Deep Learning for Automatic Detection and Segmentation of Brain Tumors from MRI Images

A report of Course Project on Image and Video Processing (EC348)
Submitted by

Deeksha M S	171EC113
Niranjan Ramesh Rao	171EC130
Varun Kumar M	171EC251

Under the guidance of
Dr. Shyamlal

in partial fulfilment of the requirements for the award of the degree of
BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

Abstract

Computer-aided tumour detection and segmentation has majorly overcome the uncertainty involved in human diagnosis. Artficial intelligence, particularly deep learning, has given accurate models for real-time tumor classification and locating them. In this study we compare state-of-art architectures: VGGnet, Alexnet, Googlenet, Resnet with the new BrainMRnet model with channel and spatial attention for tumor detection. This added attention helps locate the brain more accurately, while improved data augmentation and regularisation boosts the performance. We also employ PSPnet, Encoder-Decoder, Deeplab and Mobile-UNet with regularised weighted-class loss function for tumour segmenation. The best results are shown by BrainMRnet with 97.2% accuracy, 97.1% specificity and 96.8% sensitivity for detection and Mobile-UNet with a dice score of 96% and 83% IoU for segmentation.

Contents

1 Introduction

Brain tumor is one of the leading causes of cancer-related death globally among children and adults[14]. Precise detection of brain tumor at an early stage requires expert physicians plays a key role in successful prognosis and treatment planning.

Patient diagnosis relies on a doctor's manual evaluation of a patient and his or her test results. With no automated tools to help with a doctor's diagnosis and limited number of available doctors, not only is there a higher risk of misdiagnosis but also an increase in wait time for patients to be seen. Doctors must take the time to manually review test results, examine multiple image slices to determine health issues which takes time away from more complex diagnoses and spending time with the patient. In order to improve patient care, enhanced medical technology in the form of automated tools is necessary to increase doctor efficiency and decrease patient time in hospitals and time toward recovery.

There is a paramount need for development of automated methods to aid doctors in diagnosis in order to prevent misdiagnosis and decrease patient wait time. In particular, automation of detection and precise segmentation of brain tumors could drastically improve brain tumor survival rate.

While extensive research has successfully applied these techniques to recognizing patterns in non-medical images, the proposed research applies SOTA deep learning algorithms[13] to medical images which there is a lack of available datasets. Furthermore, applying neural networks to medical images has implications of faster and more precise diagnoses. By including images of brains without tumors, neural networks can better learn the structure of a brain and take steps towards differentiating brains with and without tumors. More generally, this differentiates physiological structures through deep learning. Applying neural networks to medical images has implications of faster and more precise diagnoses automatically, and this research introduces neural networks into the medical field where it has little current use. The rest of the paper is organized as follows: In Section 2, we introduce a literature review of the current methods that are addressing the problem of brain tumor detection. Next, the material and methods used in this study are described in detail in section 3. Experimental results are presented in Section 4. Sections 5 and 6 provide the results discussion and finally the proposed techniques conclusions.

2 Literature Survey

In the early years, traditional methods were employed for tumour detection using unsupervised and supervised feature extraction techniques for classification. The brain tumour segmentation employed representative learning using atlas-based outlier detection [2] along with joint registration [6] making use of a tumour growth model and the Expectation Maximization algorithm. The unsatisfactory results of traditional approaches as pushed the research towards exploring deep learning based methods.

In recent years the importance of detection of brain tumors has led to the proposal of various automated and semi-automatic methods for type classification and segmentation of them[7], [20], [19]. Various methods are proposed for image classification, extracting features from the input data, and applying classification algorithms on the extracted features. For instance, for brain tumor classification, Cheng et al. [4] extracted the region of interest (ROI) by using morphological operation. Then, for better feature extraction, they used a space pyramid matching (SPM) method. Extracted features were given to some classifiers such as SVM, SRC, and KNN to perform the classification operation [5]. In another approach. Asodekar et al. [1], after shape-based features extraction, applied SVM, and random forest classifiers for the classification task. More recently deep learning method have come to the forefront, as an example, Abiwinanda et al. [15] applied CNN for tumor classification. Ismael et al. used a CNN network on MR images and statistical image features as input data for tumor classification [10].

The application of fully-conected convolutional network has been extensively researched. The state-of-art models have similar architecture with best performance of the BRaTS dataset. [16] employed 2D CNN with augmented dataset to imrpove performance. A faster network was implemented by [8] with cascaded CNN with 40 times speed up. The best results were shown by [11] using residual connections in the CNN model.

In our study, we experiment attention based deep model for tumour detection and further implement several segmentation models with state-of-art architectures. We propose to improve the performance using smart techniques to deal with class imbalance during training and present robust metrics.

3 Dataset

3.1 Detection Dataset

Dataset used for detection consists of free accessible MR images categorized into two classes as normal and tumor. The images in the dataset were collected by field experts, such as doctors and radiologists and shared on the internet[1]. The total number of images are 253 and each image was obtained from the volunteer patients. Therefore, the dataset has a heterogeneous structure. The number of tumor images is 155 while the number of normal samples is 98. The resolution of the images is not stable and the image quality is not high. The images have been converted to JPEG format. Sample images of the classes in the dataset are shown below.

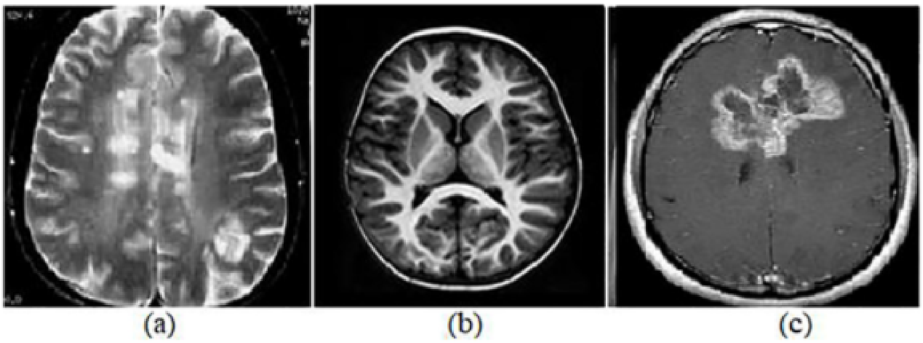
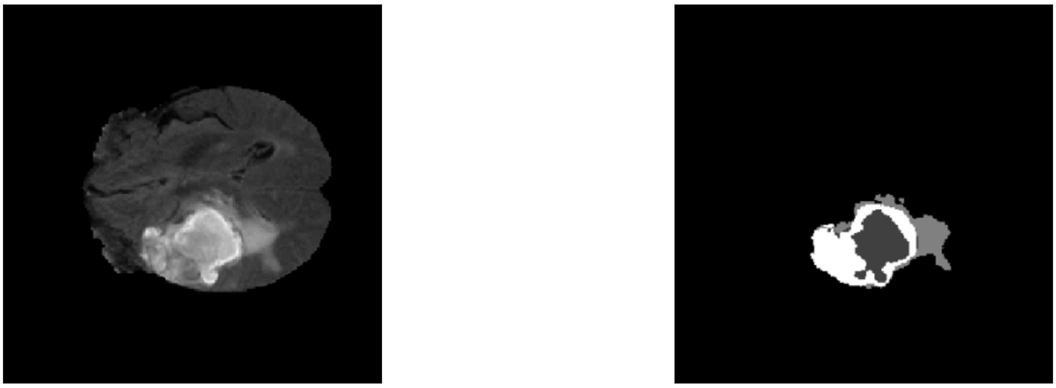


Figure 1: (a) and (b) normal samples, (c) a tumor sample

3.2 Segmentation Dataset

We used the BraTS 2015 dataset to train our segmentation model. 1550 MRI image slices of T1 modality were used for training and 165 images were used for testing. The ground truth consisted of 4 classes: Background, Tumour core (TC), Whole tumour (WT) and Enhancing Tumour (ET). The images have been converted to JPG format. Sample images of the input and segmentation ground truth in the dataset are shown below.

[1] Dataset - https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection



(a) Input image

(b) Ground truth

Figure 2: Segmentation dataset

4 Methodology

With the advent of deep learning in the image recognition domain, a lot of neural net architecture based on convolution neural net are launched every year, hence the accuracy of these algorithm is increasing day by day. Initially these recognition tasks were confined to real world pictures like human, animals and other object and much was not done in the medical domain. However, in recent years a lot of these models are also used for medical domain. A prominent problem statement in this domain is detection and segmentation of Brain Tumors.

We implement several SOTA architectures used in classification and segmentation for our problem of Brain Tumor Detection and Segmentation and evaluate their performance, additionally we combine image processing and data augmentation techniques to improve quality of data and make our models more generalized.

The following architecture are implemented :

Classification

1. AlexNet
2. VGG-16
3. GoogleNet
4. ResNet
5. BrainMRNet

Segmentation

1. Mobile U-Net
2. Deeplab
3. PSPNet
4. Encoder-Decoder

4.1 Classification Model Description

4.1.1 AlexNet

AlexNet[12] is one of the most popular neural network architectures to date proposed by Alex Krizhevsky , it is based on convolutional neural networks.

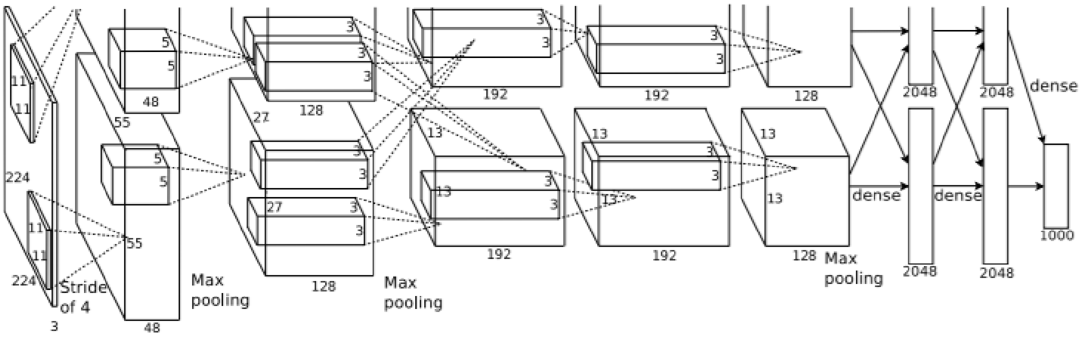


Figure 3: AlexNet Architecture

The architecture is comprised of eight layers in total, out of which the first 5 are convolutional layers and the last 3 are fully-connected. The first two convolutional layers are connected to overlapping max-pooling layers to extract a maximum number of features. The third, fourth, and fifth convolutional layers are directly connected to the fully-connected layers. All the outputs of the convolutional and fully-connected layers are connected to ReLu non-linear activation function. The final output layer is connected to a softmax activation layer, which produces a distribution of 1000 class labels.

4.1.2 VGG-16

Major improvements of VGG[18], when compared to AlexNet, include using large kernel-sized filters (sizes 11 and 5 in the first and second convolutional layers, respectively) with multiple (3×3) kernel-sized filters, one after another.

The images are passed to a stack of convolutional layers with small receptive-field filters of size (3×3). In a few configurations the filter size is set to (1 × 1), which can be identified as a linear transformation of the input channels (followed by non-linearity).

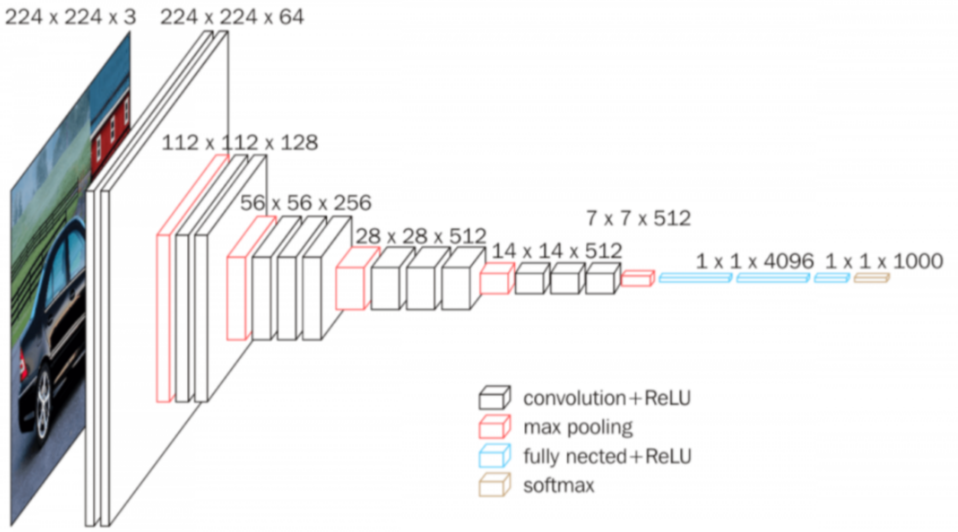


Figure 4: VGG-16 Architecture

The stride for the convolution operation is fixed to 1. Spatial pooling is carried out by five max-pooling layers, which follow several convolutional layers. The max-pooling is performed over a (2 × 2) pixel window, with stride size set to 2.

The configuration for fully-connected layers is always the same; The first two fully connected layers have 4096 channels each, the third fully connected layer performs binary classification. All the hidden layers for the VGG network are followed by the ReLu activation function.

4.1.3 GoogleNet

Also known as Inception Network-V1[21] was one of the major breakthroughs in the fields of Neural Networks, particularly for CNNs.To solve the problem of overfitting due to depth of deep learning architectures Szegedy et al proposed the GoogleNet architecture with the idea of having filters with multiple sizes that can operate on the same level. With this idea, the network actually becomes wider rather than deeper. Below is an image showing a Naive Inception Module.

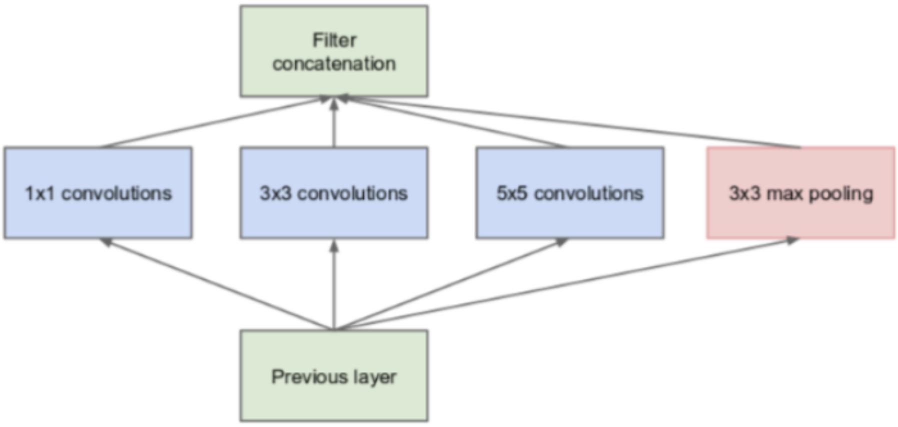


Figure 5: Inception Network

The convolution operation is performed on inputs with three filter sizes: (1 × 1), (3 × 3), and (5 × 5). A max-pooling operation is also performed with the convolutions and is then sent into the next inception module. An extra (1 × 1) convolution is added before the (3 × 3) and (5 × 5) convolutions to reduce the dimensions of the network.

The GoogleNet Architecture is 22 layers deep, with 27 pooling layers included. There are 9 inception modules stacked linearly in total. The ends of the inception modules are connected to the global average pooling layer. Below is a zoomed-out image of the full GoogleNet architecture.

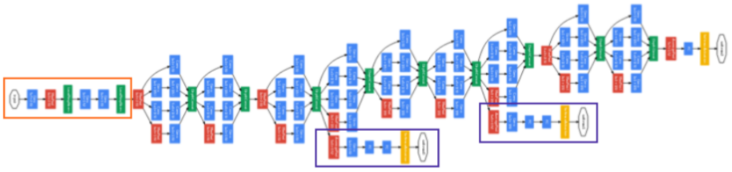


Figure 6: GoogleNet Architecture

4.1.4 ResNet

It is observed that if the number of layers is massive, network is prone to over-fitting. Increasing network depth also leads to vanishing gradient problem - where the back-propogated gradients may become infinitely small due to multiple multiplication. To overcome this Res-Net[9] introduces shortcut connections, which skips one or more layers.

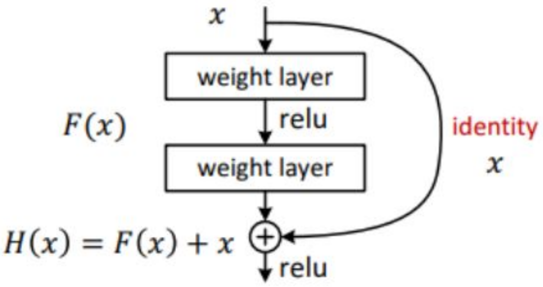


Figure 7: Residual Block

These shortcut connections perform identity mapping, which neither adds extra parameters nor does it increase computational complexity. This helps in solving the problem of vanishing gradient by allowing an alternative path for the gradient to flow through. Therefore we can construct a deeper model without risking a decrease in performance.

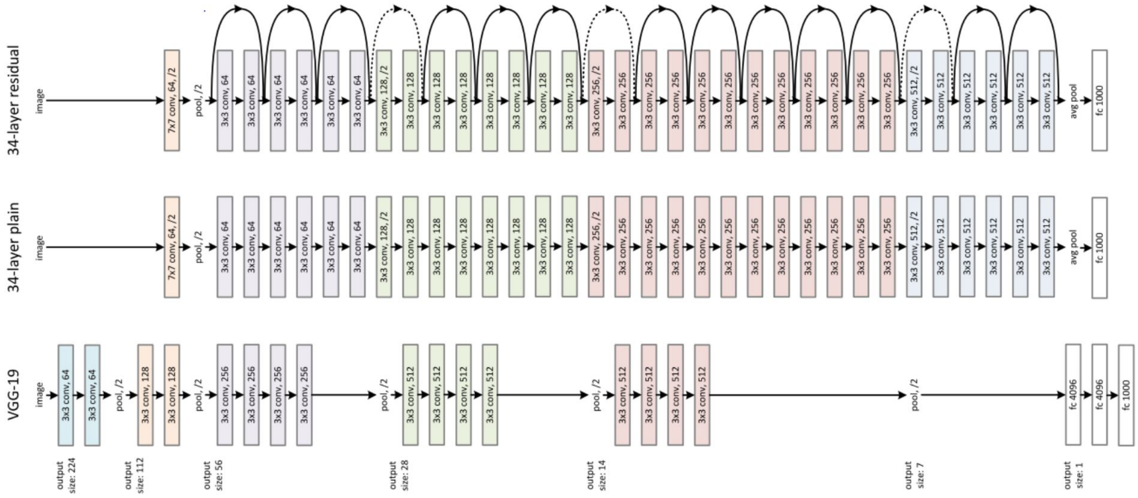


Figure 8: Res-Net Architecture

Components of a network include 3*3 filters, CNN down-sampling layers with stride 2, global average pooling layer and a 1000-way fully-connected layer with softmax in the end.

4.1.5 BrainMRNet

The BrainMRNet[22] model incorporates some of the layers of CNN architectures in its overall structure (Convolution, pooling, etc.). In addition, the dense layer is used before the classification stage. The BrainMRNet architecture consists of four main submodules:

Convolution and Dense Blocks These blocks consist of convolution and dense layers with batch normalization followed by a ReLU activation function as shown below.

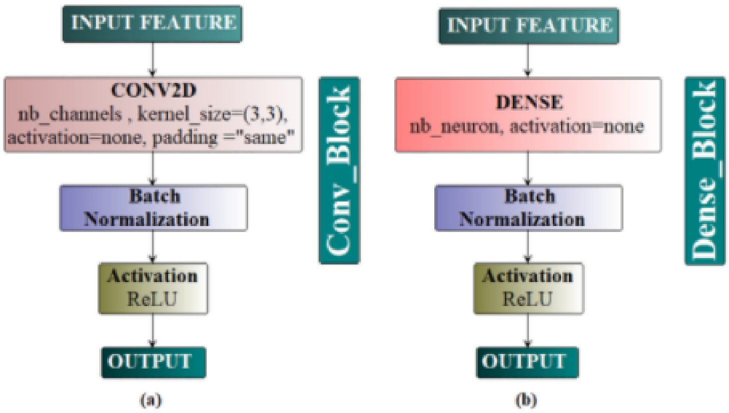


Figure 9: Convolution and Dense Blocks

CBAM Modeule CBAM allows the model to pay attention to the relevant areas of image. That is, it seeks important feature values on the image. The CBAM module consists of a combination of two sub-modules

- Channel Attention Module
- Spatial Attention Module

The channel attention module represents the channels of the activation maps extracted from the images. In fact, this module is useful for directing the neural network to important regions in feature maps. Inaddition, the channel module compresses the feature maps extracted from the images.

The spatial module uses an output pool similar to the channel module and transfers the obtained values to the convolution layer. The channel module then combines features extracted in both average and maximum pooling into a shared network.

In order to calculate the spatial attention module related operations, firstly average pooling and maximum pooling operations are applied along the channel axis. Then, the two processes (average and maximum pooling) are combined to form an effective feature identifier. Finally, the obtained information is given to the next layer.

The designs of the channel attention module and spatial attention module used in this study are shown below.

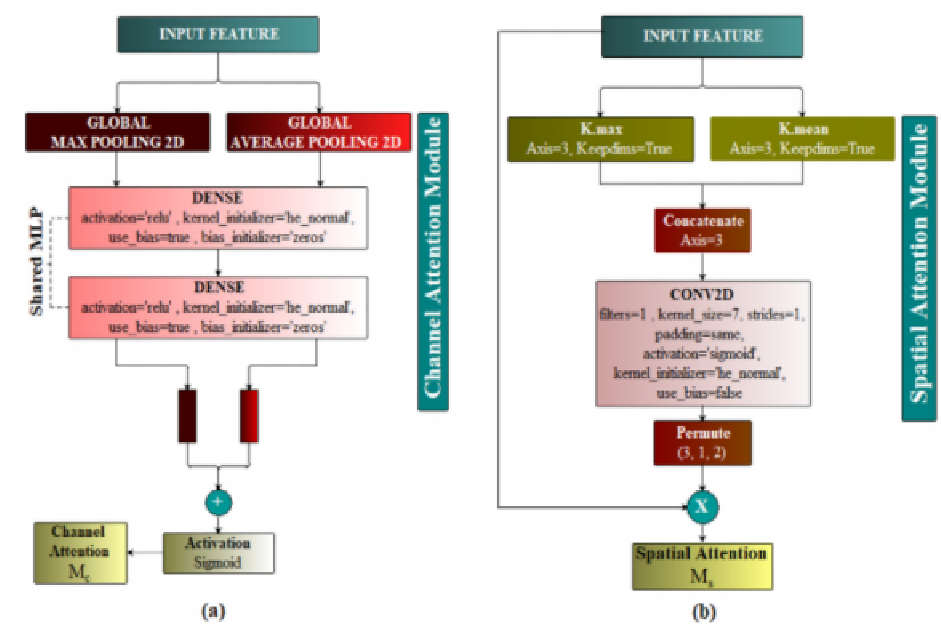

INPUT FEATURE
GLOBAL MAX POOLING 2D
GLOBAL AVERAGE POOLING 2D
Channel Attention Module
Shared MLP
DENSE
activation='relu' , kernel_initializer='he_normal',
use_bias=true , bias_initializer='zeros'
DENSE
activation='relu' , kernel_initializer='he_normal',
use_bias=true , bias_initializer='zeros'
+
Activation Sigmoid
Channel Attention M_c
(a)
INPUT FEATURE
K.max Axis=3, Keepdims=True
K.mean Axis=3, Keepdims=True
Spatial Attention Module
Concatenate Axis=3
CONV2D
filters=1 , kernel_size=7, strides=1,
padding=same,
activation='sigmoid',
kernel_initializer='he_normal',
use_bias=false
Permute (3, 1, 2)
X
Spatial Attention M_s
(b)

Figure 10: Channel and Spatial Attention

CAM and SAM focus on “what” and “where" respectively. These blocks contribute to performance by creating more efficient features when processing features.

Residual block The residual blocks used in the model feed the successive layers as well as directly feed the layers beyond two or three layers. In addition, the residual block merges the layers approximately two to three jumps away by applying additional processing to the starting layer.

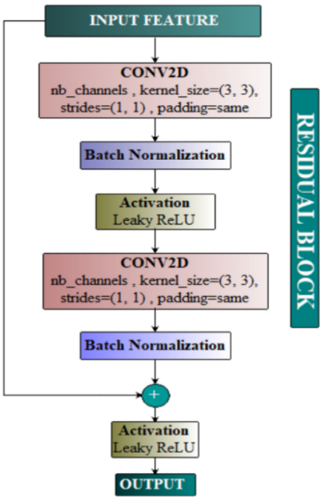

INPUT FEATURE
CONV2D
nb_channels , kernel_size=(3, 3),
strides=(1, 1) , padding=same
Batch Normalization
Activation
Leaky ReLU
CONV2D
nb_channels , kernel_size=(3, 3),
strides=(1, 1) , padding=same
Batch Normalization
+
Activation
Leaky ReLU
OUTPUT
RESIDUAL BLOCK

Figure 11: Residual Block

Hypercolumn technique The hypercolumn technique maintains the sequence of activation maps of each pixel on the image extracted along the model. And it transfers them to an array. The objective is to retrieve spatial location information (features) from the previous layers and allow for more efficient feature selection. In contrast to the hypercolumn technique, in convolutional networks, results are normally processed with the features transferred to the output of the last layer. In other words,

4.2 Segmentation Model Description

The goal of semantic segmentation is the same as traditional image classification in remote sensing, which is usually conducted by applying traditional machine learning techniques such as random forest and maximum likelihood classifier.

4.2.1 Mobile-UNet

U-net[17] architecture can be broadly thought of as an encoder network followed by a decoder network. The UNet model comprises of:

- The encoder is the first half in the architecture diagram (Figure 2). It usually is a pre-trained classification network like VGG/ResNet to encode the input image into feature representations at multiple different levels.
- The decoder semantically projects the discriminative features learnt by the encoder onto the pixel space (higher resolution) to get a dense classification.

The decoder consists of upsampling and concatenation followed by regular convolution operations.

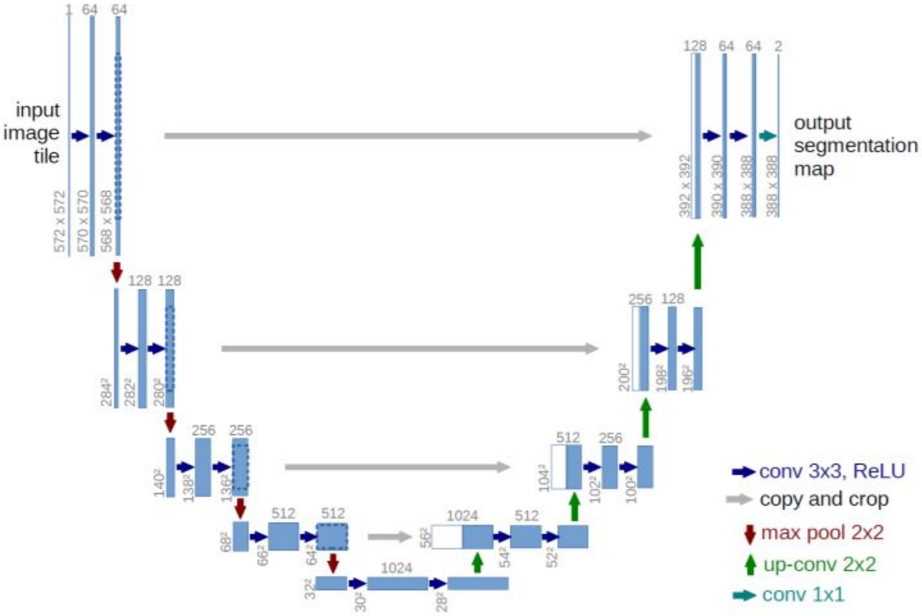


Figure 12: UNet Architecture

We perform upsampling to restore the condensed feature map to the original size of the input image. There are a few ways of upsampling such as Nearest Neighbor, Bilinear Interpolation, and Transposed Convolution from simplest to more complex.

While upsampling in the network we are also concatenating the higher resolution feature maps from the encoder network with the upsampled features in order to better learn representations with following convolutions.

Mobile nets were introduced by Google, which is a light weight neural network intended to be used in computationally limited environment. Mobile UNet follows a UNet based encoder decoder architecture with a few modifications for computational efficiency

- Introduces depthwise conv block to speed up inference speed.
- Shows high accuracy vs inference speed ratio.
- Additional parameters can be used to control the trade-off between accuracy and speed.

4.2.2 Encoder-Decoder

The encoder-decoder network consists of 2 parts the encoding and decoding network. The job of the encoder being to create a low-resolution representation that captures the semantic components in the input image and the job of the decoder being to map those back to the pixel level. The accuracy of the detail in the back-to-pixel mapping in the decoder is aided by the skip links from the corresponding levels in the encoder.

4.2.3 PSPNet

The PSPNet[23] architechture takes into account the global context of the image to predict the local level predictions hence gives better performance. The architecture contains two parts, i.e an Encoder and a Decoder:

The **PSPNet encoder** contains the CNN backbone with dilated convolutions along with the pyramid pooling module.

Dilated Convolution We replace the traditional convolutional layers with Dilated convolution layers, which helps in increasing the receptive field. Hence the feature recieved at the end of the backbone contains richer features. The receptive field for dilated convolution is larger as compared to the standard convolution. The size of the receptive field indicates how much context information we use. In PSPNet, the last two blocks of the backbone have dilation values 2 and 4 respectively.

Pyramid Pooling Module The pyramid pooling module is the main part of this model as it helps the model to capture the global context in the image which helps it to classify the pixels based on the global information present in the image.

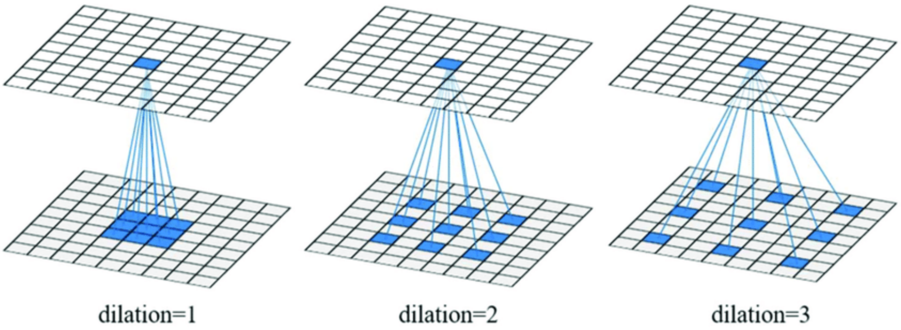


Figure 13: Dilated Convolutions

The feature map from the backbone is pooled at different sizes and then passed through a convolution layer and after which upsampling takes place on the pooled features to make them the same size as of the original feature map. Finally, the upsampled maps are concatenated with the original feature map to be passed to the decoder. This technique fuses the features different scales hence aggregating the overall context.

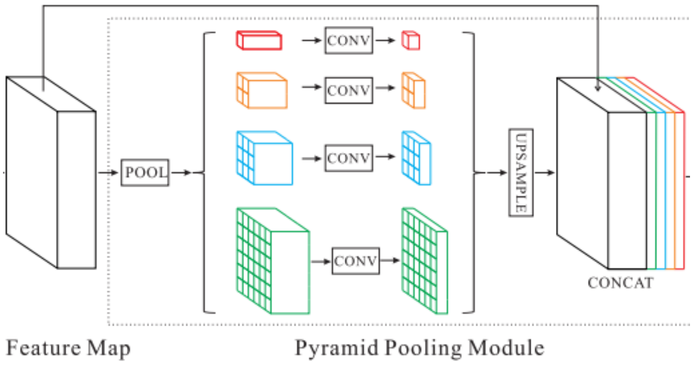


Figure 14: PSPNet Architecture

PSP Net Decoder The PSPNet model is not a complete segmentation model in itself, it is just an encoder. Extracted features by the encoder are passed to the decoder and converts them into predictions by passing them into its layers. The decoder is just another network which takes in features and results into predictions. The most common decoders that are found in various implementations of PSPNet is a convolution layer followed by a 8x bilinear-upsampling.

4.2.4 Deeplab

Deeplab[3] presents an architecture for controlling signal decimation and learning multi-scale contextual features. It uses a pre-trained ResNet(seen in previous section) as its main feature extractor network. The last ResNet block uses atrous or dilated convolution instead of regular convolution as seen in PSPNet. Also, each convolution (within this new block) uses different dilation rates to capture multi-scale context.

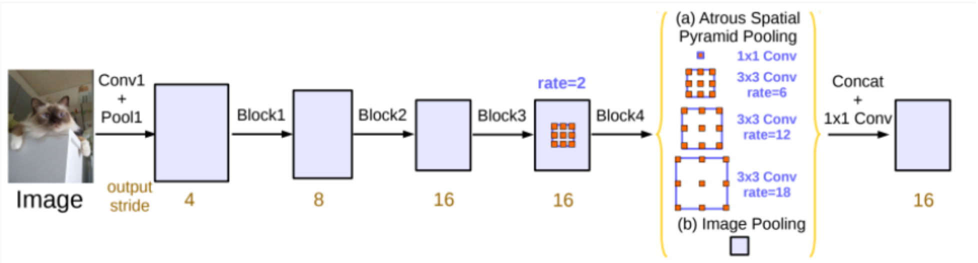


Figure 15: Deeplab Architecture

Additionally, on top of this new block, it uses Atrous Spatial Pyramid Pooling (ASPP). ASPP provides the model with multi-scale information. It adds a series atrous convolutions with different dilation rates to capture long range contexts. ASPP performs parallel 1x1 convolution and three convolutions with dilation rates =(6,12,18).

Also, to add more global context information, ASPP incorporates image-level features. First, it applies GAP to the output features from the last atrous block. Second, the resulting features are fed to a 1x1 convolution with 256 filters. Finally, the result is bilinearly upsampled to the correct dimensions.

5 Implementation Details

5.1 Preprocessing and Augmentation

5.1.1 Classification

MRI images of various sizes were resized to 224x224 pixels and normalized. Various transforms as augmentation. Additionally we perform augmentation techniques to artificially increase the diversity of our dataset in order to increase the dataset size. Transforms like Vertical Flip=0.5, Horizontal Flip=0.5, Random Brightness Contrast=0.3, Shift Scale Rotate=0.5, Shift Limit=0.2, Scale Limit=0.2, Rotate Limit=20° were applied to improve model stability and generalise its ability.

5.1.2 Segmentation

The MRI images were cropped to 320 x 320 pixels. The BraTS dataset is preprocessed as follows. First, bias field correction was performed using N4ITK method to remove the bias introduced due to the MRI scans. Skull stripping is performed using the BET2 algorithm to remove artifacts not relevant to segmentation. Finally, the images are intensity normalised to the range [0,1]. Augmentation techniques like horizontal flip, vertical flip, random rotation and brightness alteration were used to increase the diversity of the segmentation dataset as well.

5.2 Loss Function

5.2.1 Classification

We use categorical cross entropy loss for the tumour detection task which is a binary classification. We apply L2 regularisation to make the model more robust with reduced overfitting.

$$Loss = -\sum_{n=1}^{N} y_{n,true}\log(y_{n,pred}) + (1 - y_{n,true})\log(1 - y_{n,pred}) \tag{1}$$

5.2.2 Segmentation

In the segmentation dataset there is a class imbalance. If the ordinary binary cross entropy loss is applied for training the loss calculation would be biased towards class with more pixels and training would not be done properly . Hence to prevent this from happening the loss is multiplied with corresponding weights. This is done as shown in the equation

$$Loss = -\sum_{n=1}^{4} w_n y_{n,true}\log(y_{n,pred}) \tag{2}$$

where the weights, w_n are calculated as follows,

$$w_n = \frac{\text{No.of pixels of class n}}{\text{Total number of pixels}} \tag{3}$$

5.3 Training parameters

The training was using Keras library of tensorflow and PyTorch. We used adam optimizer for training with a learning rate of 1e-4, with learning decay of 0.9. Learning rate plateau was applied to reduce the learning rate according to val loss improvement. We also use dropout to avoid overfitting We used binary cross entropy as our loss function and softmax layer was used to limit the probablity between 0 and 1 in last layer.

6 Results and Discussions

6.1 Classification

In our experiments, we compare the models using the following metrics: Accuracy (Acc), Sensitivity (Se), Specificty (Sp), F1 score (F1) and Precision (Pr) computed from the confusion matrix with True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) using the following formula

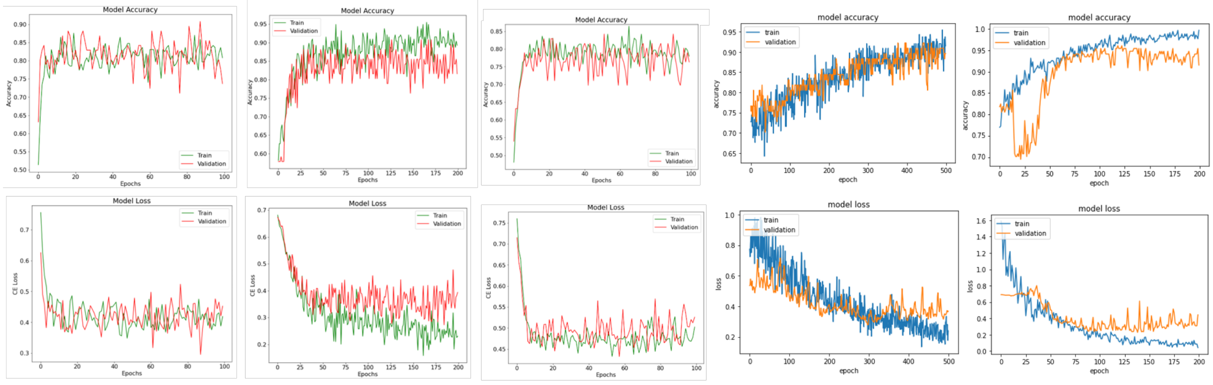
$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Se = \frac{TP}{TP + FN} \tag{5}$$

$$Sp = \frac{TN}{TN + FP} \tag{6}$$

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{7}$$

$$Pr = \frac{TP}{TP + FP} \tag{8}$$



(a) VGG-16 (b) Googlenet (c) Resnet (d) Alexnet (e) BrainMRnet

Figure 16: Training curves for tumour detection models

The training process was successfully implemented using pytorch modules with Tesla K80 GPU for hardware acceleration. The limited dataset resulted in jagged curves with oscillations, which however was slightly improved with data augmentation. BrainMRnet outperformed the rest of the models with 97.2% accuracy, 97.1% specificity and 96.8% sensitivity. This is slightly better than base paper due to L2 regularisation. The other models are overfit during training resulting in comparitively poor test results.

Architecture	Acc	Se	Sp	F1	Pr
BrainMRnet	**0.972**	**0.968**	**0.971**	**0.942**	**0.925**
Resnet	0.889	0.8438	0.9231	0.8852	0.931
Googlenet	0.908	0.846	0.963	0.903	0.965
VGG16	0.931	0.886	0.944	0.949	0.951
Alexnet	0.947	0.893	0.952	0.956	0.941

Table 1: Computed metrics for different implemented models

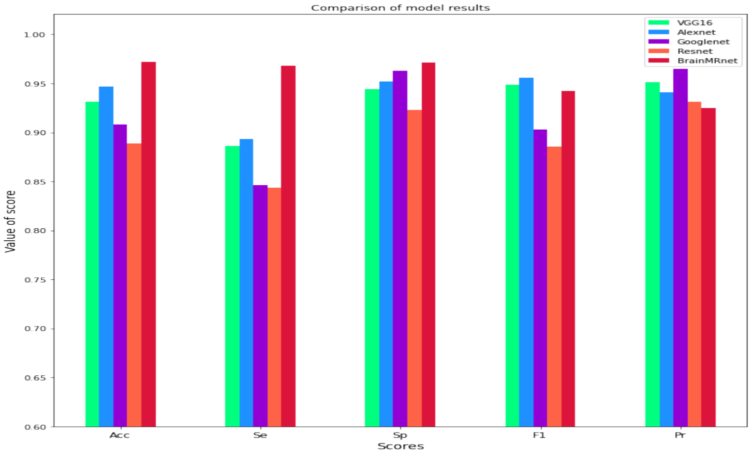


Figure 17: Evaluation metrics for tumour detection models

6.2 Segmentation

The segmentation models were implemented in Tensorflow and were run on Tesla K80 GPU for hardware acceleration. The models were trained for a total of 50 epochs with weighted categorical cross entropy as the loss function and with LR scheduling for the optimizer. The weights applied to each class were: Background (1.0), Whole tumour (4.0), Tumour core (15.0) and Enhanced Tumour (10.0). Segmentation models are best understood using the dice and Intersection of Union (IoU) scores. Additionally, the class-wise accuracies are also computed and tabulated.

Table 2 summarizes and compares the metrics obtained for the different implemented models. It is observed that Mobile-UNet outperforms the other models in all the metrics obtaining a highest dice score of 96% and an IoU score of 83%. Deeplab had a comparitvely lower dice score than UNet as they weren't able to learn the contours very well. Figure 19 shows a graphical comparison of Table 2.

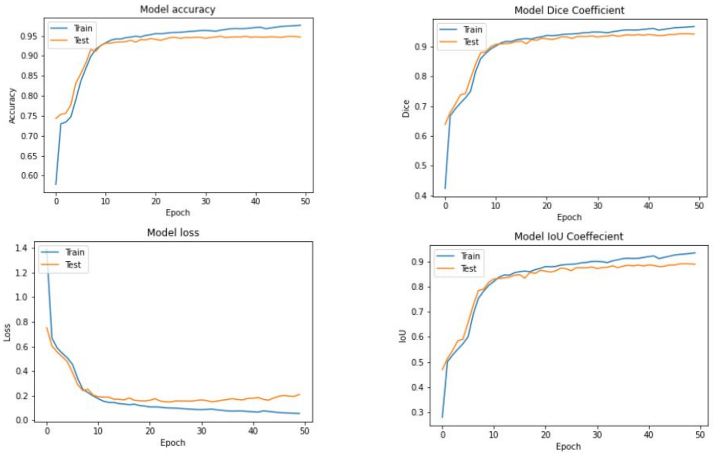


Figure 18: Training curves for Mobile-Unet

Architecture	Accuracy	Dice	IoU	Background	TC	WT	ET
Mobile-Unet	**0.97**	**0.96**	**0.83**	**0.98**	**0.82**	**0.82**	**0.89**
Deeplab	0.93	0.90	0.78	0.93	0.78	0.76	0.87
PSPNet	0.87	0.87	0.72	0.88	0.73	0.71	0.74
Encoder-Decoder	0.89	0.82	0.71	0.90	0.72	0.71	0.75

Table 2: Computed metrics for different implemented models

Paper	TC	WT	ET
Mobile-Unet	**0.82**	**0.82**	**0.89**
Haveai et al. [8]	0.81	0.72	0.58
Pereira et al. [16]	0.88	0.83	0.77
Kamnitsas et al. [11]	0.89	0.75	0.72

Table 3: Comparison of metrics with existing literature

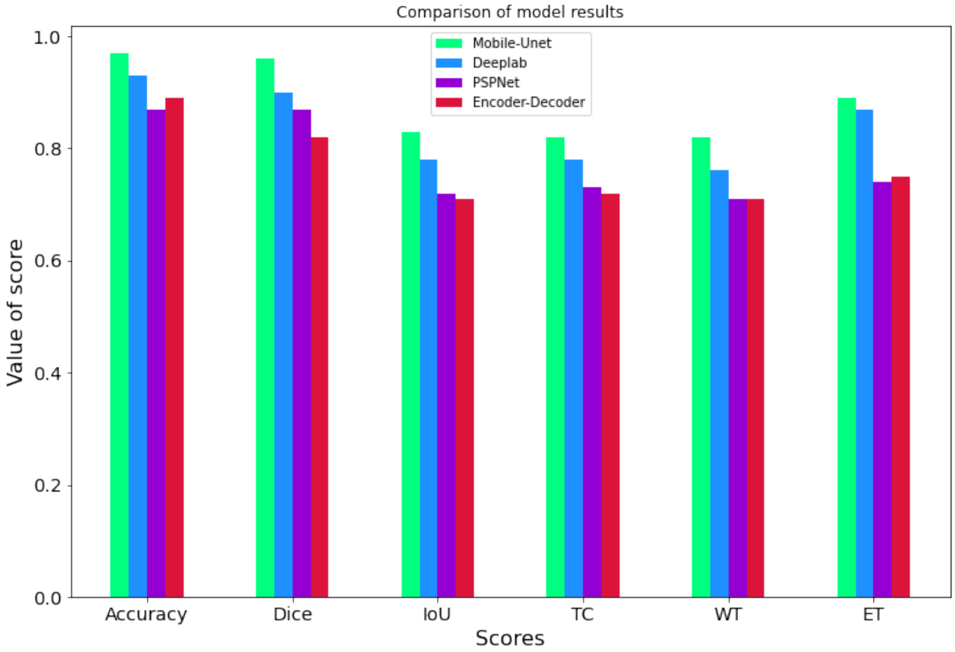
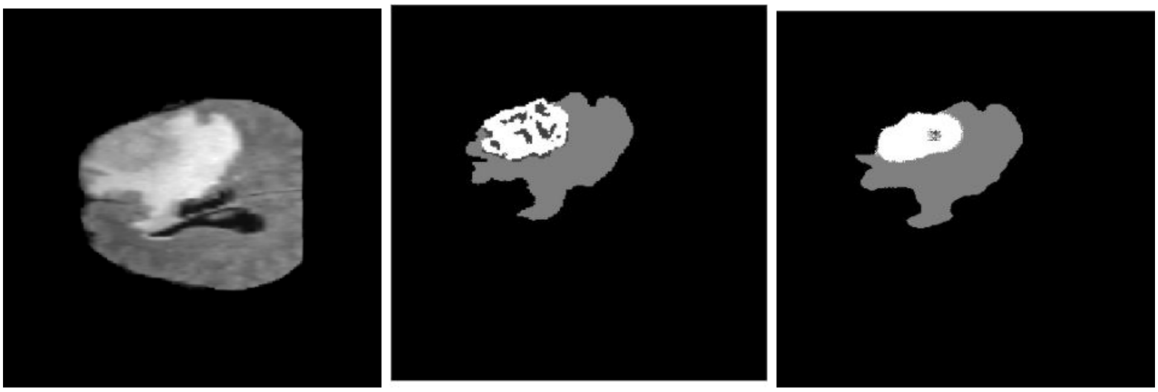
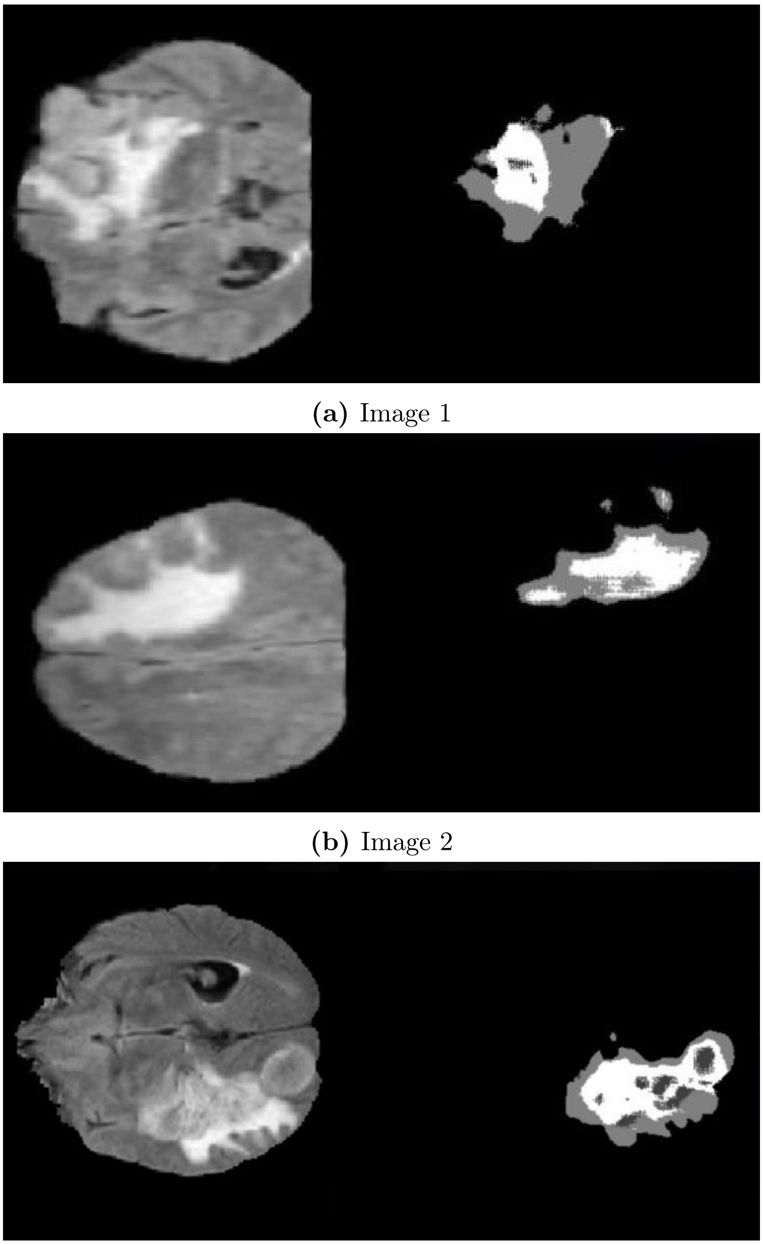


Figure 19: Comparison of metrics for segmentation models



(a) Input image (b) Ground truth (c) Predicted output

Figure 20: Output of Mobile-UNet for validation set



(a) Image 1

(b) Image 2

(c) Image 3

Figure 21: Output of Mobile-UNet for test set

7 Conclusion

We implemented brain tumour detection and segmentation using state-of-art architecture. We observed that spatial attention with regularised training improved the robustness of classification task using the proposed BrainMRnet model. Including hypercolumn to aggregate the features and residual connections to reduce the negative tendencies improved the performance of the model. We also developed Mobile-Unet model for segmentation with results better than its contemporary architectures. This was achieved using weighted loss function due to the unbalanced pixel ratio in classes. We observed that the results can be improved with data augmentation indicating the need to increase the data collection process for a more generalised model. Hence, we establish the superiority of artificial intelligence in tumour detection which can be of great importance for medical community in real-time emergency diagnostics.

8 Future works

In the future, efforts can be focused on the combining detection and segmentation into a single architecture, hence mobilising the diagnosis with more accuracy. The results of segmentation can be used as priori distribution for classification and vice-versa. Combining multi-modal imaging data from mutliple views, along with joint registration is prospective direction to work along. This way, the detection models can be made more robust and fast for quick diagnostic results.

References

[1] Bhagyashri Asodekar and Sonal Gore. “Brain Tumor Classification Using Shape Analysis of MRI Images”. In: *SSRN Electronic Journal* (Jan. 2019). DOI: 10.2139/ssrn.3425335.

[2] Babak Ehteshami Bejnordi et al. “A brain tumor segmentation framework based on outlier detection”. In: . *Medical image analysis* 8.3 (2004), pp. 275–283.

[3] Liang-Chieh Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[4] Jun Cheng et al. “Correction: Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition”. In: *PLoS ONE* 10 (Dec. 2015). DOI: 10.1371/journal.pone.0144479.

[5] Jun Cheng et al. “Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition”. In: *PLoS ONE* 10 (Oct. 2015). DOI: 10.1371/journal.pone.0140381.

[6] A Gooya et al. “Joint segmentation and deformable registration of brain scans guided by a tumor growth model”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI* 6892 (2011), pp. 532–540.

[7] Abdu Gumaei et al. “A Hybrid Feature Extraction Method With Regularized Extreme Learning Machine for Brain Tumor Classification”. In: *IEEE Access* 7 (2019), pp. 36266–36273. DOI: 10.1109/ACCESS.2019.2904145.

[8] Mohammad Havaei et al. “Brain Tumor Segmentation with Deep Neural Networks”. In: *CoRR* abs/1505.03540 (2015). arXiv: 1505.03540. URL: http://arxiv.org/abs/1505.03540.

[9] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[10] Mustafa R. Ismael and Ikhlas Abdel-Qader. “Brain Tumor Classification via Statistical Features and Back-Propagation Neural Network”. In: *2018 IEEE International Conference on Electro/Information Technology (EIT)*. 2018, pp. 0252–0257. DOI: 10.1109/EIT.2018.8500308.

[11] Konstantinos Kamnitsas et al. “DeepMedic for Brain Tumor Segmentation”. In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. by Alessandro Crimi et al. Cham: Springer International Publishing, 2016, pp. 138–149. ISBN: 978-3-319-55524-9.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[13] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (Dec. 2017), pp. 60–88. ISSN: 1361-8415. DOI: 10.1016/j.media.2017.07.005. URL: http://dx.doi.org/10.1016/j.media.2017.07.005.

[14] McKinney PA. "McKinney PABrain tumours: incidence, survival, and aetiology". In: *Journal of Neurology, Neurosurgery & Psychiatry* (2004).

[15] Krishna Pathak et al. "Classification of Brain Tumor Using Convolutional Neural Network". In: (2019), pp. 128–132. DOI: 10.1109/ICECA.2019.8821931.

[16] Sérgio Pereira et al. "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images". In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1240–1251. DOI: 10.1109/TMI.2016.2538465.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[18] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[19] Zahra Sobhaninia et al. *Brain Tumor Segmentation by Cascaded Deep Neural Networks Using Multiple Image Scales*. 2020. arXiv: 2002.01975 [eess.IV].

[20] Zahra Sobhaninia et al. *Brain Tumor Segmentation Using Deep Learning by Type Specific Sorting of Images*. 2018. arXiv: 1809.07786 [cs.CV].

[21] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].

[22] Mesut Toğaçar, Burhan Ergen, and Zafer Cömert. "BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model". In: *Medical Hypotheses* 134 (2020), p. 109531. ISSN: 0306-9877. DOI: https://doi.org/10.1016/j.mehy.2019.109531. URL: https://www.sciencedirect.com/science/article/pii/S0306987719313416.

[23] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6230–6239. DOI: 10.1109/CVPR.2017.660.