achieved an accuracy of $\mathbf{95.78}\%$ when it was run on the testing set. We had split the dataset consisting of 1992 images into a training set comprising 1494 images ($\mathbf{75}\%$ of the total MRI scans) and a testing set comprising 498 images of ($\mathbf{25}\%$ of the total MRI scans).

The performance of the model was measured using the testing test that contained images that were new and unseen to the model as shown in table 1. This simulated how the model would perform in real life scenarios with previously unseen MRI scans. The precision of this model was $\mathbf{96.70}\%$, and it's recall was $\mathbf{95.65}\%$. Furthermore, the model achieved an F1 score of $\mathbf{96.17}\%$. This performance can be analyzed using the confusion matrix generated by the model as shown in figure 6.

TABLE I
Performance of FastAi classification Model

Model	Classification (FastAi)
Epochs	100
Accuracy	$\mathbf{95.78}\%$
Precision	$\mathbf{96.70}\%$
Recall	$\mathbf{95.65}\%$
F1 Score	$\mathbf{96.17}\%$

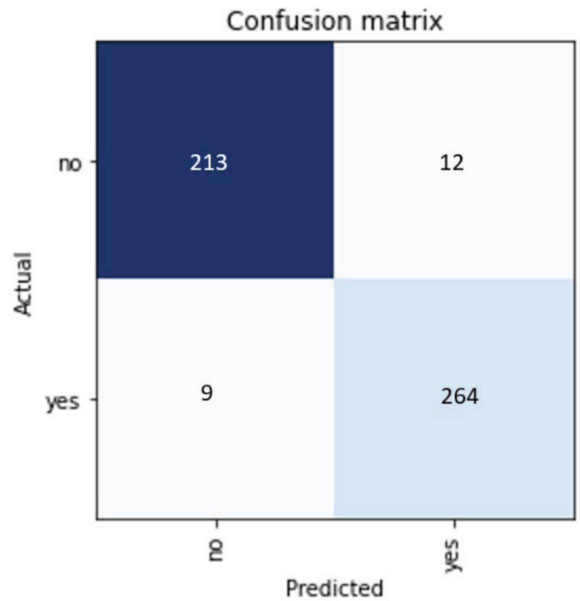

Fig. 6. Confusion matrix produced by the classification model.

A comparison of our FastAi classification model with some of the classification models implemented by other studies are shown in table 2.

TABLE II
Comparison of FastAi classification Model with Other Studies

Author	Methodology	Accuracy
Sachdeva et al. [27]	Artificial Neural Network (ANN)	$\mathbf{85.23}\%$
Shenbagarajan et al. [28]	ACM + ANNLM	$\mathbf{93.74}\%$
Antonie et al. [29]	SVM	$\mathbf{70}\%$
Our Model	FastAi	$\mathbf{95.78}\%$

B. Brain Tumor Detection in YOLOv5

The object detection model created by YOLOv5 also performed exceptionally well on the test set images. It was able to reach an accuracy of $\mathbf{85.95}\%$ on the previously unseen images. After running the model on the test set, we received output images that contained bounding boxes around the tumor regions that indicated the existence of a brain tumor on the MRI scan images. Figure 7 shows the ground truth of the training set images.

Moreover, we can observe the output generated by the model in figure 8. As it can be seen from the figure that the output images contained bounding boxes as well as a confidence score that indicates how accurate the bounding box is.

The YOLOv5 model was trained for 250 epochs. This model achieved an accuracy score of $\mathbf{85.95}\%$, an F1 score of $\mathbf{88.30}\%$ and a mAP at .5 score of $\mathbf{89.30}\%$. These performance metrics of the YOLOv5 model can be observed in table 2.

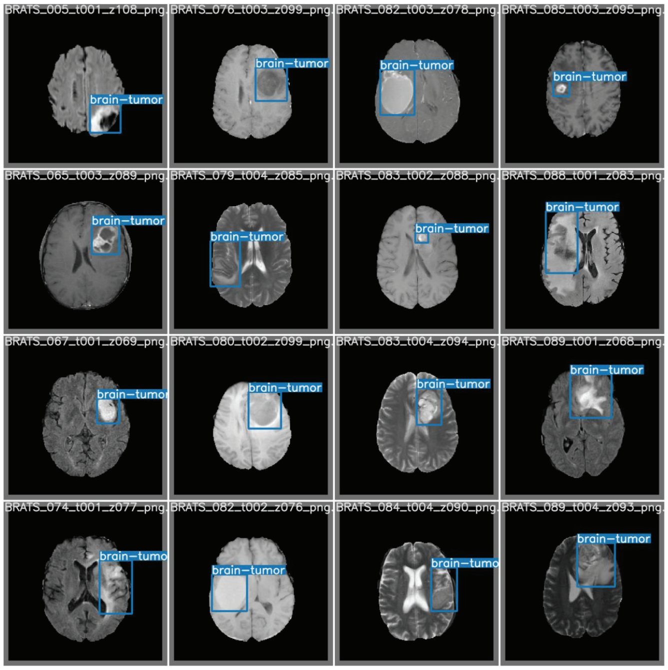

Fig. 7. Ground truth of training data.

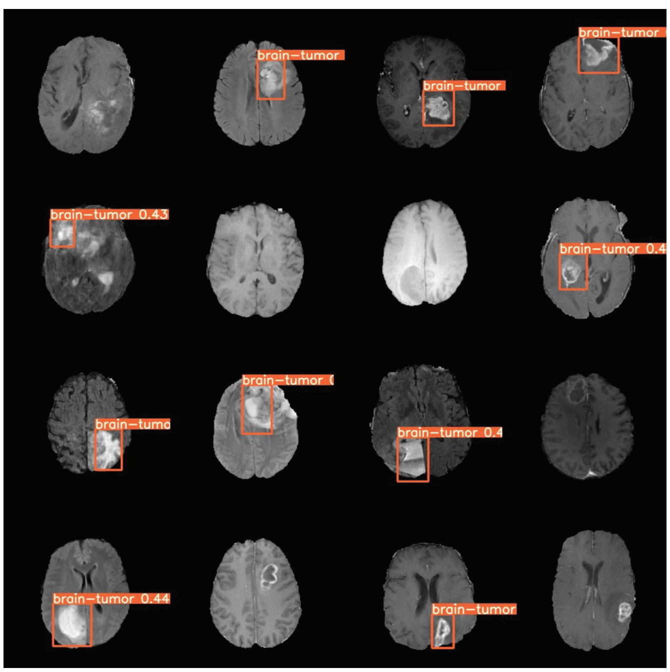

Fig. 8. Output generated by the model on the test set.

V. Conclusion

In this paper, we have proposed two different deep learning-based approaches for brain tumor detection and classification using YOLOv5 and FastAi, respectively. These models will be instrumental in building an automated brain tumor detection system for real-time use. Such a system will go a long way in improving the accuracy and effectiveness of brain tumor and cancer diagnosis. And it will drastically improve the capabilities of our healthcare system. Our CNN based classification model achieved an accuracy of $\mathbf{95.78}\%$ and our YOLOv5 based detection model achieved an accuracy of $\mathbf{85.95}\%$. However, our experiments have some limitations as well. We only used a subset of the BRATS 2018 dataset as we did not have the hardware necessary for running our model on such a large image-based dataset. This system will provide much better results if it were to be trained on a larger dataset for a more extended period. In spite of these limitations, these two deep learning based models described in this paper can be used for real time brain tumor detection and diagnosis on MRI scans.

TABLE III
Performance of YOLOv5 Object Detection Model

Model	Object Detection (YOLOv5)
Epochs	250
Accuracy	$\mathbf{85.95}\%$
Precision	$\mathbf{92.67}\%$
Recall	$\mathbf{84.33}\%$
F1 Score	$\mathbf{88.30}\%$
mAP@50	$\mathbf{89.30}\%$

References

[1] Pham, D. L., Xu, C., Prince, J. L. (2000). Current methods in medical image segmentation. Annual review of biomedical engineering, 2(1), 315-337.

[2] Sharma, N., Aggarwal, L. M. (2010). Automated medical image segmentation techniques. Journal of medical physics/Association of Medical Physicists of India, 35(1), 3.

[3] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In Deep learning in medical image analysis and multimodal learning for clinical decision support (pp. 3-11). Springer, Cham.

[4] Taha, A. A., Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC medical imaging, 15(1), 1-28.

[5] Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Rad, A. E., Rehman, A., Uddin, M. (2014). Medical image segmentation methods, algorithms, and applications. IETE Technical Review, 31(3), 199-213.

[6] Sapra, P., Singh, R., Khurana, S. (2013). Brain tumor detection using neural network. International Journal of Science and Modern Engineering (IJISME) ISSN, 2319-6386.

[7] Amin, J., Sharif, M., Yasmin, M., Fernandes, S. L. (2017). A distinctive approach in brain tumor detection and classification using MRI. Pattern Recognition Letters.

[8] Zhang, S., Xu, G. (2016). A novel approach for brain tumor detection using MRI Images. Journal of Biomedical Science and Engineering, 9(10), 44-52.

[9] Alfonse, M., Salem, A. B. M. (2016). An automatic classification of brain tumors through MRI using support vector machine. Egy. Comp. Sci. J, 40(3).

[10] Dong, H., Yang, G., Liu, F., Mo, Y., Guo, Y. (2017, July). Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In annual conference on medical image understanding and analysis (pp. 506-517). Springer, Cham.

[11] Deepak, S., Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. Computers in biology and medicine, 111, 103345.

[12] Ballester, P., Araujo, R. (2016, February). On the performance of GoogLeNet and AlexNet applied to sketches. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).

[13] Berrar, D. (2019). Cross-validation. Encyclopedia of bioinformatics and computational biology, 1, 542-545.

[14] Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). Ieee.
[15] Sajjad, M., Khan, S., Muhammad, K., Wu, W., Ullah, A., Baik, S. W. (2019). Multi-grade brain tumor classification using deep CNN with extensive data augmentation. Journal of computational science, 30, 174-182.
[16] Abiwinanda, N., Hanif, M., Hesaputra, S. T., Handayani, A., Mengko, T. R. (2019). Brain tumor classification using convolutional neural network. In World congress on medical physics and biomedical engineering 2018 (pp. 183-189). Springer, Singapore.
[17] Ronneberger, O., Fischer, P. Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, .
[18] Diakogiannis, Foivos Waldner, Francois Caccetta, Peter Wu, Chen. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing. 16. 94-114. 10.1016/j.isprsjprs.2020.01.013.
[19] Künsch, H., Geman, S., Kehagias, A. (1995). Hidden Markov Random Fields. The Annals of Applied Probability, 5(3), 577-602. Retrieved April 25, 2021, from http://www.jstor.org/stable/2245115
[20] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, ”You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on
[21] C. Öksüz and M. K. Güllü, ”Brain Tumor Localization Using Yolo v2,” 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302385.
[22] Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., ... Eaton-Rosen, Z. (2018). Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. arXiv preprint arXiv:1811.02629.
[23] Howard, J., amp; Gugger, S. (2020). Fastai: A layered api for deep learning. Information, 11(2), 108. doi:10.3390/info11020108 Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
[24] , J., Wang, H., Zhang, X., Li, X., Yang, H. (2017, October). An object detection system based on YOLO in traffic scene. In 2017 6th International Conference on Computer Science and Network Technology (ICCSNT) (pp. 315-319). IEEE.
[25] Shafiee, M. J., Chywl, B., Li, F., Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. arXiv preprint arXiv:1709.05943.
[26] Nguyen, D. T., Nguyen, T. N., Kim, H., Lee, H. J. (2019). A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(8), 1861-1873.
[27] , Sachdeva, J., Kumar, V., Gupta, I. et al. Segmentation, Feature Extraction, and Multiclass Brain Tumor Classification. J Digit Imaging 26, 1141–1150 (2013). https://doi.org/10.1007/s10278-013-9600-0
[28] SA. Shenbagarajan, V. Ramalingam, C. Balasubramanian and S. Palanivel, Tumor Diagnosis in MRI Brain Image using ACM Segmentation and ANN-LM Classification Techniques, Indian Journal of Science and Technology, Vol 9(1), DOI:10.17485/ijst/2016/v9i1/78766, January 2016.
[29] Antonie L., Automated Segmentation and Classification of Brain Magnetic Resonance Imaging, C615 Project, 2008.