

OBJECTIVE

The objective of this synopsis is to outline the development of an automated model for brain tumor segmentation from MRI scans, with a specific focus on addressing the increasing prevalence of brain tumors in India. The scarcity of trained radiologists and neurosurgeons in rural and underprivileged areas has led to a critical need for automated solutions to assist in early tumor detection and segmentation. The primary aim is to leverage deep learning techniques, particularly the modified U-Net architecture, to automate the segmentation process and eliminate the requirement for specialized medical professionals to identify tumorous areas from MRI scans. The dataset given by Mateusz Buda, Ashirbani Saha, Maciej A. Mazurowski used in "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm." Computers in Biology and Medicine, 2019 is utilized for training and validation, providing a robust foundation for the model's development and evaluation.

The experiments conducted using the Google Colab platform demonstrate the model's efficacy in accurately segmenting brain tumors from 3D MRIs. The ultimate vision is to create a user-friendly platform, such as a website or application, where individuals worldwide can easily upload their 3D MRI scans for automated tumor segmentation, thereby enhancing accessibility to tumor detection and diagnosis, particularly in regions with limited medical resources. This initiative aims to contribute to the advancement of medical imaging technology and improve patient outcomes, especially in underserved communities.

INDEX

- 1) Introduction
- 2) Methods & Datasets
 - i. Basics of Deep Learning & Convolutional Neural networks
 - ii. Modified U-Net
 - iii. Dataset & Evaluation
- 3) Experiments & Results
- 4) Conclusion and Future Scope
- 5) References

INTRODUCTION

Motivation

In India, the prevalence of brain-related tumors is 5-10 per 100,000 people and brain tumors in children's are the second most frequent malignancy which accounts for around 26 percent of all cancers. According to the International Association of Cancer Registries (IARC), in India, there are over 28,000 cases of brain tumors per year, and over 24,000 people pass away from them[1]. Due to the scarcity of trained and well qualified radiologists & neurosurgeons in backward and rural areas, villages and even in some towns these cases are increasing day by day as the tumors can be controlled in the initial stages so we wanted to develop such a model that would automatically segment the tumorous areas from the MRI scans.

Related Works

This year on the Brats-2018 dataset, the top performing submissions of research related to brain tumor segmentation included Assef Raad Hmeed [2] who proposed the semantic segmentation using U-Net model which is a Fully Convolutional Network (FCNN) and these results were promised by segmenting the brain tumor within various sub-areas.

Another similar work related to brain tumor segmentation in 2022 has been carried out by Abdullah Al Nasim, MD [3] who proposed the segmentation model using U-Net using CNN, he improved the 2D U-Net network and trained with the BraTS dataset to find the 4 areas which he was looking for i.e. the overlaps of necrotic, edematous, growing and healthy tissue for performing segmentation.

Another work involving brain tumor segmentation this year carried out by Ping Zheng, Xunfei Zhu, Wenbo Guo [4]. An improved U-Net network was proposed to segment brain tumor to improve segmentation effect of brain tumor. It was achieved by using improved segmentation performance by adding hybrid dilated convolution (HDC).

Our Work

Our work is to completely automate the process of brain tumor segmentation, hence there will not be any requirement of a trained radiologist to segment the tumorous areas from the MRI scans. In our model we are performing the semantic segmentation of Magnetic Resonance Images (MRI) of brain to identify the areas of the brain having tumor using a modified U-Net architecture, it is a great architecture having great accuracy for image segmentation tasks. Dataset used for training and validation purpose of our model is dataset is provided by Kaggle used in "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm." Computers in Biology and Medicine, 2019.

METHODS & DATASETS

We are performing brain tumor segmentation task using a modified U-Net architecture along with a VGG-16 Convolutional Neural Network (CNN) which is based on deep learning model.

CNN based Deep Learning Model

A Machine Learning technique that learns features and tasks directly from data by running inputs through a biologically inspired Neural Network architecture is called as Deep Learning.

Figure 1 states that Deep learning is a subset of Machine Learning which is in turn a subset of Artificial Intelligence.

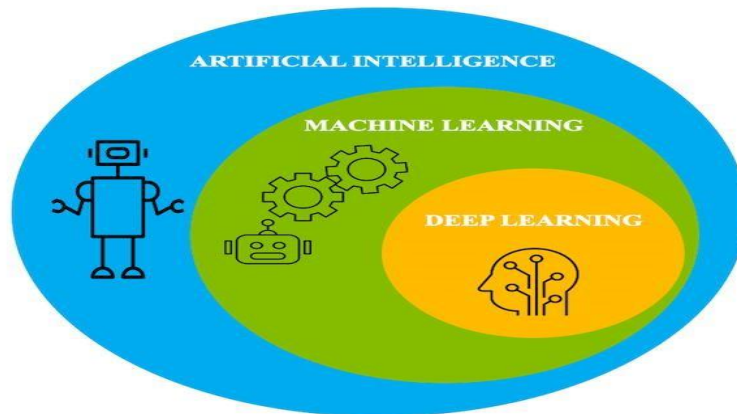


Fig. 1 Deep learning as a subset of machine learning which is a part of Artificial Intelligence

Why deep learning?

This is because ML is always machine like, they need a lot of domain expertise, human intervention and are of specific tasks. In case of deep learning, features can be learnt just from the raw data whereas in case of machine learning we are supposed to tell the machine about each and every information.

Deep Learning nowadays has become more prominent because of some reasons such as presence of vast amount of data, improved hardware architecture and new software architectures.

Neural Network:

It forms the basis of deep learning, here the algorithms are inspired by the neurons of human brain, just like neurons build up the brain, the fundamental element which builds up the Neural Network here is also called as neuron. Neural Network takes in data, train themselves to understand patterns in data and provide useful predictions

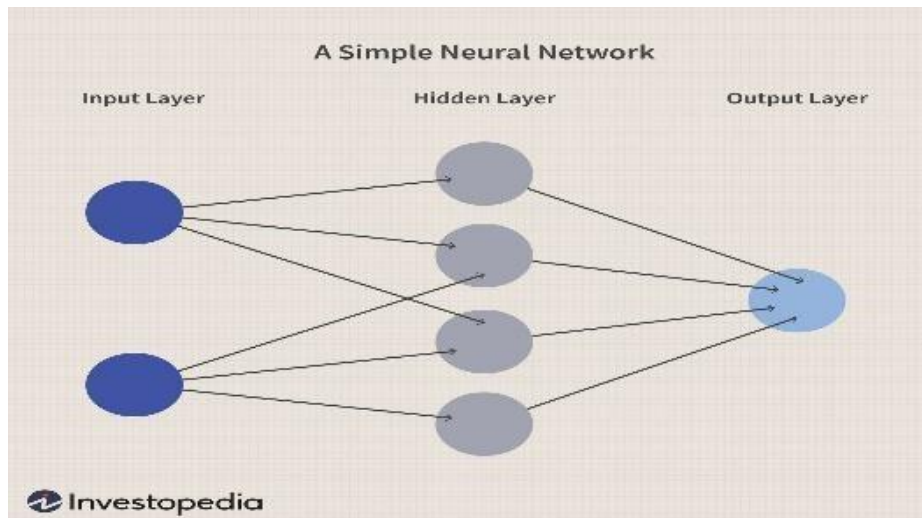


Fig. 2 A simple Neural Network

Important terms related to Neural Network:

- **Activation function:**

Introduces non-linearity in the network and decide whether a neuron can contribute to the next layer. Types of activation function used:

- Step function
- Linear function
- Sigmoid function
- Tan h function

- **Loss functions:**

They enable us to quantify the error, there are a plenty of loss functions used for different purposes.

- Binary Cross Entropy
- Mean Squared error
- Hubber Loss
- BCEDiceLoss

- **Optimisers:**

During training, we adjust the parameters to minimize the loss function and make our model as optimized as possible. They do this by tying together the loss function and model parameters by updating the network based on the response of the loss function. In other words, it moulds our model by adjusting the weights and biases to more accurate ones.

Loss functions guides the optimisers, it tells the optimiser whether its moving in right/wrong direction.

Some useful optimisers are mentioned below:

- Gradient Descent algorithm
- Stochastic gradient descent (SGD)
- Adam
- RMSprop

- **Regularization:**

The core problem in deep learning is that the model should perform well on training data as well as on new test data inputs and then our deep learning model will be said to be completed.

One of the most common problems encountered is Overfitting. (In this case we don't get accurate readings for new test data).

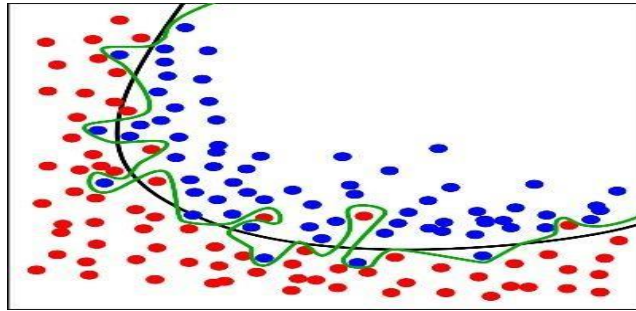


Fig. 3 Overfitting (Green line in the image represents overfitting)

There are various methods to tackle overfitting:

- **Dropout:** Let's say we have a Neural Network with 2 hidden layers, what dropout does is it randomly removes some nodes and their connections from network, as it captures the randomness and memorises less of training data and hence generalise better and build a more robust predictable model.

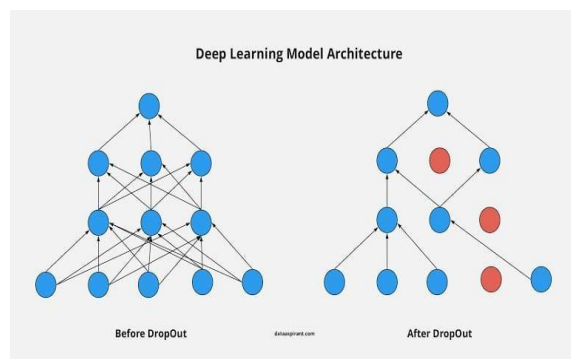


Fig. 4 Dropout

- **Early Stopping:** Training error decreases steadily, but validation error increases after a certain point. In other words Stopping at the point where the error in the validation starts to increase is known as Early stopping.

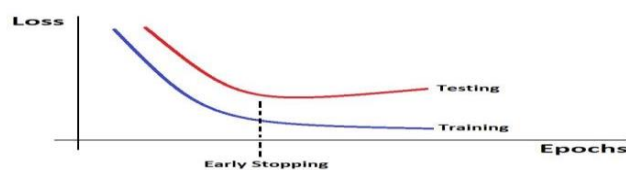


Fig. 5 Early Stopping

We need some terms when our dataset is too large, as we can't pass the data to compare at once, hence we divide them in smaller chunks and feed these chunks to Neural Network one by one.

- **Epochs:** When the entire dataset is passed forward and backward through the neural network only once, it is called as an epoch. We use multiple epochs to help our model generalise, there are no right no. of epochs.
- **Batch & Batch size:** We divide large datasets into smaller batches and feed those batches into the neural network. Total no. of training examples in a batch is called the batch size.

Neural Network are basically of 3 types:

- Fully connected feed forward neural network
- Recurrent Neural network
- **Convolution Neural network:** It is a type of neural network designed for specific tasks like image classification. It is inspired by the organisation of Neurons in the visual cortex of the human brain which is good for processing data like images, audio and video.

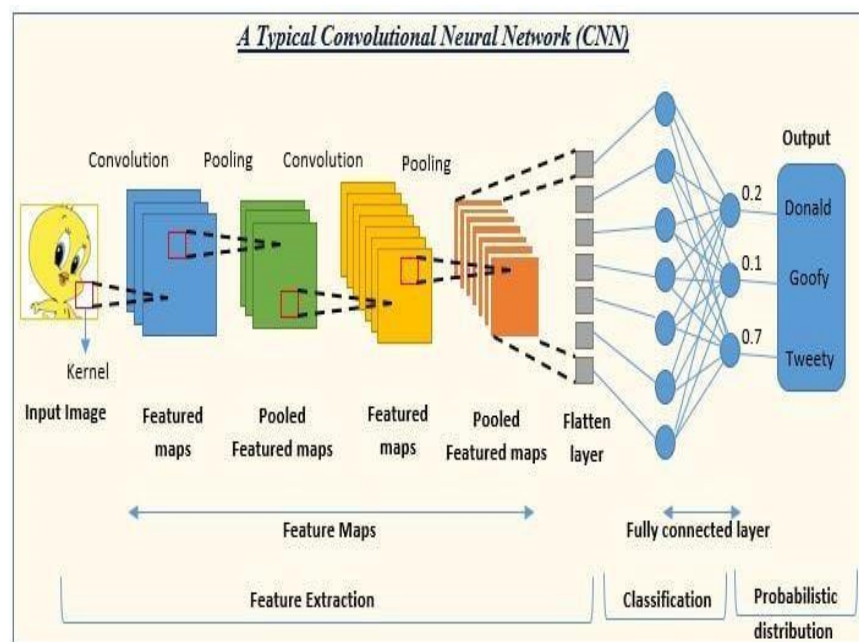


Fig. 6 Convolutional Neural Network

Hidden layers of CNN consist of:

1. Convolutional layers
2. Pooling layers
3. Fully connected layers
4. Normalisation layers

Convolution allows us to extract visual features from a 2D array in small chunks, each neuron in the convolution layer is responsible for small cluster of neurons in the preceding layer.

Pooling (also known as sub-sampling), its objective is to further reduce the no. of neurons necessary in subsequent layers of network, while still retaining the important information. These are of 2 types:

Max pooling: Based on picking up the maximum value from selected region.

Min pooling: Based on picking up the minimum value.

VGG-16: VGG-16 is a convolutional neural network that is 16 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature[4]. It is known for its simplicity and good performance on a variety of image classification tasks. It has been widely used as a baseline model in many Image classification benchmarks and has inspired the development of many other CNN models.

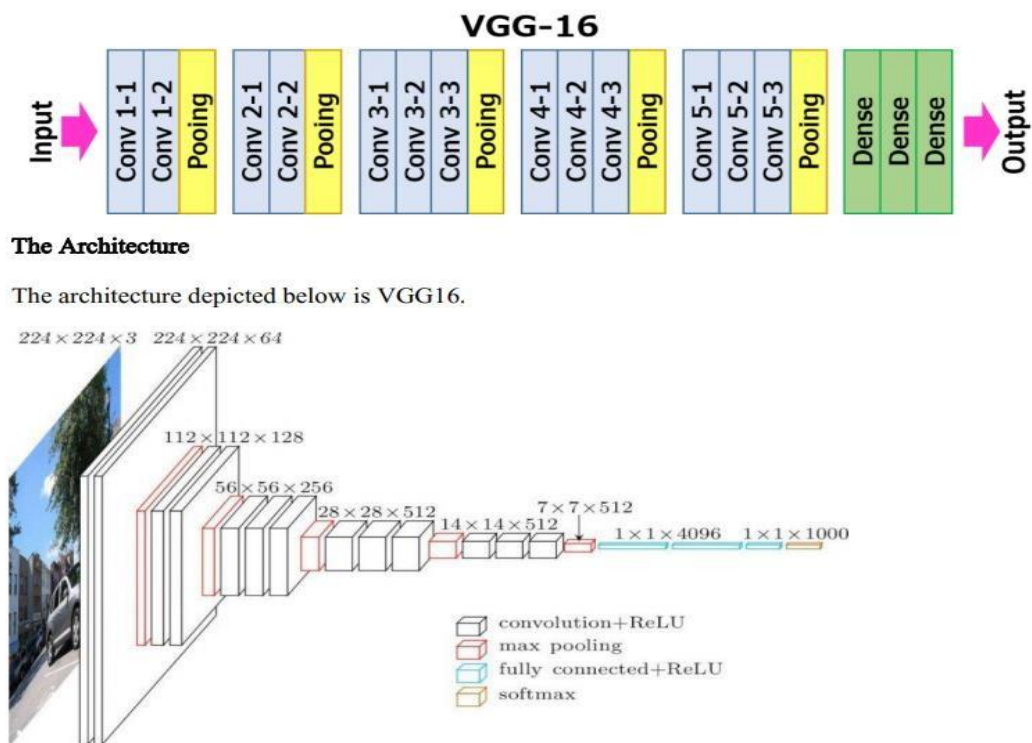


Fig. 7 VGG16, A CNN with 16 layers

In the above section we have stated about what is deep learning and convolutional neural network which we will be using as the backbone of our modified U-Net. Now let's understand about our main architecture, the U-Net architecture.

Modified U-Net Architecture: (U-Net with VGG16 as its backbone)

U-Net is a deep learning architecture for semantic segmentation, it is used for bio-medical imaging. It is a great architecture with a great accuracy and hence we started using it for other image segmentation tasks.

Segmentation: It involves dividing a visual input into segments to make image analysis easier. Segments are made up of sets of one or more pixels. Image segmentation sorts pixels into larger components while also eliminating the need to consider each pixel as a unit.

Image Segmentation is done in 2 ways:

1. Semantic Segmentation: In case of semantic segmentation, we don't need to care about how many people/objects do we have in that image, only classes are segmented.
2. Instance Segmentation: In this case it specifies how many objects/persons we have i.e. we need to specify that it is the object 1 from class 'X', object 2 from class 'Y'.



https://www.researchgate.net/figure/Semantic-segmentation-left-and-instance-segmentation-right-8_fig1_339328277

3

Fig. 8 Semantic and Instance Segmentation

U-Net is an architecture to perform semantic segmentation and that's all what we need in our project as we have to perform brain tumour segmentation where we don't need to know about its different parts, just 2 classes have to be specified one the brain tumour and other the background.

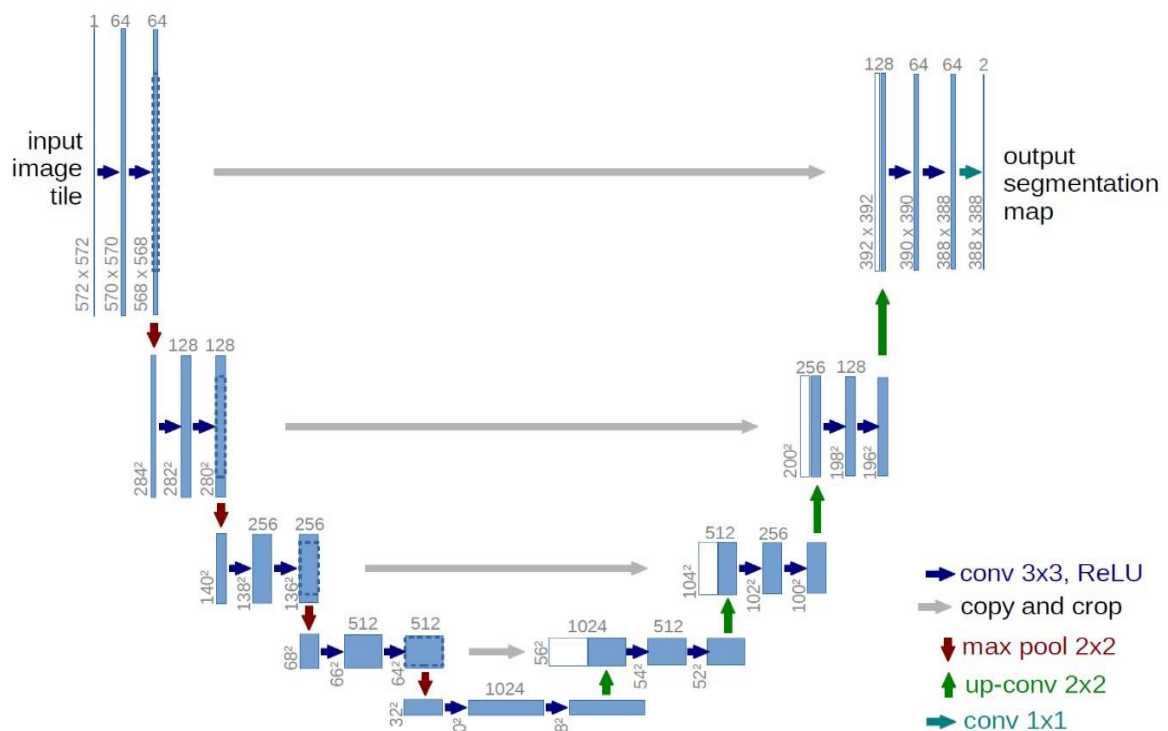


Fig. 9 U-Net architecture with visual depiction of convolution layers

Its architecture is in the shape of U, because of which it is called as U-net, one of the easiest architectures of deep learning. There are only convolution layers, followed by Max pooling again and again and then up-convolution followed by max pooling and involves a relu activation function after each convolution layer.

We are doing this because U-net is divided into 2 parts, one is called as **Encoder**, if we look only in this part it can be assumed as a Normal image classification structure. After each output of this block we have a relu activation function used here, at the end of encoder part we get a final output which can be passed through a SoftMax layer to see the output but this is not what we need here, the output needed is a mask with the same width & height as of an input image so that we can specify each pixel and its class, for this we need to go through another half of U-net i.e. the **Decoder**, this decoder will take the feature that we have extracted from the encoder part and use them here to build the output image so that we can have the probability of each pixel of an image, instead of probability for all the image.

The various specifications used in our model are stated below:

Activation function:

ReLU (Rectified Linear unit): We are using ReLU as the Activation function in our model, though it seems to be linear, but it is infact non-linear. This means we can stack layers but since it is bounded from 0 to infinity, it has a chance of blowing up the activation function. Ideally, we want only few neurons to participate in output processing, hence this is where ReLU comes into picture, it provides us with a network with randomly initialised weights and almost 50% of the network yield zero activation & hence it becomes lighter and faster.

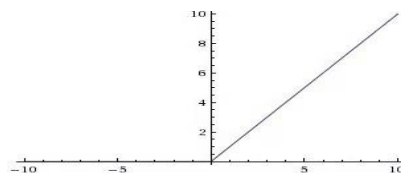


Fig. 10 Rectified Linear unit activation function

Loss function:

BCE Dice loss: It is used to calculate the similarity between two images. We proposed a loss function which is a combination of BCE loss, Focal loss and Dice loss, where BCE is used for pixel-wise classification, Focal loss is used for learning hard examples and Dice loss is used for learning better boundary representation.

$$J(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N H(p_n, q_n) = -\frac{1}{N} \sum_{n=1}^N \left[y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right]$$

Optimiser:

Adam: It stands for Adaptive Moment Estimation which uses the concept of momentum i.e., it's basically our way of telling the Neural Network whether we want the past changes to affect future changes by adding fraction of past gradient to current one.

We use Adam optimiser with initial learning rate of $\alpha_0 = 1e-3$.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[\frac{\delta L}{\delta w_t} \right] \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[\frac{\delta L}{\delta w_t} \right]^2$$

The above formula represents the working of Adam optimizer. Here β_1 and β_2 represent the decay rate of the average of the gradients.

Dataset (LGG Segmentation Dataset):

LGG Segmentation Dataset [5] is a dataset which provides multimodal 3D brain MRIS and ground truth brain tumor segmentations annotated by physicians, consisting of 2 MRI modalities per case (MRI Images and FLAIR). This dataset contains brain MR images together with manual FLAIR abnormality segmentation masks created by hand are part of this dataset. Annotations include 3 tumor subregions- the enhancing tumor, the peritumoral edema, and the necrotic and non-enhancing tumor core. The annotations were combined into 3 nested subregions-whole tumor (WT), tumor core (TC), and enhancing tumor (ET). The data were collected from 19 institutions, using various MRI scanners [6].

It is a collection of MRI scans of brain tumors with annotations identifying their existence and location make up the LGG Segmentation dataset. These pictures came from TCIA, or The Cancer Imaging Archive. The dataset is extensively utilised in the field of medical image analysis for testing automatic brain tumor segmentation methods. A training set, a validation set, and a test set make up the LGG Segmentation dataset. They correspond to 110 patients included in The Cancer Genome Atlas (TCGA) lower-grade glioma collection with at least fluid-attenuated inversion recovery (FLAIR) sequence and genomic cluster data available. The test set's scans are kept private and are only used to assess the segmentation algorithms that were submitted.

Evaluation:

The Evaluation metrics which we are using to evaluate our segmentation model are:

1.) Dice Coefficient (BCE Dice loss):

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Here we simply put the dice coefficient to be **2 * the Area of Overlap divided by the total number of pixels in both images** as stated above and shown in the below mentioned figure [8].

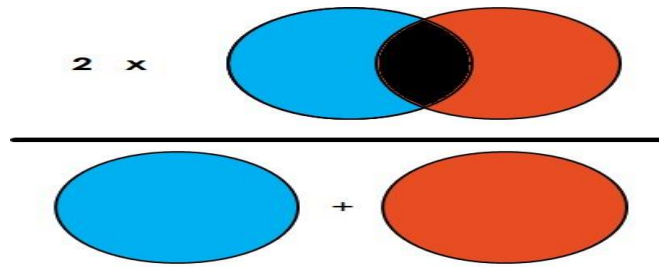


Fig. 11 Venn diagram depicting the calculation of dice coefficient

2.) Intersection over Union (IoU, Jaccard Index):

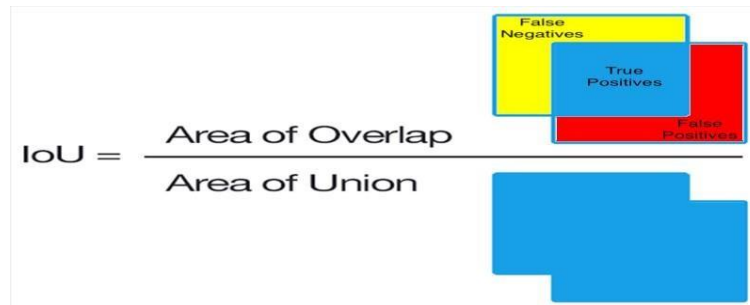


Fig. 12 Formulae to calculate the IoU Score

One of the most popular metrics in semantic segmentation is the Jaccard index, often known as the intersection over union. It is the simplest and most powerful metric available.

IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth[8].

This metric ranges from 0-1(0-100%) with 0 signifying no overlap and 1 signifying perfect overlapping segmentation. The IoU and the Dice coefficient are quite similar. They are positively associated, thus if one claims that model A is superior to model B at picture segmentation, the other will also claim the same. They both have a range from 0 to 1, with 1 denoting the highest resemblance between expected and truth, similar to the IoU.

3.) Pixel Accuracy:

It is perhaps the easiest to understand conceptually. It is defined as the percent of pixels in your image that are classified correctly. While it is easy to understand, it is in no way the best metric.

It can also be defined as a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the number of correct predictions to the total number of predictions [8].

EXPERIMENTS & RESULTS

System Details:

System we used to train our model is "Google Colab," a product of Google Research, that is employed for the training and testing of our model. It is particularly well suited to machine learning, data analysis, and education and enables anyone to write and run arbitrary Python code through the browser. It is a hosted jupyter notebook service that do not requires any kind of setup, while providing access free of charge to computing resources including GPUs [9].

Google colab offers 2 different GPU models since 2019: K80 and T4. However, one cannot select which of them they want due to availability issues. Maximum execution time and maximum idle time provided in google colab is 12 hours and 90 minutes respectively.

The code that we execute on google colab is executed in a virtual machine environment, and doesn't use the local machines hardware or software. It doesn't use the host OS (Windows, here). Instead, it uses a Linux environment, which can be verified by running the following snippet in a code cell.

GPU setting must be turned on from the change runtime type option in the runtime settings drop down menu. This is because the speed of our Image segmentation model is increased 10 times more for parallel implementation of algorithm as compared to sequential.

The experimental result shows that the use of GPU **improved performance efficiency of level set method-based segmentation when run in parallel.**

We implemented our network or model in Pytorch and trained it on Google colab along with the runtime type to be set on GPU mode using the LGG Segmentation training dataset without any additional data. As stated earlier this dataset consists of 2 different MRI Modalities, but we have selectively trained our model on 'Flair' and MR images Modality which is said to be the most efficient one.

Results:

Now the dataset LGG Segmentation Dataset was trained in our image segmentation model using modified UNet. The results have been calculated using the 2 most effective evaluation metrics i.e., Dice coefficient and IoU score and additional metrics that is Binary Accuracy. The table given below shows different values of given evaluation metrics based on no. of epochs used:

S.No.	Number of epochs	Dice Score	IoU Score
1.	100	0.8183	0.5043
2.	200	0.7847	0.5387
3.	300	0.7436	0.7098

Output of 3D MRI brain tumor segmentation using modified U-Net architecture:

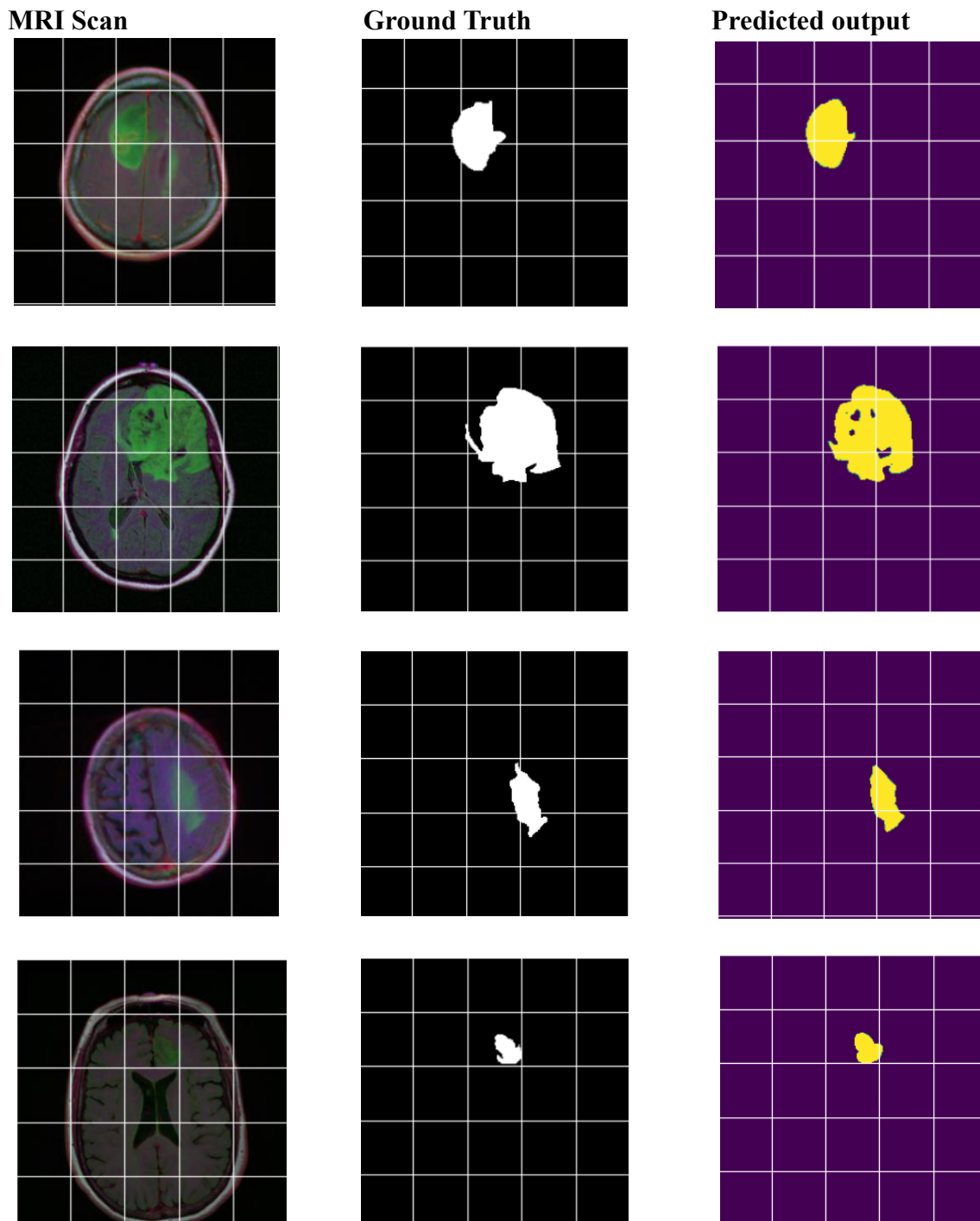


Fig. 13 Visual comparison of images with its ground truth and predicted output

CONCLUSION & FUTURE SCOPE

In this work, we have described a semantic image segmentation model using modified U-Net for brain tumor segmentation from 3D MRIs obtained from the LGG segmentation dataset. We have mainly focused on automating the whole brain tumor detection process which was earlier carried out by only specialised Neurosurgeons.

But now with the help of our model any patient having a 3D MRI scan of his/her brain can simply upload the scan in our model and within few minutes will get the detailed segmented output of whether brain tumor is present or not and if its there our model will give the segmented output of the areas having tumor as shown in the result.

Our LGG Segmentation testing dataset results for 300 epochs found to be the most accurate and precise among all are **0.7436** as Dice coefficient and **0.7098** as IoU score or Jaccard Index.

In the last layer of U-Net we have used sigmoid as the activation function which produces only 2 classes in segmentation mask which is why we are getting only single class segmentation output, in future we intend to use other activation function like softmax in the place of sigmoid function to produce multi-class segmentation output.

Stating about the future scope of our project, we are planning of creating a website or an application/user interface that will be user friendly as any patient in any part of the world can access the webpage and upload his/her 3D MRI scan of brain and they'll get the segmented output of whether the tumor is present or not.

REFERENCES

- [1]. Brain tumor statistics:
<https://health.economictimes.indiatimes.com/news/industry/worldbrain-tumour-day-2022-together-we-are-stronger/92078118>
- [2]. Assef Raad Hmeed, Salah A. Aliesawi and Wesam M. Jasim: Deep semantic segmentation for MRI brain tumor, AIP Conference Proceedings 2400, 020023 (2022):
<https://aip.scitation.org/doi/abs/10.1063/5.0112348>
- [3]. Abdullah Al Nasim, MD; Al Munem, Abdullah; Islam, Maksuda; Aminul Haque Palash, Md; Mahim Anjum Haque, MD.; Shah, Faisal Muhammad: Brain Tumor Segmentation using Enhanced U-Net Model with Empirical Analysis, International Conference on Computer and Information Technology (ICCIT) 2022.
- [4]. Ping Zheng, Xunfei Zhu, Wenbu Guo; Brain Tumor Segmentation based on an improved U-Net: <https://bmcm imaging.biomedcentral.com/articles/10.1186/s12880-022-00931-1>
- [5]. LGG Segmentation Dataset download: <https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation>
- [6]. About LGG Segmentation dataset: <https://huggingface.co/datasets/gymprathap/Brain-MRI-LGG-Segmentation>
- [7]. More about dataset: <https://chat.openai.com/chat>
- [8]. Evaluation metrics: <https://towardsdatascience.com/metrics-to-evaluate-your-semanticsegmentation-model-6bcb99639aa2>
- [9]. GoogleColaboratory: <https://research.google.com/colaboratory/faq.html#:~:text=The%20Basics&text=Colaboratory%2C%20or%20%E2%80%9CColab%E2%80%9D%20for,learning%2C%20data%20analysis%20and%20education.>