AI for Medical Science

Focusing object detection

Rathod, Kishansinh (SRH Hochschule Heidelberg Student) - 11013344

Submitted to : Dr. Simon Ziegler

Index

1. Introduction

1.1.Description

AI is currently trending within the healthcare field, majorly for the tasks of patient engagement and adherence, diagnosis and treatment recommendations, and most of the administrative activities of the workforce. The use of deep learning in medical diagnosis to diagnose cancer is a big AI development in medicine. According to a recent study published in the Journal of the National Cancer Institute, the AI device can detect breast cancer with the same sensitivity as a typical breast radiologist. 95 percent confidence intervals have been demonstrated by both radiologists and the AI system. With AI networks' ability to continuously train themselves, there's a good chance that their efficiency will improve dramatically in the near future.

Increased use of imaging methods, such as X-rays, CAT scans, and MRIs, is one big development in the medical field, resulting in vast volumes of complex data. While the use of imaging in medical practice is growing, as is the workload associated with data processing, the number of qualified radiologists remains relatively constant. The amount of data being processed is outpacing processing capabilities, resulting in overworked radiologists. As the use of CT and MRI scans has increased in recent years, further work must be performed by the same number of radiologists. That’s where AI comes in the picture because AI is useful for automating processes that are either continues or repetitive but based on complex data that changes over time. DL algorithms excel at identifying complex patterns in unstructured data automatically. Deep learning is therefore especially intriguing in the field of medical imaging.

Deep learning technology’s ability to interpret images and identify patterns opens up the possibility of developing algorithms to aid doctors in diagnosing diseases more quickly and accurately. Furthermore, such algorithms can learn over the time continuously and improve their accuracy at predicting the correct diagnosis.

So in this research report, I provided a brief overview of such method currently being used in the field of medical imaging and diagnosis. Method such as Classification, Object detection and Image segmentation.

1.2.Image Classification vs Object Detection

One of the most common questions in the computer vision field is what the difference is between image classification, object detection, and image segmentation. All the newcomers in the field can probably get confused here. So, here is small explanation on the terms.

Consider the Image give here, by looking at Image we can directly identify that this is dog. Now question here is how? and the answer is because you were shown a picture and asked to classify the class to which class it belongs (a dog, in this case), and that is what Image Classification is about. In other words, **Classification** is a method of labelling in which a label is assigned to an image or video in order to address the question, "What is in this image or video?"

Now consider what if image has dog and cat both. Then how to label or classify this type of image. For that we can train our classifier for multi class images but what if we want to know the location of object? And process of getting location of object is known as Image Localization. **Object localization** involves drawing a bounding box around one or more objects in an image. It helps us to find the location of a single object in the image.

When we have multiple objects in the given image then we can use concept of **Object detection.** In simple word, Object detection combines these two techniques: it draws a bounding box around each object of interest in the image and assigns a class label to them. Given below is example of Image localization and Object detection

Image 1.2.2 : Image classification vs Object detection
https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81

In this project I tried to Implement these two techniques for medical images, to be precise Brain MRI images with tumor. So, for the classification task, my model classifies two different label, one image which contain tumor and one without tumor. When talking about object detection, I trained model with different state of art to find the specific position tumor in brain MRI images.

1.3. What is MRI? (Dataset Introduction)

Magnetic resonance imaging (MRI) is a medical imaging technique that produces detailed pictures of the organs and tissues in your body using a magnetic field and computer-generated radio waves. Big, tube-shaped magnets are the majority of MRI machines. The magnetic field temporarily realigns water molecules in your body while you lie inside an MRI unit. Radio waves cause faint signals to be created by these aligned atoms, which are used to create cross-sectional MRI images, such as slices in a bread loaf.

Neuroimaging techniques such as magnetic resonance imaging (MRI) have been shown to provide biological proof that neurodegenerative cognitive impairment is neurodegenerative, as they provide extensive information on subcortical structures, strong grey matter comparison, and brain tissue integrity. In particular, it is known that changes in the occipital, parietal, prefrontal and temporal lobes can be understood using MRI, including cortical injury, focal lesions and grey matter loss. As complex, unstructured data structures, brain MRI scans are characterized and therefore require sophisticated means to perform efficient, quantitative analysis. Although neuropsychological tests and MRI scans have been tried and checked for their diagnostic and psychometric strengths and shortcomings, there is a small body of work that has attempted to understand the combined effect of integrating these data sets for MCI diagnosis. [**Error! Reference source not found.**].

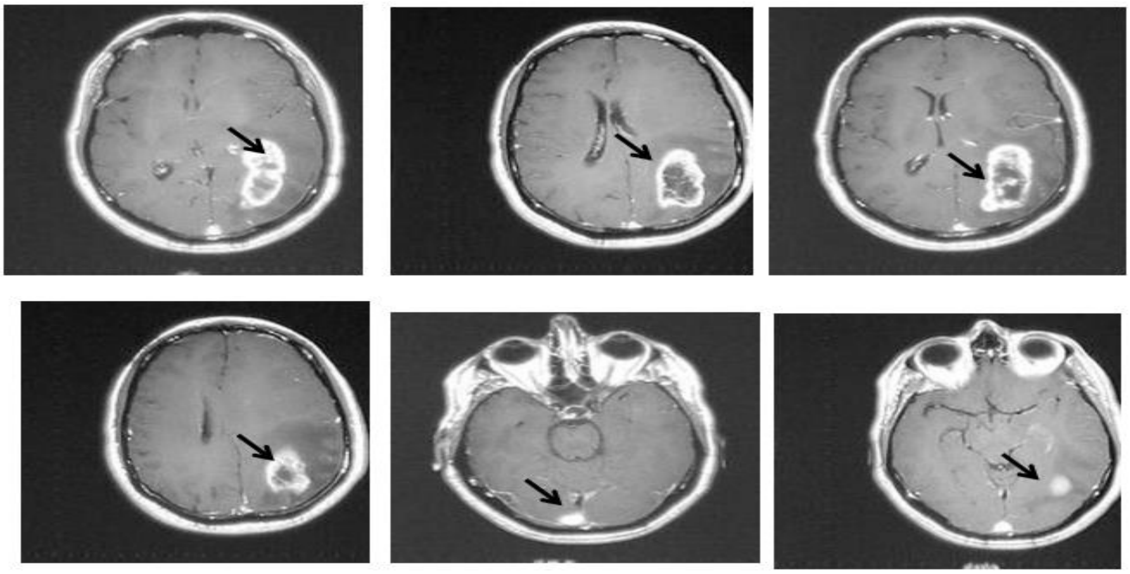


Image 1.3.1 : Sample MRI images of brain (https://www.researchgate.net/figure/Sample-MRI-images-of-brain-The-arrows-show-the-presence-of-tumors_fig1_228761309)

2. Research objective

During this research project I was intended to explore the field of Deep learning techniques for Medical Imaging. Hereby, I identify some of the research objectives

- **Implementation of transfer learning for medical images for classification problem**:
- **Find state-of-art techniques for object detection**
- **Implement object detection on brain mri images and compare results**

3. Image Classification

Image classification is the process of a computer analyzing an image and determining which "class" it belongs to. A class can be a category, such as "car," "animal," "building," and so on. For example, you input an image of a dog then Image classification is the process of the computer analyzing the image and telling you the probability that it's a dog. Image classification is no big deal for humans, but for machine it is a perfect example of Moravec's paradox.

(*In the 1980s, Hans Moravec, Rodney Brooks, Marvin Minsky and others articulated and discussed this AI paradox. As Moravec put it: "***It is comparatively easy to make computers exhibit adult level performance and difficult or impossible to give them the skills of a one-year-old.***" Moravec's paradox is a phenomenon surrounding the abilities of AI-powered tools. It observes that tasks humans find complex are easy to teach AI. Compared, that is, to simple, sensorimotor skills that come instinctively to humans.*) [3].

Early image classification problem was depended on raw pixel data. That means computer consider individual pixel of image to understand it. The issue is that two images of the same object may appear to be completely different. They may have a variety of backgrounds, angles, poses, and so on. Computers found it difficult to correctly 'see' and categorize images. And that's where deep learning came into the picture.

Deep learning is a form of machine learning that allows machines to learn from data. It is a subset of artificial intelligence (AI). Deep learning makes use of neural networks.
A neural network consists the different input filters through hidden layers of nodes. Each of these nodes processes the data and relays the results to the next layer of nodes. This continues until it reaches an output layer or final layer, at which point the computer provides the result. There is different type of neural network exist and amongst them most common and popular for image classification is CNN (convolutional neural network).
In this project I used deep learning and CNN for image classification. I worked on brain mri images (with brain tumor) to classify them in two categories: Images with tumor and images without tumor.

Work done so far

Before starting this project, I already worked on classification problem with different CNN model. I trained VGG16 from scratch. The dataset I used is explained below

Dataset

The dataset I choose is small open dataset from www.kaggle.com. The dataset contains 253 brain mri images with two labels (category): no and yes. Images with label 'yes' means images with brain tumor and label 'no' means images without tumor.
For the classification experiment, dataset divided in 80-20%. 80 percent for training purpose with both label and 20 percent of images for testing dataset.

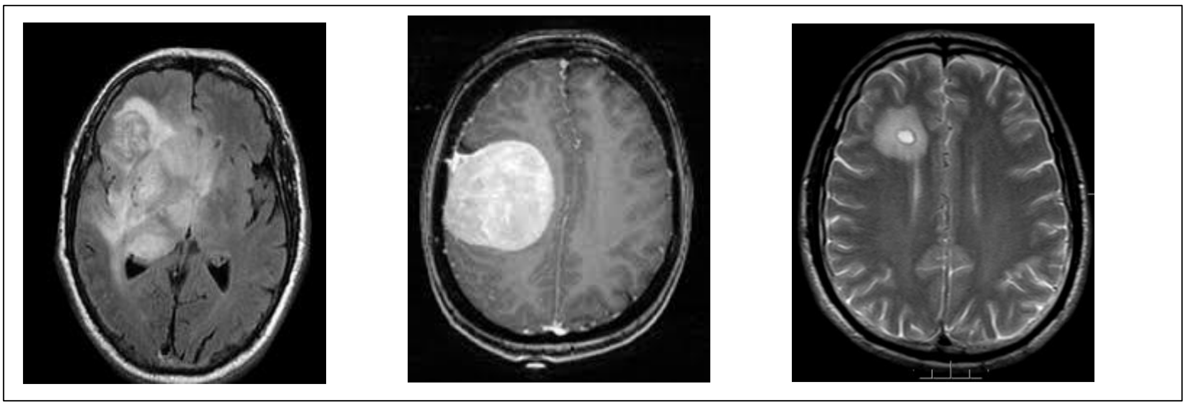


Fig 3.1 : Sample Image with label ***Yes***

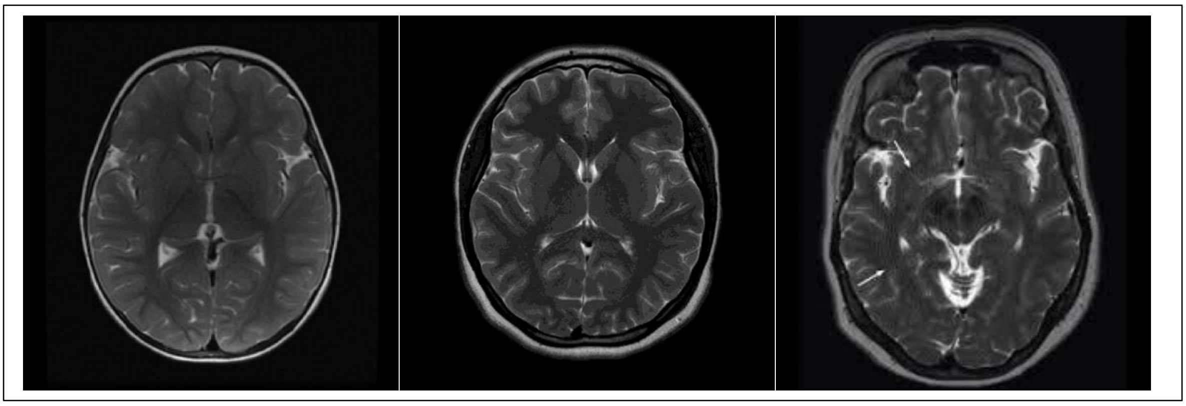


Fig 3.2: Sample Image with label ***No***

By using given dataset, I trained most famous VGG16 architecture. It was one of the famous model submitted to ILSVRC-2014. The model achieves 92.7% top-5 test accuracy in ImageNet (ImageNet is a dataset of over 14 million images with to 1000 classes). It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another[4]. Given below is the architecture of VGG16

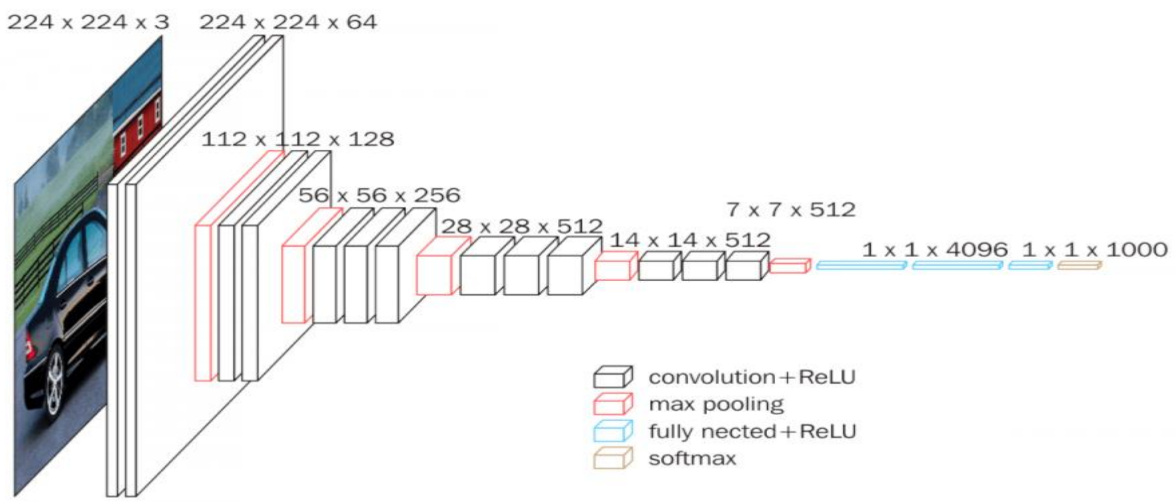


Image 3.3 : VGG16 architecture. [*https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c*](https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c)

VGG16 was trained on local machine with above explained dataset. Given below was the result from training. The model achieved very good training accuracy around 90% with validation accuracy around 75% which is good for small dataset. But the problem was here that when model tested on unseen image, it was able to correctly classify 3 images out of 7 unseen images. Which say somehow model is overfitted. Means during training it gives food accuracy but does not remember everything while training and gives false classification sometimes while check on unseen data.

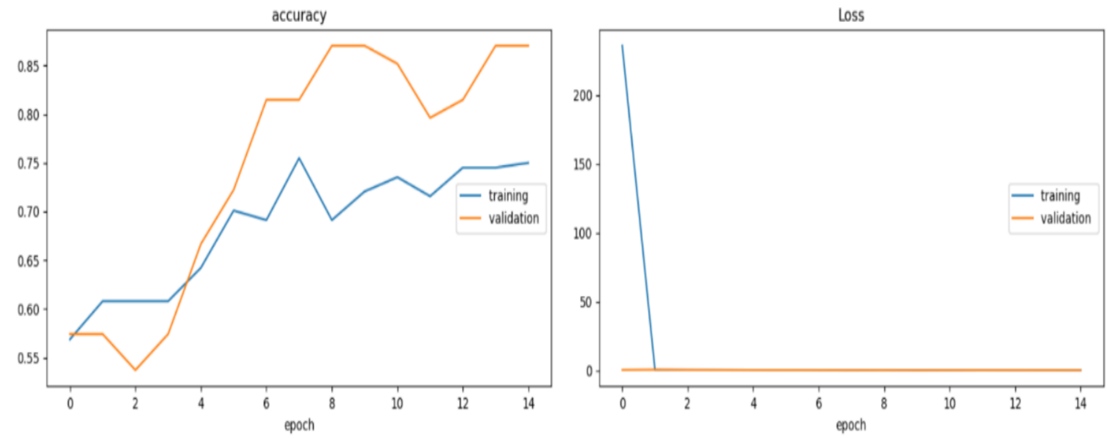


Image 3.4 : VGG16 Brain Mri tumor classification Accuracy vs Loss graph

To overcome this problem, I tried trial and error approach. I made several changes and addition in original VGG16 model and train them many times with changes to check the performance. I found some good addition to original VGG16, which gave slight better result compare to the original architecture by adding Batch Normalization and Droupout function after every convolution of VGG. The training graph is given below which shows architecture shows training accuracy 90% plus while validation is around 85%. Model was tested on same 7 unseen images and performed slight better. It was able to correctly classify five out of seven images. Which is very good but still we can not consider this result on unseen data while working on critical medical data. Medical data need higher accuracy.

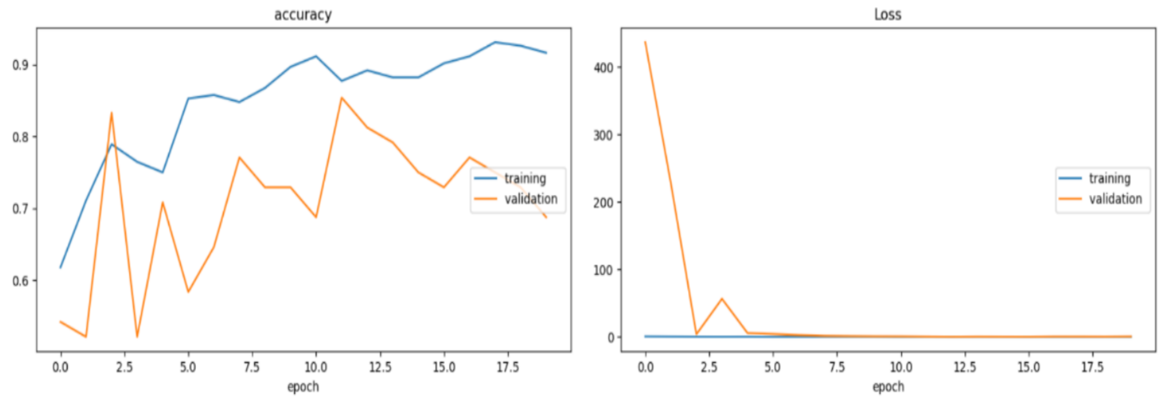


Image 3.5 : Proposed model Accuracy vs Loss graph

Now next to overcome this, I went for implement current trending method in deep learning which is transfer learning.

What is Transfer Learning?

Humans are born with the ability to transfer their skills through activities. We apply what we've learned from one task to solve other tasks that are identical. It is easy for us to transfer our skills or knowledge for the more related the tasks. Consider this statement to understand, "If you know math and statistics, then you can learn machine learning."

Machine learning and deep learning algorithms have traditionally been designed to operate in isolation. These algorithms have been honed to complete a particular task. If you make any changes then the models must be redesigned from the ground up.

Transfer learning is a ML technique where a model trained on one task and which can be reuse on a second related task.

"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned."

- (Chapter 11: Transfer Learning, Handbook of Research on Machine Learning Applications, 2009., n.d.)

Consider image 3.6 given below to better understand how transfer learning is different from traditional ML techniques

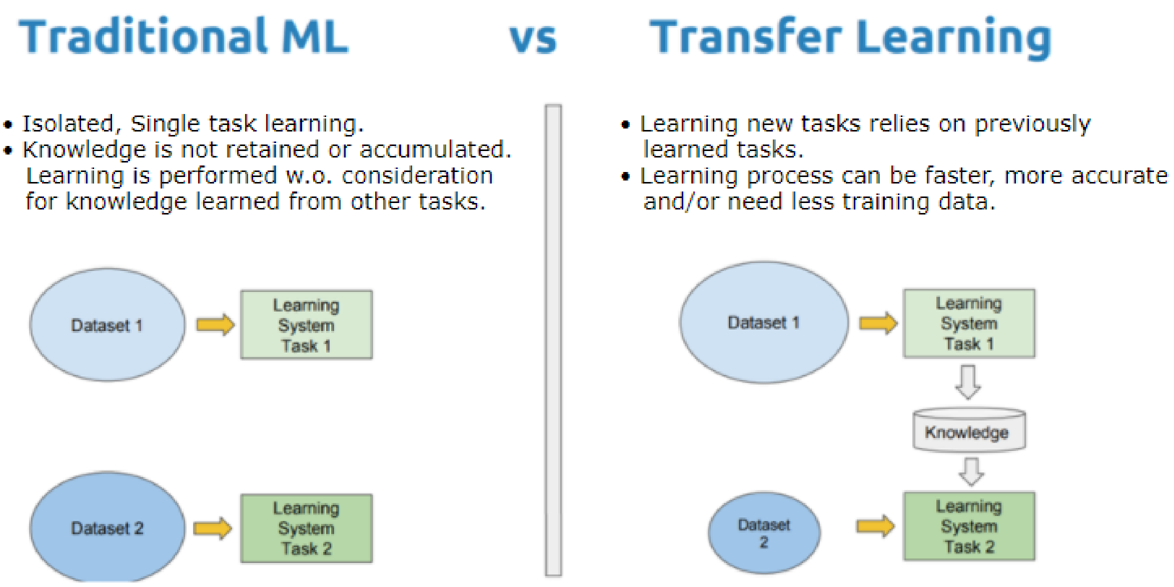


Image 3.6: Traditional ML vs Transfer Learning

Transfer learning can be implemented/use by two very common methods:

- Develop your own model
- Use of pre-trained model

In this I tried second method for this project. I use pre-trained model

Implementation of Transfer learning

Here, in older classification method for Brain tumor classification (MRI) was good during training but not optimal when it comes to unseen data. The best model was only able to recognize the 5 out of 7 unseen images correctly. So, I planned to apply transfer learning and check the results using experiment.

Here, I selected same VGG16 model which is already pre-trained on large dataset of imagenet. The stats related to experiment is given below:

Details

Trained on : Google colab

GPU : Tesla K80

Epochs : 50

Graphs of training

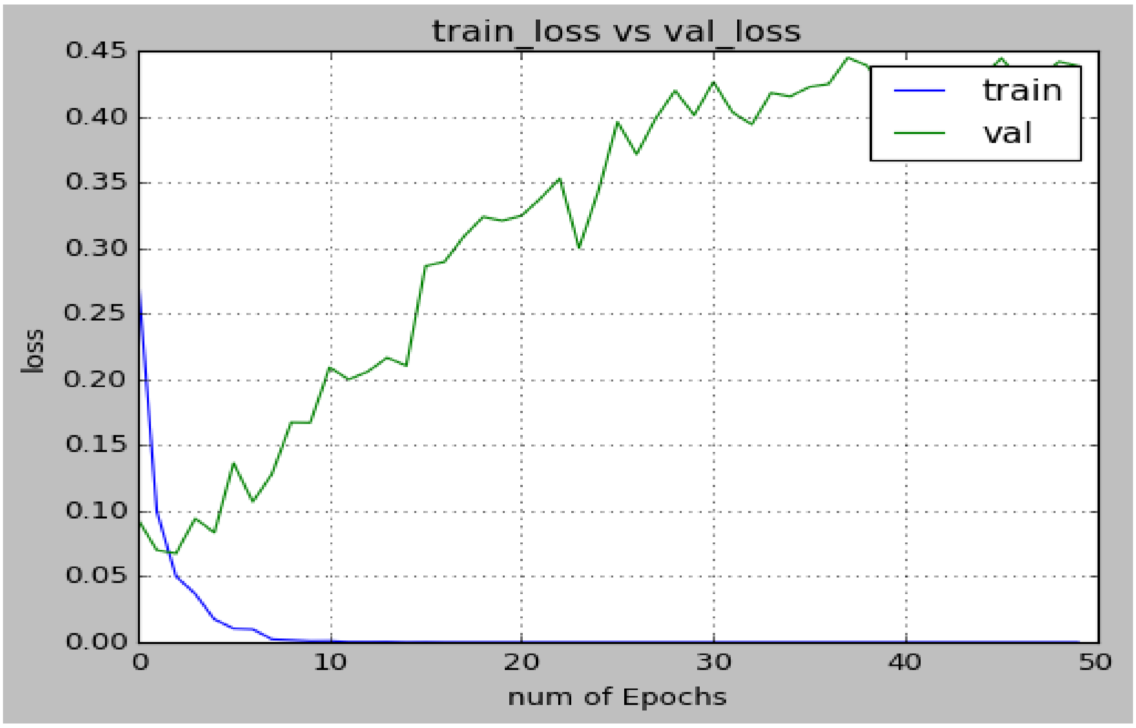


Image 3.7: Training vs Validity loss graph VGG16 Transfer learning for brain mri

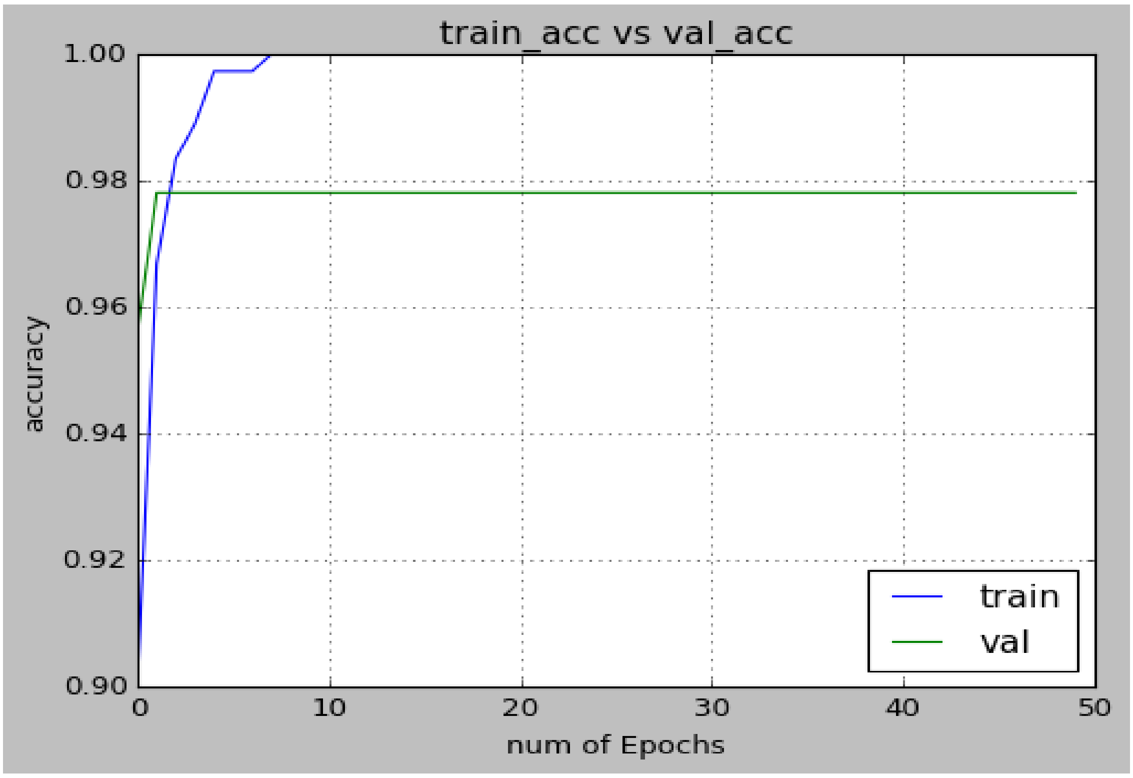


Image 3.8 : Training vs Validity Accuracy graph VGG16 Transfer learning for brain mri

To conclude the training, from graph it is clear that model gave very good training accuracy of 100% and validation accuracy is also almost 98% which is very great for any classification model

4. Object Detection

Object detection is a Computer Vision problem that involves recognizing and locating of objects in an image with specific class. Object localization can be interpreted in a variety of ways, the most common and popular way is by drawing a bounding box around the object. Image classification and object localization are combined in object detection algorithms. It takes an image as input and generates one or more bounding boxes, each with a class label attached. These algorithms can do multi-class classification and localization, as well as can detect multiple occurrences of objects in single image. There is different challenge in object detection such as it is unable detect accurate area of an object, perimeter of an object from image.

Before Deep Learning, object detection was a multi-step method that began with edge detection and feature extraction using techniques such as SIFT, HOG, and others. These images were then compared to actual object models, which were typically at multiple scale sizes, in order to detect and locate objects in the image. Traditional object detection is mainly three stage of process :

i) Informative region selection,
ii) Feature extraction,
iii) Object classification

The modern approach of deep learning is able to tackle some limitations of older approaches. The architectures are capable to learn very complex features like modern image classification techniques. From the study, I found there are two types of deep learning object detection models. One is **region proposal based** and another one is **regression-based.** Given below Image 4.1 shows the types of different object detection algorithm.

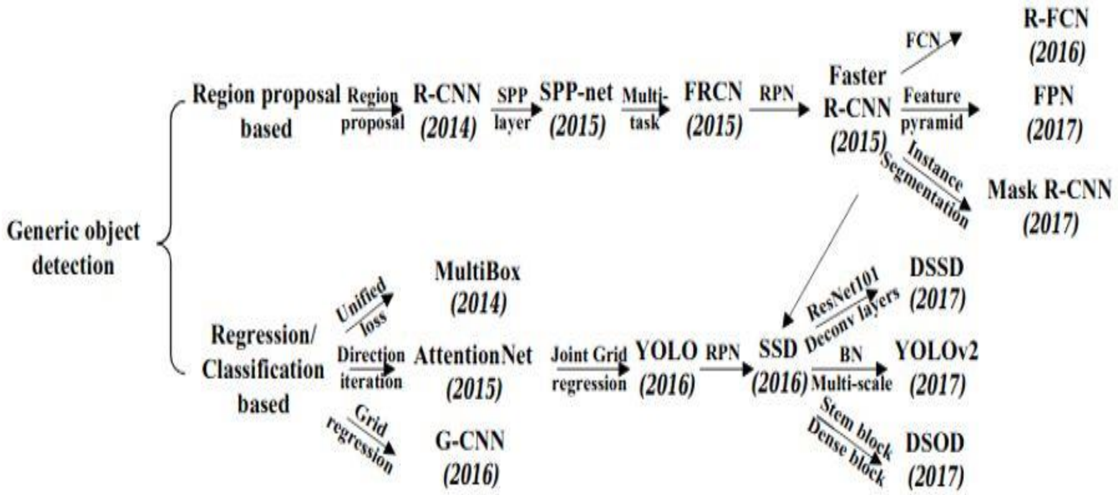


Image 4.1 *https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/*

Following is short overview of some object detection algorithm

4.1.The Region proposal-based framework

1. R-CNN

Ross Girshick proposed R-CNN in 2014. It obtained a mean average precision of 53.3% with more than 30% improvement over the previous best result on PASCAL VOC 2012. R-CNN work on selective search to extract just 2000 regions from the image which are called region proposals. These 2000 region proposals are fit into a square and fed to a CNN which produces a 4096-dimensional feature vector as output. The convolutional neural network acts as a feature extractor and the final output (dense) layer have the features extracted from the image which are fed into a Support Vector Machine to classify and find the object within that candidate region proposal. Moreover, algorithm also gave value of four coordinate to draw a bounding box on object [5].

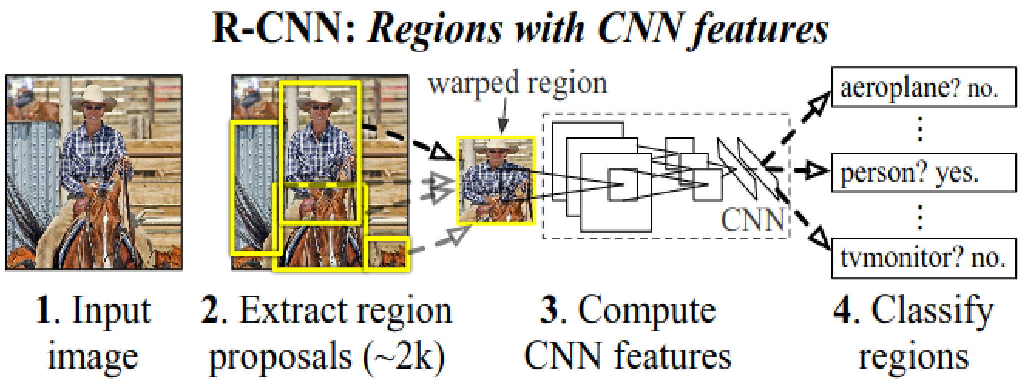


Image 4.1.1 : R-CNN Architecture
https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/

Limitations of R-CNN

- Training is multi-stage process
- Training is very expensive as it stores extracted feature on disk
- Time consuming as it uses selective search which has very high time complexity

2. Fast R-CNN

R-CNN model was very successful but still with some limitation. Ross Girshick, then, proposed an addition to overcome the speed issues of R-CNN in a 2015 paper of Fast R-CNN. Before Fast R-CNN, one more technique was proposed to speed up called spatial pyramid pooling networks, or SPPnets. In Fast R-CNN, first the entire image is processed with standard convolution architecture (such as VGG16) and develop feature map. This process is very similar to SPP-Net. Later on, a fixed-length feature vector is extracted from each region proposal with a region of interest (RoI) pooling layer. After that, all feature vector given to FC layers individually, before feeding into two output layers. Regarding output layer, one is responsible for producing probabilities for all possible C+ 1 categories. While other one encodes bounding-box positions with both value of X and Y [6].

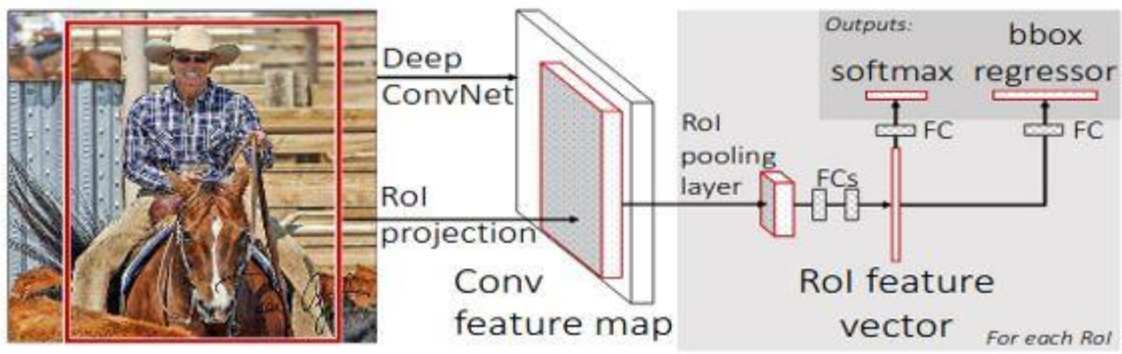


Image 4.1.2 : Architecture Fast R-CNN
https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/

Fast R-CNN can save expense by using single stage process via multi-task loss except for region proposal generation. On other hand It also use selective search method and again it is time consuming which make network slow.

3. Faster R-CNN

To overcome the issue with Fast R-CNN, Ren et al. introduced one more Region Proposal Network. In this, instead of selective search it uses a whole separate network to predict the region proposals. It became a nearly cost-free by sharing full-image convolutional features with detection networks. Region Proposal Network is achieved with a fully convolutional network. This fully convolutional network has can predict object bounds and scores at each position simultaneously. Faster R-CNN is single network but made of two modules. Module: i) **Region Proposal Network** and ii) **Fast R-CNN.** Both operate on the same output of a deep CNN. The RPN acts as an attention mechanism for the Fast R-CNN network. It informs the second network that where to pay attention.

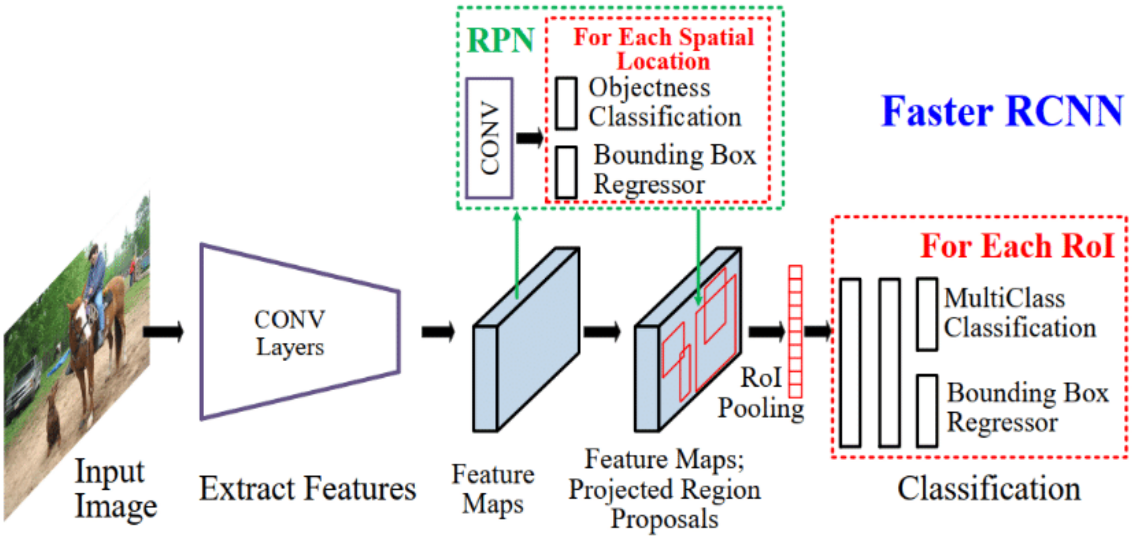


Image 4.1.3 : Architecture Faster R-CNN, https://analyticsindiamag.com/top-8-algorithms-for-object-detection/

4.2. Regression Based Model

1. SSD - Single Shot Detector

Single Shot MultiBox Detector (SSD) was proposed by Liu et al. It is inspired from the anchors adopted in MultiBox, RPN, and multi-scale representation. SSD mainly divided in two components: a **backbone** model and **SSD head. Backbone** model is usually a pre-trained classification network which act as a feature extractor. This can be mostly a network like ResNet which trained on large ImageNet dataset and the fully connected (classification) layer has been removed. So, network extract feature from the image while preserving the spatial structure of the image at a lower resolution. The head is convolutional layers in addition to backbone. The final layers activations give the outputs as the bounding boxes and classes of objects. Single Shot Detector is fast in detection as well as training but has the accuracy problem.

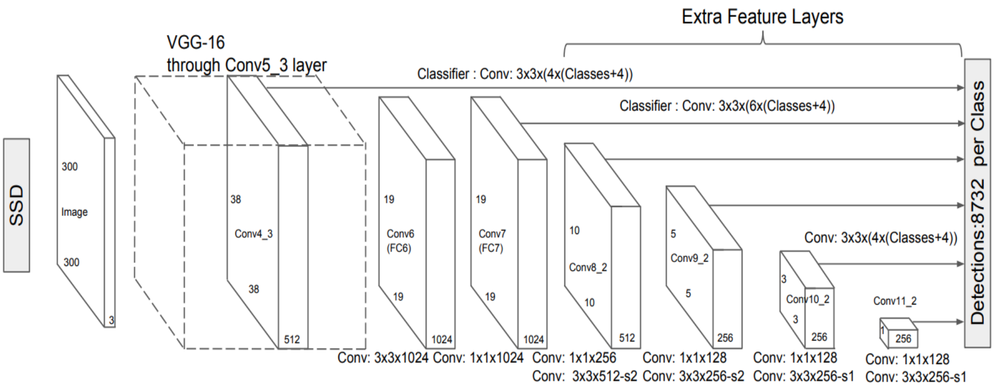


Image 4.2.1 : SSD Architecture, https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/

Amongst both type of Region proposal based network and Classification based network I choose to go with classification based network for my project of brain tumor object detection in brain mri images. I selected two best of state-of-art, EfficientDet by Google ai and YOLO (You Only Look Once). The detail description of the model is given in the part 5 Implementation

5. Implementation

5.1. EfficientDet

EfficientDet model is created by Google Brain team in July 2020. In object detection tasks, it achieved the highest accuracy with the fewest training epochs. With minimal computing capacity, this architecture outperforms YOLO and AmoebeaNet. **EfficientDet** is an advance model of **EfficientNet.** EfficientNet was state of art object detection model in 2019 which achived 84.4% more accuracy**.** EfficientDet use the ImageNet pre-trained EfficientNet as the network backbone. it outperformed other model already such as the RetinaNet, Mask R-CNN, and YOLOv3 architecture. They proposed some new optimization techniques such as i) BiFPN ii) New computational scaling techniques. EfficcientDet runs 5x faster on CPU and 2x-4x faster on GPU compare to another model. Following is the architecture of EfficientDet. The two challenges solved by EfficientDet is:

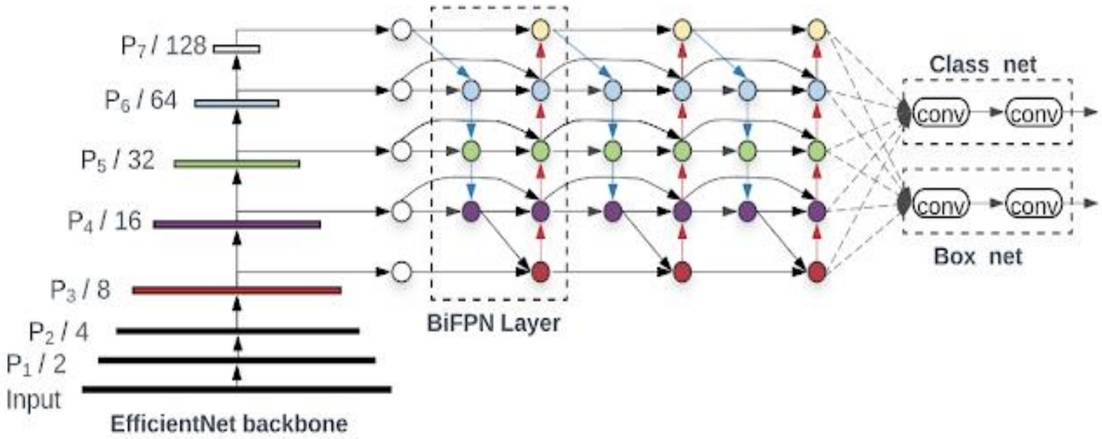


Image 5.1.1 : Architecture EfficientDet - https://ai.googleblog.com/2020/04/efficientdet-towards-scalable-and.html

Efficient multi-scale feature fusion

FPN (Feature Pyramid Networks) has become the major factor for fusing multi-scale features. The majority of the fusion techniques used in these networks overlook the importance of filters when fusing. They encapsulate them without making any distinctions. Not all features, intuitively, contribute equally to the production features. As a result, a more effective multi-scale fusion strategy is needed.

BiFPN

BiFPN refers to a Bi-directional Feature Pyramid Network. BiFPN was inspired by Feature Pyramid Network (FPN) in which information inherits only in one way direction. Apart from FPN it also inspired from PANet/NAS-FPN. BiFPN implements the multi-level feature fusion idea. which allows information to flow in both the top-down and bottom-up directions while maintaining normal and efficient connections.

$$W_{bifpn} = 64 \cdot \left(1.35^{\phi}\right), \qquad D_{bifpn} = 3 + \phi$$

BiFPN network

Architecture of BiFPN

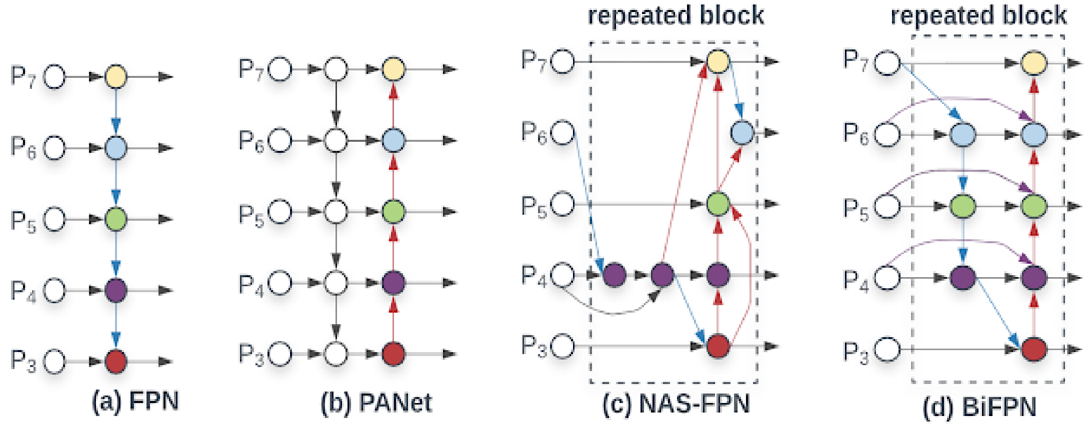


Image 5.1.2 : Comparison of BiFPN and other feature networks.
https://ai.googleblog.com/2020/04/efficientdet-towards-scalable-and.html

Google AI team evaluate EfficientDet on the COCO dataset for object detection. It managed to EfficientDet-D7 achieves a mean average precision (mAP) of 52.2, which was higher than the prior state-of-the-art model by 1.5 points. Moreover, using 4x fewer parameters and 9.4x less computation. It was also 5x-11x faster on CPU than other detectors [13].

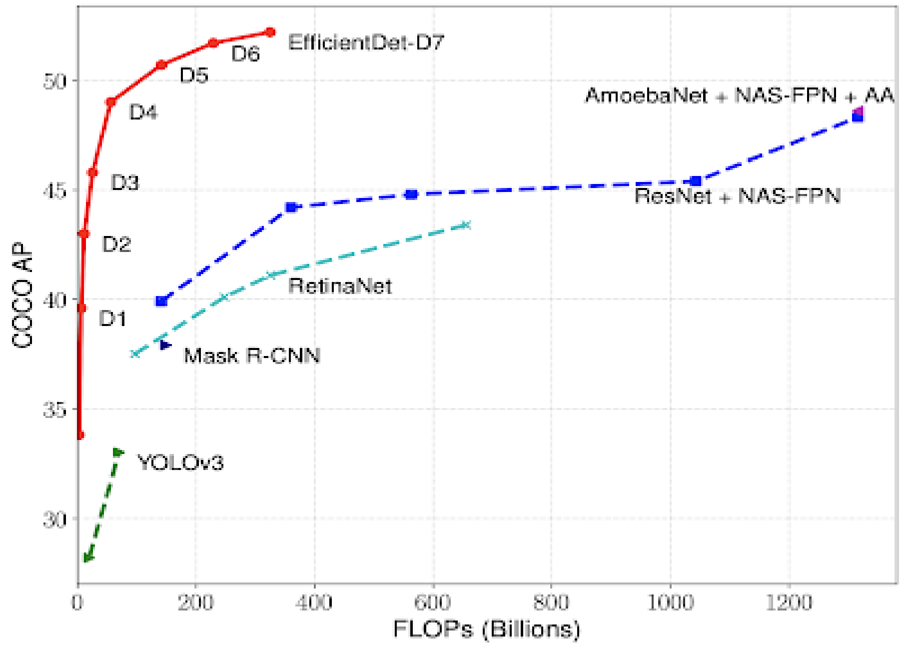


Image 5.1.3 : Model performance (EfficientDet).
https://ai.googleblog.com/2020/04/efficientdet-towards-scalable-and.html

Practical Implementation Details

Git repo :

https://github.com/Kishansinh94/ObjectDetection/tree/main/EfficientDet_Tumor-20210428T193402Z-001

Dataset : All annotated images in Pacal VOC format are available on github link given above

GPU : Tesla K80 (See below for more information)

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 465.19.01    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   34C    P8    28W / 149W |      0MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Training time: 8-hour apprx.

Epochs: 10,000

Graphs

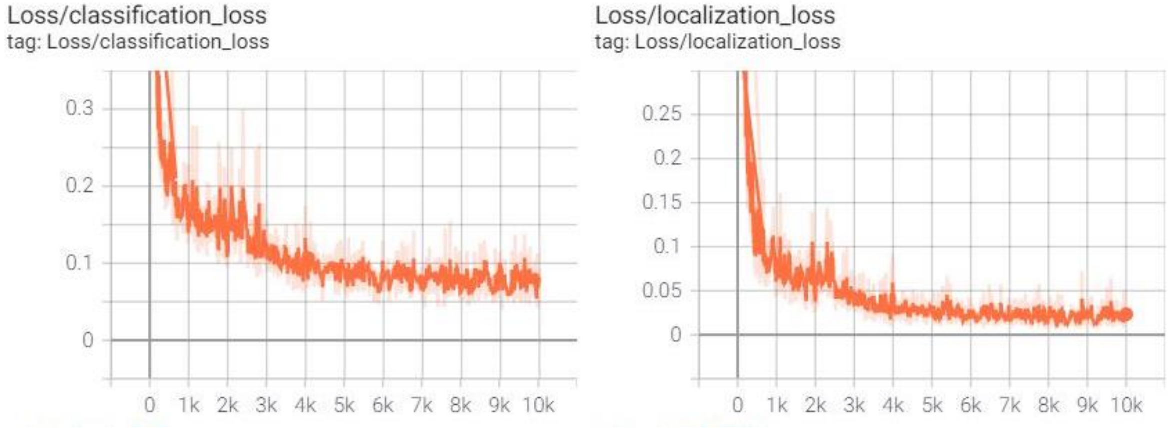


Image 5.1.4 : Classification and Localization loss

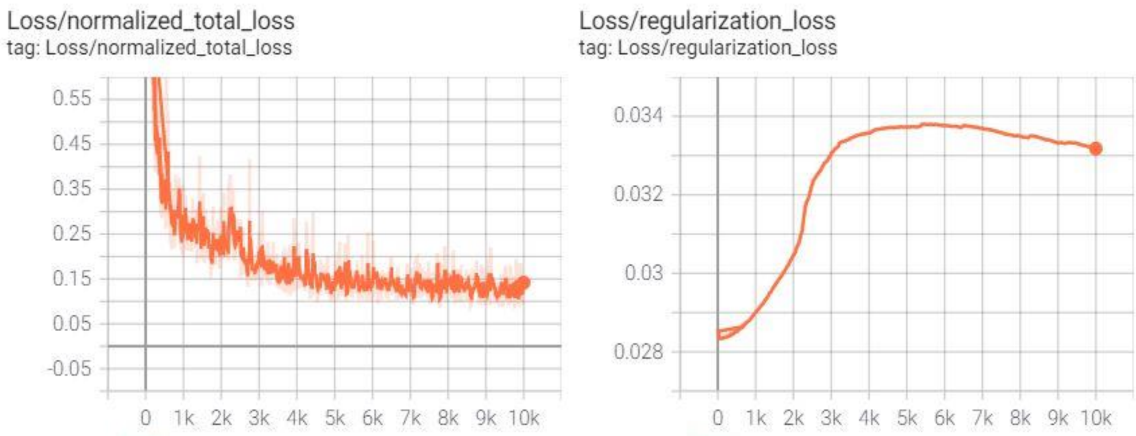


Image 5.1.5 : Normalized loss and regularization loss

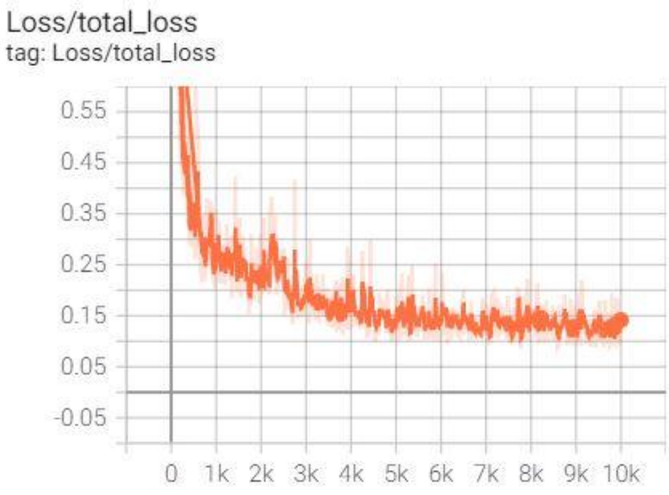


Image 5.1.6 : Graph Total loss

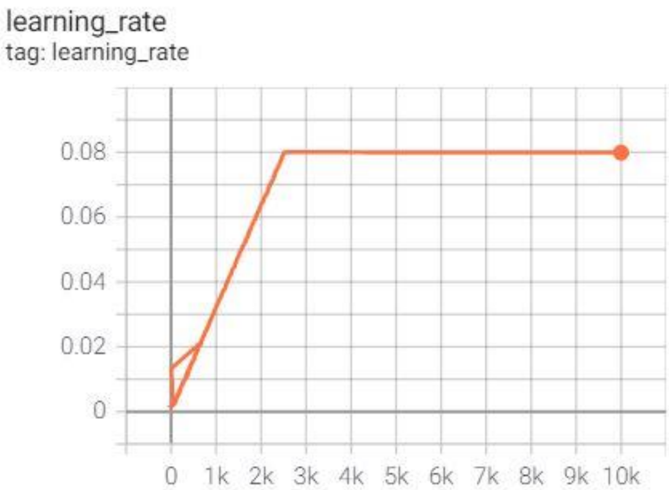
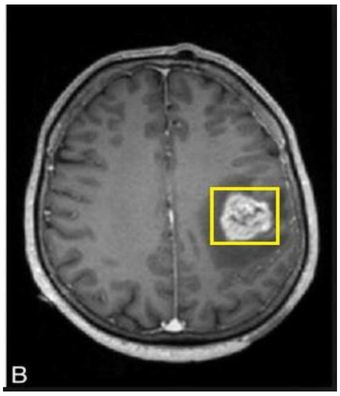
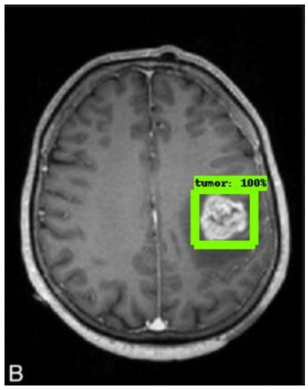
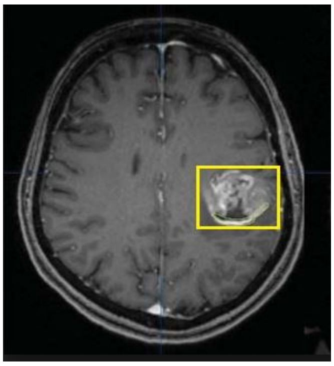
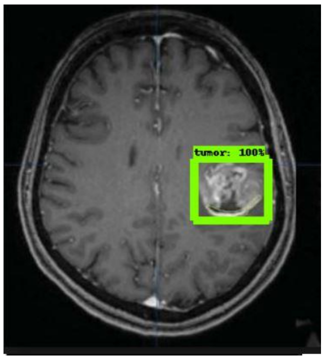
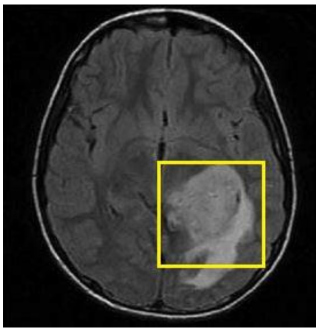
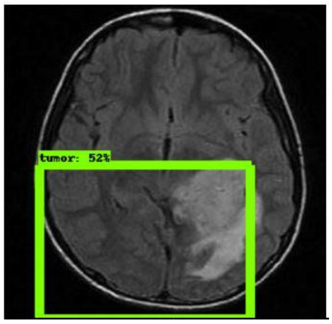
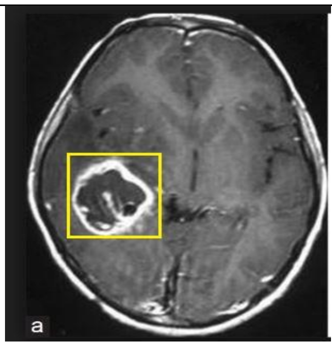
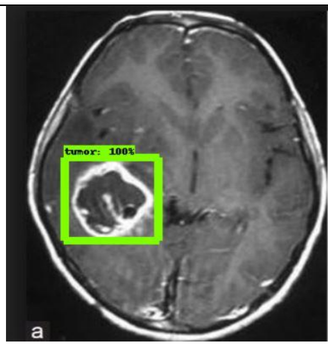


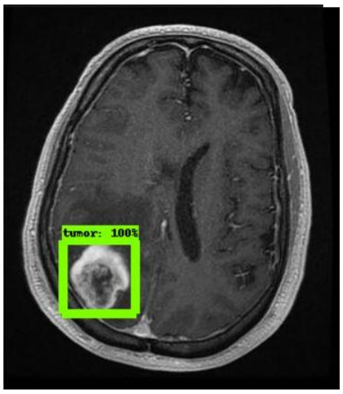
Image 5.1.7: Graph Learning rate

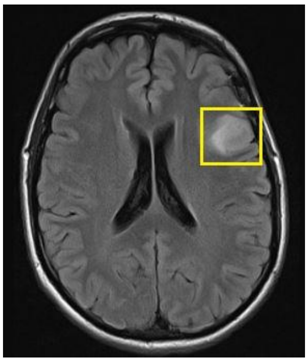
Table 5.1 : Ground Truth vs Prediction by Model

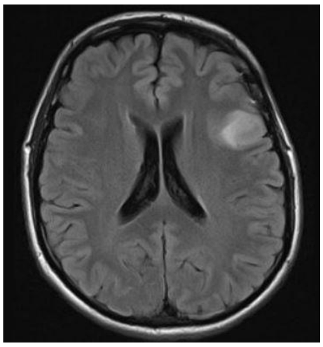
Ground truth	Predicted
	
	
	

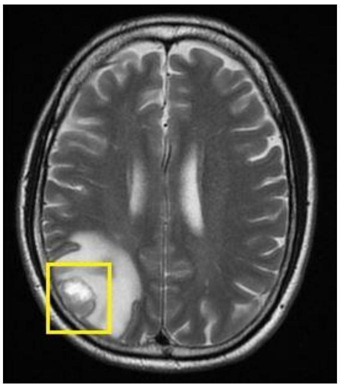

a

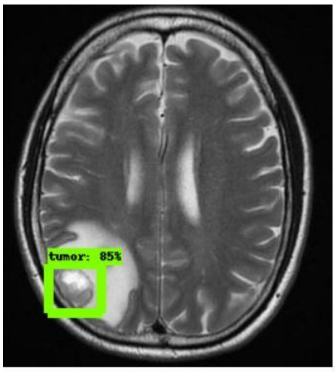

tumor: 100%
a


tumor: 100%








tumor: 85%

5.2. YOLO

YOLO is a state-of-the-art object detection algorithm that is incredibly fast and accurate. YOLO architecture is inspired by GooLeNet model for image classification. It introduced a new, simplified way to do simultaneous object detection in images. It uses a single CNN operating directly on the image and outputting bounding boxes and class probabilities. It incorporates several elements from the above networks, including inception modules and pretraining a smaller version of the network. It's fast enough to enable real-time processing. YOLO makes it easy to trade accuracy for speed by reducing the model size. YOLOv3-tiny was able to process images at over 200 frames per second on a standard benchmark data set, while still producing reasonable predictions [**Error! Reference source not found.**]. Yolo use totally different approach from the prior detection model. Yolo uses darknet as its backbone network. In output prediction bounding box may be appeared to be larger than the grid. Yolov1 can detect max of 49 objects in an image because it use 7x7 stride.

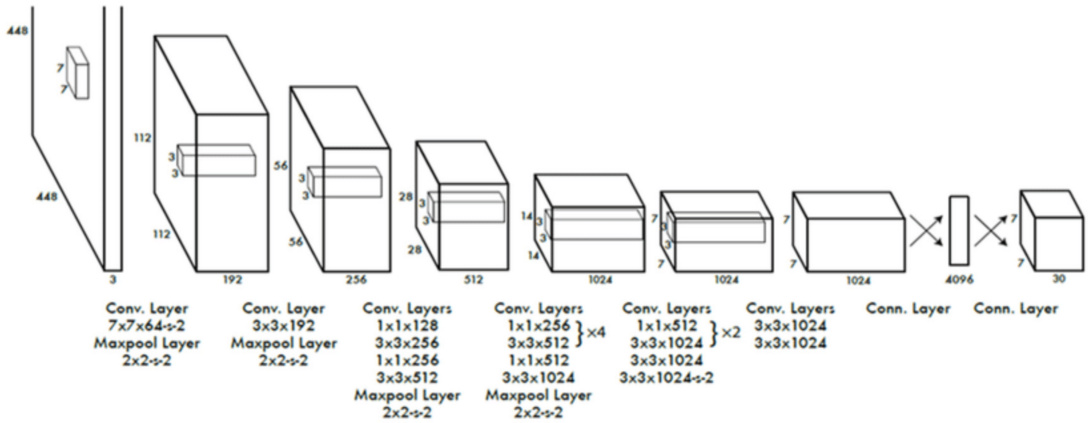


Image 5.2.1 : YOLO architecture, [*https://www.researchgate.net/figure/YOLO-architecture-YOLO-architecture-is-inspired-by-GooLeNet-model-for-image_fig2_329038564*](https://www.researchgate.net/figure/YOLO-architecture-YOLO-architecture-is-inspired-by-GooLeNet-model-for-image_fig2_329038564)

What is Darknet?

Darknet is as open-source neural network framework. It is written in C and CUDA. For prediction Darknet load information from its config file and its weight then it classifies input image and give output of top 10 predicted classes for the image. Below is the table of darkent-53 which is used in Yolo3

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

YOLOv3

Yolov2 has been improved and renamed YOLO v3 to reflect the gradual improvements. The competition is all about how reliable and easily objects are identified, as many object detection algorithms have been around for a while. YOLO v3 has all we need for real-time object detection and correct classification. This was dubbed an incremental development by the authors of Yolo. Here are some advancement

- **Class Predictions**
  Instead of softmax, which was used in YOLO v2, logistic classifiers are used for each class in YOLO v3. We can have multi-label classification in YOLO v3 by doing so.

- **Darknet-53**
  YOLO v2 used Darknet-19 as feature extractor and YOLO v3 uses the Darknet-53. It has 53 convolutional layers. It is more deeper compare to YOLO v2. It formed mainly of 3x3 and 1x1 filters and it also with shortcut connections.

- **Feature Pyramid Networks (FPN)**
  YOLOv3 use similar techniques for prediction which is used by FPN in which 3 predictions are being carried out for every location. After that features are extracted from each prediction. By using this, it improves the ability at different scale.

Implementation on Google colab

Git repo : https://github.com/Kishansinh94/ObjectDetection

GPU : Tesla K80

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 465.19.01    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   34C    P8    28W / 149W |      0MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Training graph

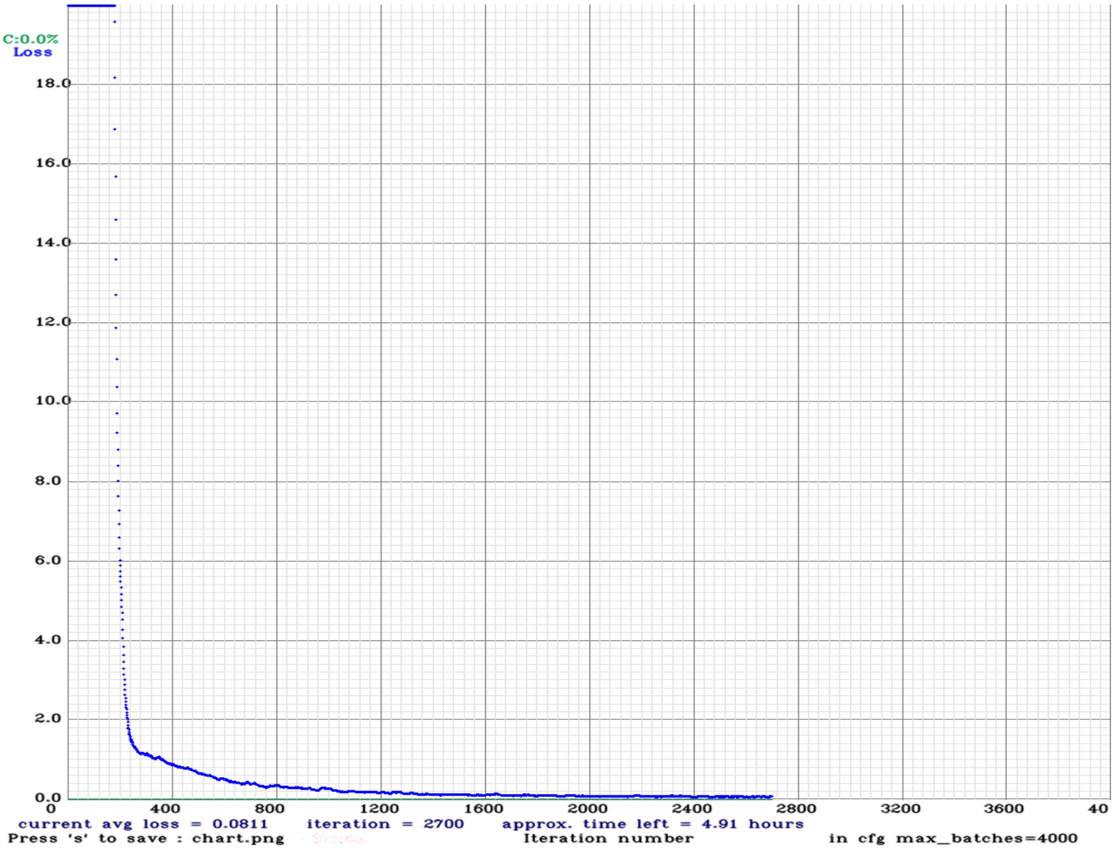
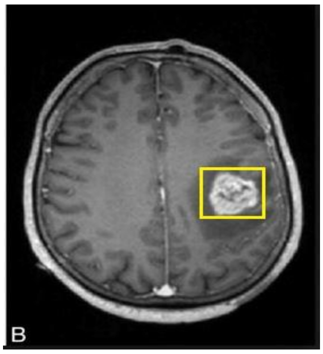
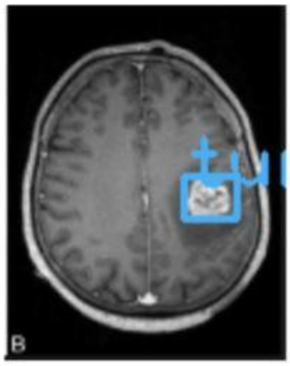
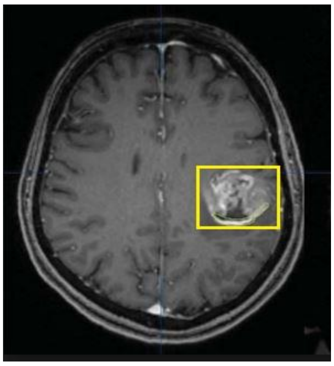
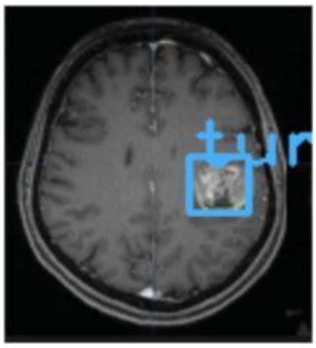
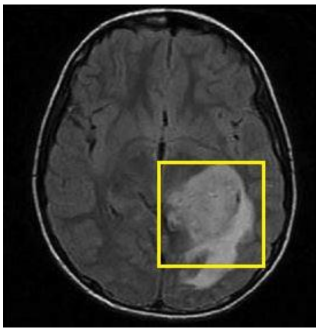
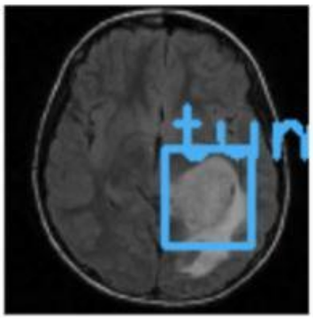
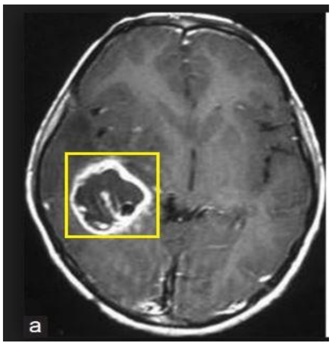
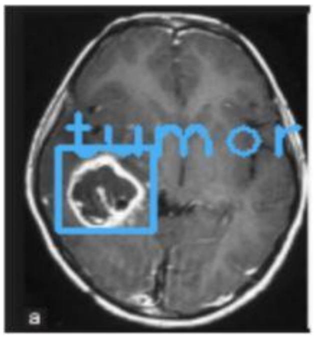
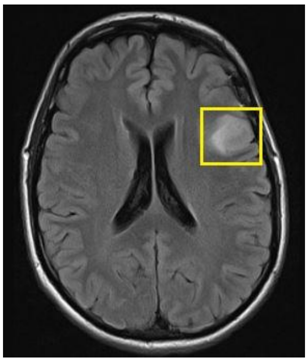
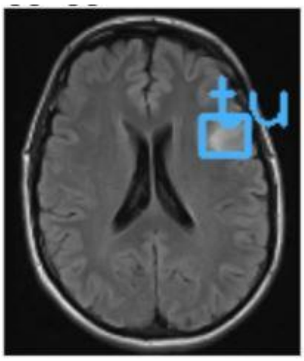
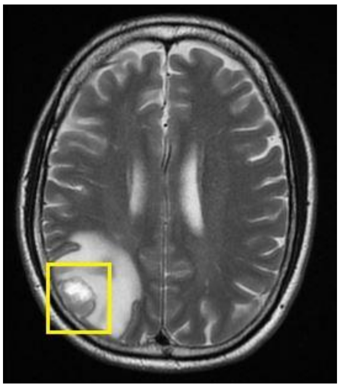
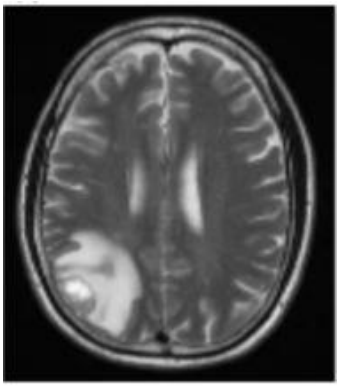


Image 5.2.2 : Graph YOLOv3 training

Table 5.2 : Result, Ground Truth vs Prediction by YOLOv3

Ground truth	Predicted
	
	
	
	

	 tumo
	 tu
	

6. Conclusion

AI is now changing the medical industry by assisting hospitals in better organizing their workflow, making diagnostics and decision-making easier for physicians, and providing useful lifestyle changes for patients. Moreover, in recent time AI has more contribution in disease classification and detection using deep learning.

The current trending thing in deep learning is object detection which have many applications from autonomous driving to disease detection. So here, I carried out experiment for object detection in medical Images which brain MRI images. Experiment is about tumor detection in brain MRI images. I used two state of art techniques and conclusion of experiment is as below:

Google brain team developed the most powerful and efficient object detection algorithm so far. On other hand, YOLO family is also one of the state-of-art for object detection, it is known for its fast processing and less resources. After experiment I found, Yolo is slow in training but has very good accuracy in color image. When it comes for medical image then it also performed wonderful with just around of 150 training images and training time of aprx 8 hours around 2500 epochs. On other hand the Google AI team claimed EfficientDet as one of the best object detection. No doubt it performed wonderful on color images. In this experiment, I found it slight less accurate with same training time. I trained the EfficientDet D0 with same dataset for around of 8 hours but in that time, it was able to complete 10000 training epochs which shows good training speed and but performance was compromised bit. Out of seven unseen image YOLOv3 correctly detected object for six images where EfficientDet was able to detect in six and out of six two are false detection. The detection speed and and accuracy are unknown on real time feed and video data.

Both models are very good and there are many advancements in the version of both techniques but for particular this experiment YOLOv3 performed slight better then EfficientDet. However, there are lots of hyperparameter in both model which can improve the performance.

7. References

1. Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016.
2. https://pjreddie.com/darknet/yolo/
3. https://www.thinkautomation.com/bots-and-ai/what-is-moravecs-paradox-and-what-does-it-mean-for-modern-ai/
4. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
5. Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
6. Girshick, Ross. "Fast r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448. 2015.
7. Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "Efficientdet: Scalable and efficient object detection." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781-10790. 2020.
8. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).
9. Dong, Hao, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. "Automatic brain tumor detection and segmentation using u-net based fully convolutional networks." In *annual conference on medical image understanding and analysis*, pp. 506-517. Springer, Cham, 2017.
10. Pan, Sinno Jialin and Qiang Yang. “A Survey on Transfer Learning.” *IEEE Transactions on Knowledge and Data Engineering* 22 (2010): 1345-1359.
11. Hon, Marcia, and Naimul Mefraz Khan. "Towards Alzheimer's disease classification through transfer learning." In *2017 IEEE International conference on bioinformatics and biomedicine (BIBM)*, pp. 1166-1169. IEEE, 2017.
12. https://machinelearningmastery.com/transfer-learning-for-deep-learning/
13. https://ai.googleblog.com/2020/04/efficientdet-towards-scalable-and.html
14. https://www.cellstrat.com/2020/03/27/object-detection-with-efficientnet-and-efficientdet/
15. https://luminovo.ai/blog/ai-in-medical-imaging
16. https://roboticsandautomationnews.com/2020/03/09/how-ai-technologies-accelerate-progress-in-medical-diagnosis/31184/#:~:text=AI%2Ddriven%20software%20can%20be,processing%20photos%20of%20skin%20lesions.
17. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International Conference on Machine Learning*, pp. 6105-6114. PMLR, 2019.