

# Colored MRI biomedical image tumor classification and segmentation based on transfer learning of modified Y-Net

Nassr Nafeaa Khamis<sup>1\*</sup> and Rahma Saadi Mustaf<sup>2</sup>

<sup>1,2</sup>A-Nahrain University, Systems Engineering Department, Aljadriya, Baghdad, Iraq

**Abstract.** By incorporating the colored MRI identification synthesis into the MRI segmentation model with transfer learning AI Y-Net, this study clearly shows the high potential of a multidisciplinary system-level study for diagnoses. This way, such a system can provide integrity of the goal without compromising the quality of each one and saving time consumption. Another alternative to such integration is to be used for enhancement and segmentation that is accurate and robust to the variabilities in scanner and acquisition protocols. System Level Simulator is the deep learning based on Kearse AI deep learning network specified to Y-VGG16 net results of outstanding performance in medical image segmentation. Based on the literature, there are different AI models for the diagnosis system, which are different of what is proposed in this paper. A partial-frozen network is applied to the U-net to compare results between different fine-tuning FT strategies. The network operation is also evaluated depending on the dataset size, showing the importance of the combination of dataset, TL and data augmentation (DA). Transfer learning (TL) helps us for MRI medical image segmentation deep learning with more accurate performances of the TL technique. The system hybrid the Y-Net architecture with Transfer learning to reduce the domain-shift effect in brain MRI segmentation results of the automated deep learning segmentation approach.

## 1 Introduction

The science of medical image improves medical diagnosis in an efficient and effective manner that helps to find solutions of medical issues. There are different market techniques such as Magnetic resonance imaging (MRI), Endoscopy, Computed tomography (CT), X-ray, Positron emission tomography (PET), Ultrasound, etc. MRI is an important role in brain tumor diagnosis for clear human brain structure. Generally, brain tumors are difficult to diagnose because they are abnormal cell aggregations that grow within the brain tissues [1]. To extract information from MRI imaging, artificial intelligence techniques have great attention due to their successes in intelligent medicine [2]. Convolutional neural networks have a significant impact on digital image processing especially medical imaging [3]. So, the main focus of this article is to gain knowledge of AI and how to be applied for medical diagnosis enhanced for

---

\* nassrnafea@gmail.com

MRI brain classification. The method presented in the next sub-sections is mainly based on machine learning, specifically deep learning. Where the general principles of deep learning are presented to be motivated for identification and segmentation tasks in neuro-oncology to assign their abilities and limitations [4]. Convolutional Neural Networks (CNN) [LeCun 1995] are used for image processing and analysis (classification, segmentation) [5]. The local operations such as upsampling, convolution, and pooling are based on the application of spatial relations between pixels in 2D or voxels in 3D. The objectives of such a design are to reduce the network parameters number and to reduce the computational costs due to very large image inputs where parallelized operations are applied. Generally, CNNs based image segmentation training is in an end-to-end manner. Tumor segmentation of MRI medical imaging is particularly important for very challenging tasks due to MRI tissue nature and the diversity of types of equipment [6]. Most of the current methods for image segmentation are CNN-based, such particular methods are brain tumor segmentation as [Kamnitsas 2017a, Myronenko 2018] [7], [8]. The objective of the methodological contributions of this article is to address the evaluation of the main results by using a new DL proposed network, Y-net. Y-net for the purpose of identification and segmentation performs thousands of convolutions, max-pooling, and up-samplings compared to typical U-net [Ronneberger 2015] achieved segmentation only [9]. During an iteration of the training, the different network step operations are stored in the memory of the processing unit's local server. The backpropagation algorithm is applied to compute the gradients of the loss function. A typical MRI is composed of several millions of voxels. Generally, current available GPUs suffer of a shortage of memory spaces for MRI requirements. For this reason, current segmentation models have usually trained cloud networks.

## 2 Related Work

Nowadays, many recent researchers have focused on the segmentation and classification of brain tumors. The author [10] demonstrated five steps for the segmentation and classification of brain tumors. Firstly, an enhancement source image was applied to a 17-layer CNN architecture segmentation network, secondly, a modified MobileNetV2 CNN was used for feature extraction and selection using the entropy-controlled method. Thirdly, the classification using Multiclass Support Vector Machine M-SVM. Fourthly, a segment tumor using 17-Layer CNN was applied and fifthly classify tumor into either meningioma, glioma, or pituitary results of Accuracy 97.47%, Dice coefficient index of 96.71%.

While, Munir et al. [11] segmented brain tumors using a U-Net CNN. The goal of the authors was to reduce large structural deviations and spatial variability. The dataset are taken from the BraTS 2019 dataset, which is processed by CNNs that are effective at recognizing brain tumors. The introduced system, as a result, has a Dice coefficient of 0.9694. Furthermore, the authors Pravitasari et al. [12] demonstrated a U-Net the -VGG16 for segmentation of tumor consisting of two path contracting, another path containing convolutional layer with weights transferred from VGG -16 which has been frozen, and an expensive path for a new convolutional layer to complete extraction. The whole network resulted in about 2,324,353 Parameters and consume 32 minutes to train the model. this model achieved a loss of 0.054 and an accuracy of 0.96 and Ghazanfar et al. [3] demonstrated three-stage for segmentation and classification of tumor. The first stage build binary classification using CNN which consists of nine layers with total trainable parameters of about 217,954, the second stage segmented tumor using Neighboring FCM Which extracted tumor region from tumorous by discarding normal tumor, while the third stage classified image into either edema, necrosis, enhancing or non-enhancing with Total Trainable Parameters about 241,624 as results accuracy 96.88, 96.29 for both binary and multi-classification.

### 3 MRI Imaging Deep Learning Network

Multiple MR sequences can be used to image a variety of tumor tissues, including the necrotic core, active rim, and edema. The varying abilities of human experts make manual tumor segmentation difficult and consume a lot of time. While programmed division strategy could help in checking the growth movement by giving the specific confinement of cancer sub-districts and their volume. Nevertheless, the thousands of operations make CNNs segmentation of tumors in large medical images a very difficult task of pooling, convolutions, and up sampling. Current CNN's proposed architecture is designed to extract 2D and 3D context from input images [13].

Probabilistic priors of brain tumors are difficult to be used due to varieties in location, size, and shape. Also, MR tumor image voxels are highly overlapped with the intensities of other brain structures. Depending on the acquisition system, MR image intensities highly vary from one imaging center to another. Due to these aspects, information has to be analyzed.

Since the needing of extensive computing and time resources pre-trained models are used as Transfer Learning (TL). TL is used to improve the target learning based on the knowledge of the source task to reduce training time [13]. Such a task is implemented by hybrid the U-Net with another network such as VGG-Net [12] simplifying the U-Net architecture. The VGG-Net as shown in Fig. 1. has an architecture layout similar to the encoder of U-Net, therefore, the implementation U-Net chooses VGG-Net to replace the encoder path that hybrid between these two powerful architectures as shown in Fig. 2. The same action is applied for our proposed Y-net. The resultant architecture will be discussed later in the analysis and results

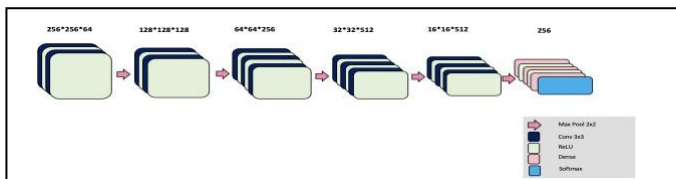


Fig. 1. The Architecture of VGG-16 Net.

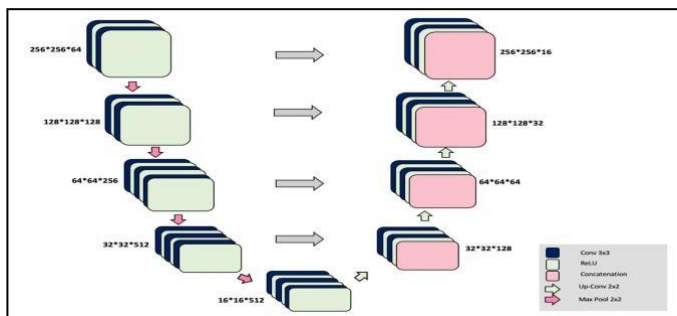


Fig. 2. The Architecture of U-Net with VGG-16.

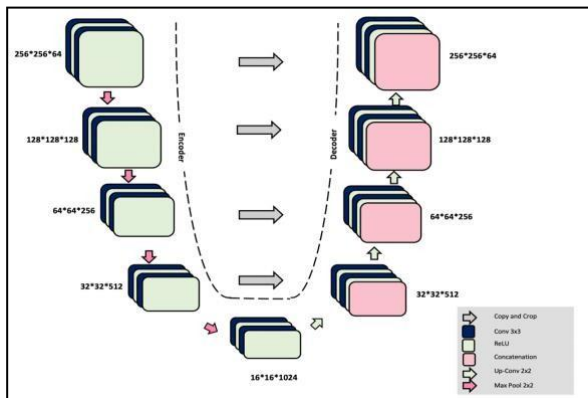
## 4 Research Methodology

### 4.1 Y-Net with Transfer Learning Architecture

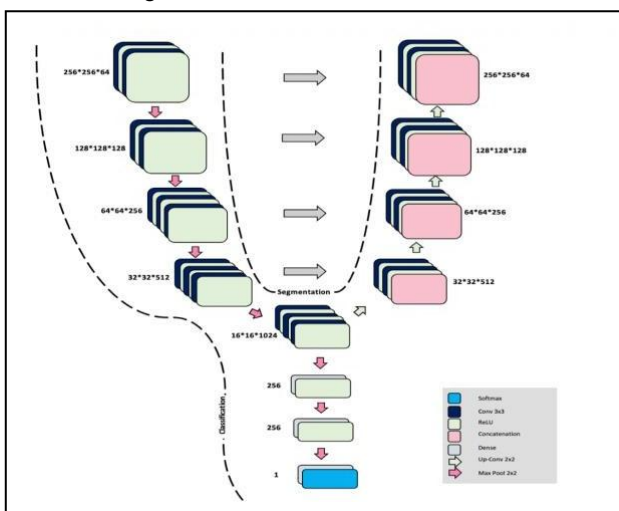
In this section, Y-NET architecture network is introduced for the purpose of detection and classification. As an architecture, the Y-net is constructed based to the idea of U-Net architecture. Both of them are suggested for biomedical image segmentation. To describe the architecture of Y-net, it is proper to introduce U-net architecture in brief, as shown in Fig. 3., the U-shape consists of an encoder layer (the left path) and the decoder layer (the right path). The encoder network output is the feature map/vector, such output features representing the input information. The decoder has a structure in the opposite way that uses the encoder feature to provide the intended output. The encoder reduces the input matrix size and increases the feature numbers, while the decoder returns the matrix size to its original size by minimizing the feature number results of the ground truth in every pixel. To incorporate the TL in U-net, U-VGG16 was used to replace the encoder path of U-Net as a hybrid between these structures [12]. The used VGG16 already has pre-trained weights by ImageNet that were applied to the U-VGG16 system resulting in the saving of training parameters and training time [14].

This research focuses on the structure of Y-Net and how to be applied for identification and segmentation. The Y-net architecture as given in Fig. 4 consists of two main functions, encoder-decoder for the function of segmentation and encoder- VGG16 fully connected layers for the function of identification. The Y-net encoder is done by using the full convolution of VGG16 while the decoder is done by concatenating the full convolution encoder layers into symmetrical decoder full transposed Conv layers. The full convolution layers of the decoder are an expansion style order of the encoder layer designed as transposed Conv ones. The identification function shares the same encoder of segmentation incorporating with the last three fully connected layers of VGG16 and an additional layer of dense Blocks of the same filter size. To make the feature vector of the identifier of binary form, the last conv layer is of  $1 \times 1$  sigmoid activation of the threshold value of 0.5.

The encoder is transferred-trained based on VGG-16 net using ImageNet one time for both functions. In other words, we replaced the encoder with a pre-trained network Visual Geometry Group, to reduce the number of parameters in the convolutional layers and thus improve the training time as well as to classify brain either has a tumor or not and then segment to calculate the area of tumor in one architecture which we called Y-Net. Due to a limited number of medical images, TL is suggested for cross-domain as pre-train using ImageNet. ImageNet is an expert in detecting of shapes, edges, and textures from natural images, then modify and fine-tune the model with US images[15]. Incorporating VGG16 represents the main parts of the proposed system results of Y-VGG16Net which is replaced by VGG19 for other applications results of Y-VGG19.



**Fig. 3.** U-Net Architecture for Segment a Tumor.



**Fig. 4.** Y-Net Architecture for both classification with VGG-16 and segmentation with U-Net and VGG-16.

### 4.2 Y-Net with Training and Testing

Based on the proposed system design requirement, it is implemented using Cloud Kaggle engine to save in memory and execution time for both training and testing. Due to the library or dataset availability on the cloud, there is no need to install anyone. Accelerators as CPU, GPU P100, GPU T4x2 or TPU v3-8 can easily be chosen, where GPU T4x2 is used. To support the available data sets, augmented data sets are created based on the original data sets, Kaggle data sets. To learn effectively, deep learning needs a large corpus of training data, but collecting this data can sometimes be very expensive and unrealistic. Using the original data, public Kaggle Dataset, and augmented one results of about 3929 containing 65.05% positive while remaining is negative. The dataset images help to train and test the suggested system for Brain Tumor Segmentation and identification effectively.

The Y-net architecture as given in Fig. 4. is trained and tested in this section for the task of segmentation and identification. To evaluate Y-net as constructed in section (4.1) above, we firstly run U-Net for the segmentation and CNN for classification processes, resulting in trainable parameters of about 31,031,745 and 30,197,249 respectively since no weights were

pre-trained before that consume a lot of time to train all parameter. All the process for segmentation and classification consume about 135 minutes. While pre-trained run U-Net for the segmentation and VGG16 for classification processes, resulting in trainable parameters of about 1,480,929 and 32,769 respectively. Pre-trained Y-net as proposed is run for one time for both segmentation and classification processes results in trainable parameters of about 1,480,929. All the process for segmentation and classification consume about 45 minutes. To reduce the number of parameters, the architecture of Fig. 4. is considered, where the pre-trained encoder is incorporated for both the segmentation and classification.

## 5 Performance Metric

To evaluate our model during the training and testing phases, we apply the most widely evaluation metrics Loss Function [15], Dice Coefficient DSC(A,B) [16, 17], and IoU [18] as follows:.

$$DSC(A,B) = (2*T.P)/(2*T.P + F.P+F.N)=(2*|A \cap B|)/(|A|+|B|) \quad (1)$$

where A , B denote predicted region and ground truth respectively and T.P denote True Positive while F.P False Positive and F.N False Negative

$$IoU(A,B) = (2*T.P)/(2*F.P + T.P+F.N)= (|A \cap B|)/(|A \cup B|) \quad (2)$$

where A , B denote predicted region and ground truth respectively and T.P denote True Positive while F.P False Positive and F.N False Negative

## 6 Y-Net Results

As described in section V, we stated the effect of using transfer learning for both of classification and segmentation processes. At first, we train the model to classify the image to have a tumor or not with CNN and then we attach a head of Transfer learning using VGG-16. As result total trainable parameter for CNN is about 30,197,249, see Table 1. while for pre-trained VGG-16 about 32,769, see Table 2. Furthermore, CNN has higher accuracy than VGG 16 but the time to execute the model of VGG -16 Is faster than CNN as shown in Table 3. Fig. 5. Training, accuracy, loss, validation Loss, and validation accuracy; (a) CNN (b) With VGG-16.

**Table 1.** Classification Model Summary for CNN Model.

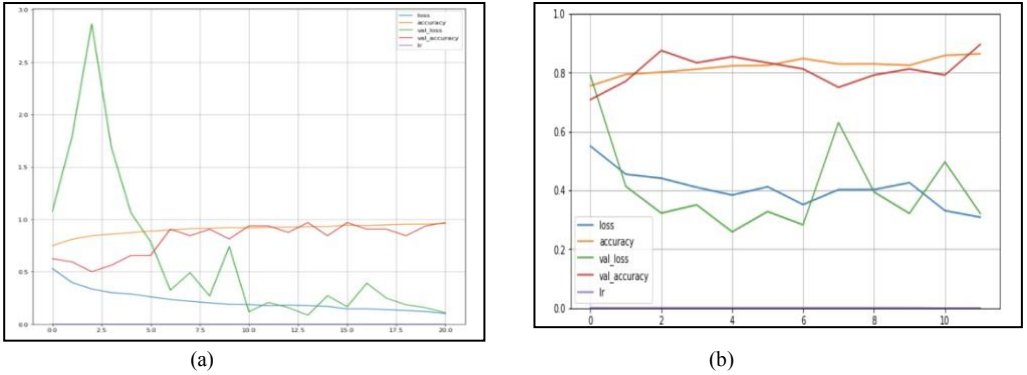
Layer	Type	Output Shape	Parameter
L1	2D Convolution	(254,254,128)	3584
L2	Batch Normalization	(254,254,128)	512
L3	MaxPooling	(84,84,128)	0
L4	2D Convolution	(82,82,256)	295168
L5	Batch Normalization	(82,82,256)	1024
L6	MaxPooling	(27,27,256)	0
L7	2D Convolution	(25,25,1024)	2360320
L8	Batch Normalization	(25,25,1024)	4096
L9	MaxPooling	(8,8,1024)	0
L10	2D Convolution	(6,6,2048)	18876416
L11	Batch Normalization	(6,6,2048)	8192
L12	MaxPooling	(2,2,2048)	0
L13	Flatten	(8192)	0
L14	Dense	(1024)	8389632
L15	Batch Normalization	(1024)	4096
L16	Dense	(256)	262400
L17	Batch Normalization	(256)	1024
L18	Dense	(1)	257
Trainable Parameters = 30,197,249			

**Table 2.** Classification Model Summary for Pre-trained VGG-16 Model using ImageNet

Layer	Type	Output Shape	Parameter
L1	Input Layer	(256,256,3)	0
L2	2D Convolution	(256,256,64)	1792
L3	2D Convolution	(256,256,64)	36928
L4	MaxPooling	(128,128,64)	0
L5	2D Convolution	(128,128,128)	73856
L6	2D Convolution	(128,128,128)	147584
L7	Max Pooling	(64,64,128)	0
L8	2D Convolution	(64,64,256)	295168
L9	2D Convolution	(64,64,256)	590080
L10	2D Convolution	(64,64,256)	590080
L11	MaxPooling	(32,32,256)	0
L12	2D Convolution	(32,32,512)	1180160
L13	2D Convolution	(32,32,512)	2359808
L14	2D Convolution	(32,32,512)	2359808
L15	MaxPooling	(16,16,512)	0
L16	2D Convolution	(16,16,512)	2359808
L17	2D Convolution	(16,16,512)	2359808
L18	2D Convolution	(16,16,512)	2359808
L19	MaxPooling	(8,8,512)	0
L20	Flatten	(32768)	0
L21	Dropout	(32768)	0
L22	Dense	(1)	32769
Trainable Parameter : 32,769			

**Table 3.** Performance and results of proposed approach

Classification	Loss	Accuracy	Time(m)
CNN	0.1021	0.9596	52
VGG 16	0.2152	0.9075	11



**Fig. 5.** Training, accuracy and loss (a) CNN (b) CNN with VGG-16

While in the segmentation process, we used first U-Net which has trainable parameter 31,031,745 as shown in Table 4 , and Y-net Segmentation Model Summary for VGG16 Pre-trained encoder Model using ImageNet with about 32,769 of the parameters to train as shown in Table 5. By comparing the Intersection over Union in the segmentation process, the Y-Net has higher than U-Net which is about 0.8255 as well as for another parameter, Dice Coefficients and Accuracy are higher than U-Net 0.9036, 0.9983 respectively, see Table 6. Time to execute the segmentation process for Y-Net is better than U- Net since the encoder is a pre-trained path and only training model for the decoder path, see Table 5.

As a result of classification and segmentation, Fig. 6. and Fig. 7. show respectively a sample of the pre-trained system output. Fig. 8. show training and validation of Loss, IoU, and Dice coefficients of U-Net and Y-Net respectively. After running the proposed model, the area of the tumor is calculated by extracting the white pixels of the mask, for the given sample is 12819 pixels, depending on the intended image as follows:

$$\text{Area of tumor} = \sqrt{(\text{no.of white pixle}) * 0.264} \quad (3)$$

The tumor is a contour surrounded by finding the location of tumor as X, and Y coordinates beside width W and height H.

$$(X, Y, W, H) = (133, 119, 60, 93).$$

**Table 4.** Classification Model Summary for CNN Model.

Layer	Type	Output shape	Parameter
L1	Input Layer	(256,256,3)	0
L2	2D Convolution	(256,256,64)	1792
L3	2D Convolution	(256,256,64)	36928
L4	Max Pooling 2D	(128,128,64)	0
L5	Convolution 2D	(128,128,128)	73856
L6	2D Convolution	(128,128,128)	147584
L7	2D Max Pooling	(64,64,128)	0
L8	2D Convolution	(64,64,256)	295168
L9	2D Convolution	(64,64,256)	590080
L10	2D Max Pooling	(32,32,256)	0
L11	2D Convolution	(32,32,512)	1180160
L12	2D Convolution	(32,32,512)	2359808
L13	2D Max Pooling	(16,16,512)	0
L14	2D Convolution	(16,16,1024)	4719616
L15	2D Convolution	(16,16,1024)	9438208
L16	2D UP Sampling	(32,32,1024)	0
L17	2D Convolution	(32,32,512)	2097664
L18	Concatenate	(32,32,1024)	0



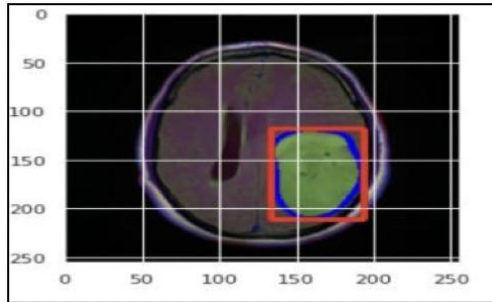
L19	2D Convolution	(32,32,512)	4719104
L20	2D Convolution	(32,32,512)	2359808
L21	2D UP Sampling	(64,64,512)	0
L22	2D Convolution	(64,64,256)	524544
L23	Concatenate	(64,64,512)	0
L24	2D Convolution	(64,64,256)	1179904
L25	2D Convolution	(64,64,256)	590080
L26	2D UP Sampling	(128,128,256)	0
L27	2D Convolution	(128,128,128)	131200
L28	Concatenate	(128,128,256)	0
L29	2D Convolution	(128,128,128)	295040
L30	2D Convolution	(128,128,128)	147584
L31	2D UP Sampling	(256,256,128)	0
L32	2D Convolution	(256,256,64)	32832
L33	Concatenate	(256,256,128)	0
L34	2D Convolution	(256,256,64)	73792
L35	2D Convolution	(256,256,64)	36928
L36	Convolution 2D	(256,256,1)	65
Trainable Parameter : 31,031,745			

**Table 5.** Y-Net Segmentation Model Summary for VGG-16 Pre-trained encoder model using ImageNet

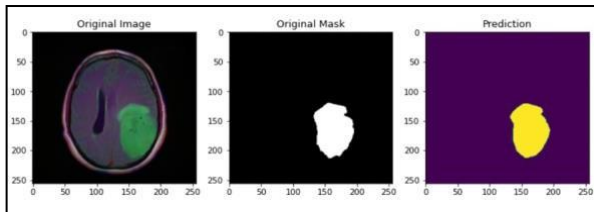
Layer	Type	Output Shape	Parameter	Trainable
L1	Input Layer	(256,256,3)	0	False
L2	2D Convolution	(256,256,64)	1792	False
L3	2D Convolution	(256,256,64)	36928	False
L4	Max Pooling	(128,128,64)	0	False
L5	2D Convolution	(128,128,128)	73856	False
L6	2D Convolution	(128,128,128)	147584	False
L7	Max Pooling	(64,64,128)	0	False
L8	2D Convolution	(64,64,256)	295168	False
L9	2D Convolution	(64,64,256)	590080	False
L10	2D Convolution	(64,64,256)	590080	False
L11	Max Pooling	(32,32,256)	0	False
L12	2D Convolution	(32,32,512)	1180160	False
L13	2D Convolution	(32,32,512)	2359808	False
L14	2D Convolution	(32,32,512)	2359808	False
L15	Max Pooling	(16,16,512)	0	False
L16	2D Convolution	(16,16,512)	2359808	False
L17	2D Convolution	(16,16,512)	2359808	False
L18	2D Convolution	(16,16,512)	2359808	False
L19	2D ConvolutionTranspose	(32,32,128)	262272	True
L20	Concatenate	(32,32,640)	0	True
L21	2D Convolution	(32,32,128)	737408	True
L22	2D Convolution	(32,32,128)	147584	True
L23	2D ConvolutionTranspose	(64,64,64)	32832	True
L24	Concatenate	(64,64,320)	0	True
L25	2D Convolution	(64,64,64)	184384	True
L26	2D Convolution	(64,64,64)	36928	True
L27	2D ConvolutionTranspose	(128,128,32)	8224	True
L28	Concatenate	(128,128,160)	0	True
L29	2D Convolution	(128,128,32)	46112	True
L30	2D Convolution	(128,128,32)	9248	True
L31	2D ConvolutionTranspose	(256,256,16)	2064	True
L32	Concatenate	(256,256,80)	0	True
L33	2D Convolution	(256,256,16)	11536	True
L34	2D Convolution	(256,256,16)	2320	True
L35	2D Convolution	(256,256,1)	17	True
Trainable Parameter : 1,480,929				

**Table 6.** Performance and results of proposed approach

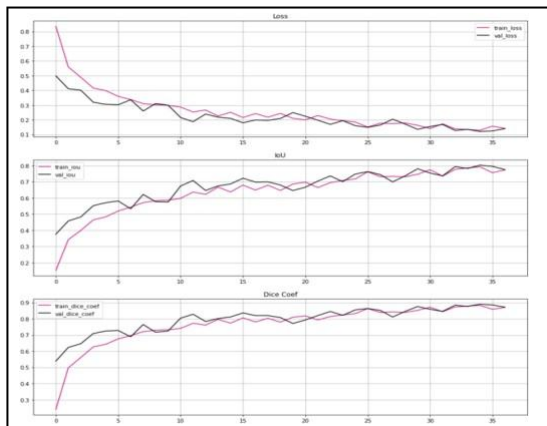
Segmentation	Loss	IoU	Dice	Accuracy	Time(m)
U-Net	0.1417	0.7731	0.8704	0.9976	128
V-Net	0.1044	0.8255	0.9036	0.9983	34



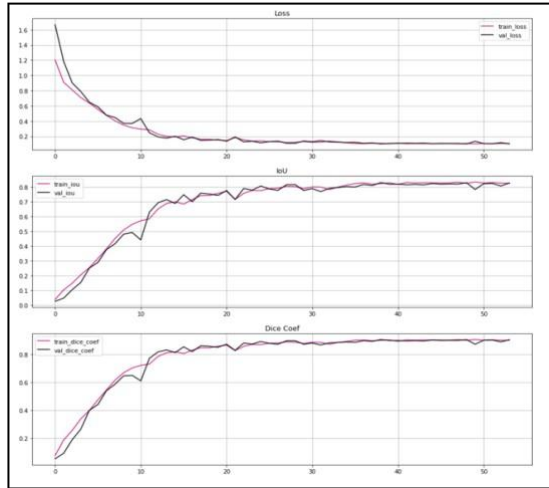
**Fig. 6.** Contour Surrounding Tumor



**Fig. 7.** Segmentation of Tumor with original mask and predict



(a)



(b)

**Fig. 8.** Training and Validation Loss, IoU and Dice Coefficients (a) U-Net (b) V-Net

## 7 Conclusion

This study presented the framework Y-net which it is equivalent to the Y-Net with the implementation of TL for both automatic classification and segmentation of MRI images. The pre-trained encoder is implemented for one time for both classification and segmentation. The visual predictions result shows significant refinement in edge smoothing and shaping. Comparison of Y-net with other different models shows the quality in results and saving in training time. The run with GPU makes the MRI classification and segmentation more reliable and runs well on the computer with a processor of Intel Core i7, 32GB RAM, VRAM and.128GB SSD. The ROI segmentation with the proposed model is very well tend to the task of brain tumor MRI image. The system is tested for many 2D patches images of different centers for extraction and training. So far, the extension of the proposed system is to include the creation of models to be trained on 3D patches for the function of segmentation and identification.

## References

1. V. E. Balas, B. K. Mishra, and R. Kumar, Eds., Handbook of deep learning in biomedical engineering: techniques and applications. London, United Kingdom ; San Diego, CA, United States: Academic Press, 2021.
2. X. Gu, Z. Shen, J. Xue, Y. Fan, and T. Ni, "Brain Tumor MR Image Classification Using Convolutional Dictionary Learning With Local Constraint," *Front. Neurosci.*, vol. 15, p. 679847, May 2021, doi: 10.3389/fnins.2021.679847.
3. [A. Younis, L. Qiang, C. O. Nyatega, M. J. Adamu, and H. B. Kawuwa, "Brain Tumor Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches," *Applied Sciences*, vol. 12, no. 14, p. 7282, Jul. 2022, doi: 10.3390/app12147282.
4. X.-X. Yin, L. Sun, Y. Fu, R. Lu, and Y. Zhang, "U-Net-Based Medical Image Segmentation," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–16, Apr. 2022, doi:10.1155/2022/4189781.

5. Y. LeCun, Y. Bengio et al. “Convolutional networks for images, speech, and time series”, *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
6. Y. Cao, W. Zhou, M. Zang, D. An, Y. Feng, and B. Yu, “MBANet: A 3D convolutional neural network with multi-branch attention for brain tumor segmentation from MRI images,” *Biomedical Signal Processing and Control*, vol. 80, p. 104296, Feb. 2023, doi: 10.1016/j.bspc.2022.104296.
7. K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. Lee, B. Kainz, D. Rueckert et al. “Ensembles of Multiple Models and Architectures for Robust Brain Tumor Segmentation” *MICCAI Brain Lesion Workshop*, 2017.
8. A. Myronenko, “3D MRI brain tumor segmentation using autoencoder regularization,” 2018, doi: 10.48550/ARXIV.1810.11654.
9. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” 2015, doi: 10.48550/ARXIV.1505.04597.
10. S. Maqsood, R. Damaševičius, and R. Maskeliūnas, “Multi-Modal Brain Tumor Detection Using Deep Neural Network and Multiclass SVM,” *Medicina*, vol. 58, no. 8, p. 1090, Aug. 2022, doi: 10.3390/medicina58081090.
11. U. Kose and J. Alzubi, Eds., “Deep Learning for Cancer Diagnosis”, vol. 908. Singapore: Springer Singapore, pp. 239-248 ,2021. doi: 10.1007/978-981-15-6321-8.
12. [A. A. Pravitasari et al., “UNet-VGG16 with transfer learning for MRI- based brain tumor segmentation,” *TELKOMNIKA*, vol. 18, no. 3, p. 1310, Jun. 2020, doi: 10.12928/telkomnika.v18i3.14753.
13. P. Mlynarski, H. Delingette, A. Criminisi, and N. Ayache, “3D convolutional neural networks for tumor segmentation using long-range 2D context,” *Computerized Medical Imaging and Graphics*, vol. 73, pp. 60–72, Apr. 2019, doi: 10.1016/j.compmedimag.2019.02.001.
14. A. Nawaz, U. Akram, A. A. Salam, A. R. Ali, A. Ur Rehman, and J. Zeb, “VGG-UNET for Brain Tumor Segmentation and Ensemble Model for Survival Prediction,” in *2021 International Conference on Robotics and Automation in Industry (ICRAI)*, Rawalpindi, Pakistan, Oct. 2021, pp. 1–6. doi: 10.1109/ICRAI54018.2021.9651367.
15. D. Cheng and E. Y. Lam, “Transfer Learning U-Net Deep Learning for Lung Ultrasound Segmentation,” 2021,doi: 10.48550/ARXIV.2110.02196.
16. P. Shi, M. Duan, L. Yang, W. Feng, L. Ding, and L. Jiang, “An Improved U-Net Image Segmentation Method and Its Application for Metallic Grain Size Statistics,” *Materials*, vol. 15, no. 13, p. 4417, Jun. 2022, doi:10.3390/ma15134417
17. K. H. Zou et al., “Statistical validation of image segmentation quality based on a spatial overlap index1,” *Academic Radiology*, vol. 11, no. 2, pp. 178–189, Feb. 2004, doi: 10.1016/S1076-6332(03)00671-8
18. M. A. Rahman and Y. Wang, “Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation,” in *Advances in Visual Computing*, vol. 10072, G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, Eds. Cham: Springer International Publishing, 2016, pp. 234–244. doi:10.1007/978-3-319-50835-1\_22.