IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Brain Tumor Screening and Identification Using Dectotron2 and RetinaNet Model

Sumit Khatua
Research Scholar
Computer Science &Engineering
Chandigarh University
Mohali, Punjab-140413, INDIA

Prashant Kumar
Research Scholar
Computer Science & Engineering
Chandigarh University
Mohali, Punjab-140413, INDIA

Naveen Chander
Professor
Computer Science &Engineering
Chandigarh University
Mohali, Punjab-140413, INDIA

***Abstract*—In the 21st century, the field of medical science has reached a remarkable level of maturity, demanding the highest precision in diagnosing and treating brain tumor. The integration of cutting-edge computer technology, specifically artificial intelligence (AI) tools, has emerged as a critical resource in enhancing brain tumor classification, particularly gliomas, for prognosis and treatment planning. This research represents the culmination of extensive efforts in deep learning, with the primary objective of identifying and characterizing brain tumors with unparalleled precision. Previous studies have explored various iterations of the Detectron2 based on open-source project and it introduce the second-generation library which enhance classification within images and videos have formed the basis of our quest to enhance tumor detection and classification detection.**

Machine learning, a dynamic field, offers an elegant solution to elevate brain tumor identification and labelling accuracy. The integration of YOLOv8-Dectotron2 represents a remarkable advancement, achieving an 95.887% detection accuracy rate, a significant improvement of 5.7% compared to prior work.

Keywords—Brain Tumor, MRI, Detctron2, Yolo, Streamlit, RetinaNET

I. INTRODUCTION

Brain tumors have emerged as a growing health concern in India, particularly among children. The prevalence of brain tumors has markedly increased in recent years., with genetic alterations being identified as a significant contributing factor. Age-adjusted cancer incidence rates have revealed a range for boys, from 18.6 to 159.6 per million, and for girls, from 11.3 to 112.4 per million. Alarmingly, research indicates that tumors are not entirely the same as those found later in life. tumors mutate and provide changes the their structure which later cause thread to treat. However, recognizing brain cancers in the human brain is the most difficult component. As a result, we need current tools and diagnostic techniques that use computerized diagnostic models to better recognize, categorize and assess brain tumors. While the incidence rates highlight the seriousness of the situation, it is important to recognize that India has historically collected less complete data on cancer cases than many other countries. This deficiency in data availability underscores the need for a concerted effort to address the burgeoning issue of brain tumors in the country. The burden of brain tumors, especially among children, presents a multifaceted challenge that transcends medical boundaries. As generations pass, this health concern threatens to become increasingly common, necessitating a proactive and innovative approach.

In response to this pressing need, our research introduces a groundbreaking solution harnessing the power of artificial intelligence (AI) through the RetinaNet-Dectotron2 model. This paper not only traces the evolution of brain tumor detection technologies but also presents Dectotron2 as a transformative tool in this domain. The fusion of advanced AI algorithms and precise detection mechanisms holds the promise of significantly enhancing diagnostic accuracy. Moreover, it empowers healthcare professionals to customize treatment strategies, ultimately leading to improved patient outcomes.

Our research shines a light on the significance of this breakthrough within the context of the escalating prevalence of brain tumors in India. We underscore the urgency of addressing this critical health issue and emphasize the vital role played by Dectotron2 in providing faster, more accurate, and more concise predictions of cancer cells in the brain. With this modern technique, we achieve an impressive 85.887% detection accuracy rate, marking a substantial improvement of 5.7% compared to previous methodologies. Our prediction technique signifies a significant advancement beyond traditional algorithms. These technological modifications not only enhance the prediction level but also streamline and expedite the process of detecting cancer in the human brain, thereby facilitating faster and more effective treatments. Key Features of YOLOv8-Dectotron2:

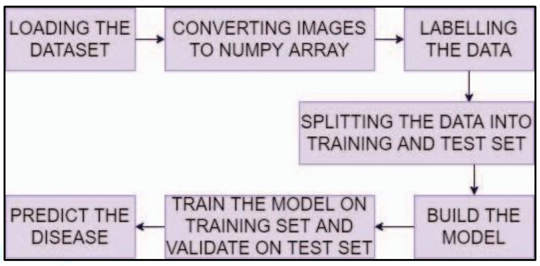
- Speed: This algorithm offers real time object prediction, significantly accelerating the detection process.
- Learning Capabilities: The extraordinary learning abilities of Dectotron2 enable it to recognize object representations and make use of them for accurate object detection.

II. Related Work

Deep learning provide wide improvement in medical field various research has been done to improve the medical area by introducing the fundamental of computer science. The integration of deep learning techniques [1], most notably convolutional neural networks (CNNs), has emerged as a pivotal development in brain tumor detection. Research efforts have explored various CNN architectures, with the MobileNet model, in particular, showcasing promising outcomes in improving the precision of MRI-based tumor identification. One approach to automating brain tumor classification involves the application of supervised learning techniques alongside feedforward back propagation neural networks[2] This automated methodology has garnered attention for its ability to effectively categorize MRI pictures into malignant and non - malignant types, marking a substantial advancement in the automated diagnosis of brain pathologies.[3] Laddha's (2014) work has significantly contributed to the field through a focus on image segmentation and thresholding operations for tumor detection in MRI scans. This research employs morphological operators and wavelet-based algorithms, demonstrating the potential for improving diagnostic accuracy through meticulous image analysis.[4] Amsaveni's (2013) research introduces a comprehensive methodology encompassing preprocessing, feature extraction utilizing Gabor filters, and classification employing neural network techniques. This holistic approach underscores the importance of advanced feature extraction methods to enhance the precision of brain tumor detection.[5] Kumar's recent work in 2021 emphasizes the pivotal role of convolutional neural networks (CNNs) in the context of brain tumor detection. Kumar's findings underscore the critical significance of image restoration and enhancement techniques as essential components in ensuring precise and reliable diagnoses.

III. Proposed Mathology

A. Working Diagram



LOADING THE DATASET
CONVERTING IMAGES TO NUMPY ARRAY
LABELLING THE DATA
SPLITTING THE DATA INTO TRAINING AND TEST SET
BUILD THE MODEL
TRAIN THE MODEL ON TRAINING SET AND VALIDATE ON TEST SET
PREDICT THE DISEASE

Fig. 1: Block Diagram of proposed scheme.

Detectron2 represents a significant evolution from its predecessor, Detectron, as it has been completely rewritten and implemented in PyTorch. Its core strengths lie in its modular design, which offers greater flexibility, and it boasts significantly improved prediction accuracy. Moreover, Detectron2 is optimized for swift training on a wide range of GPU servers. We have adopted this advanced algorithm to enhance the flexibility and quality of our data prediction outputs, thereby augmenting our accuracy rates. Its user-friendly interface makes it easy to work with, and it encourages open-source contributions while simplifying implementation without the need to fork the entire codebase. In essence, Detectron2 empowers us to leverage cutting-edge object detection capabilities while streamlining our development process and fostering collaborative innovation.
The brain tumour detection result efficiency gets increased using detectron2 algorithm then the other detection modules with more proficient and accurate output.

B. PyTorch

Originally, Detectron employed Caffe2 as its framework. However, the transition to PyTorch has introduced a more efficient programming model, offering researchers greater flexibility for model design and experimentation.

Detectron2's compatibility with PyTorch aligns perfectly with our deep learning research objectives in object detection. This integration not only enhances our capacity for model development and refinement but also accelerates our capacity for innovation in this domain, leading to advancements in object detection technologies.

C. Design of Detectron2

By combining existing models from the first Detectron, such as Faster R- CNN, Mask R- CNN, RetinaNet, and DensePose, Detectron2 expand on its core functionality. Additionally, it introduces additional models such as TensorMask, Panoptic FPN, and

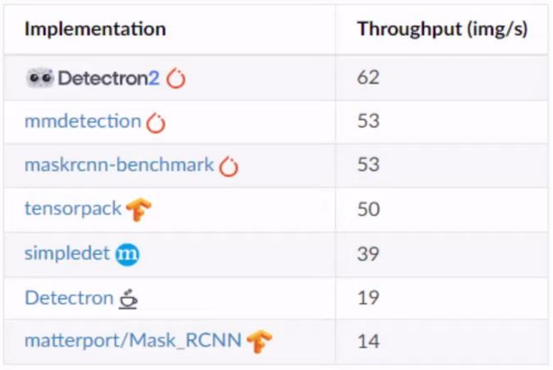
Cascade R- CNN as part of a continuous effort to strengthen its algorithmic foundation.. This streamlined interpretation also includes precious advancements like coetaneous Batch Norm and extends support to new datasets like LVIS, farther perfecting its capabilities for advanced computer vision exploration.

D. YOLO

YOLO is a technique that uses neural networks to detect objects in real-time. This algorithm's popularity is a result of how fast and accurate it is. It has been used in many different contexts to identify people, animals, parking meters, and traffic lights. The YOLO algorithm is significant for the reasons listed below:

- Speed: This approach boosts detection speed by redicting objects in real-time.
- High accuracy: The YOLO prediction approach produces accurate results with little background errors.
- Learning abilities: The algorithm has excellent learning abilities that allow it to pick up on object representations and use them for object detection.

Residual blocks, bounding box regression, and intersection over union (IOU) are the three methods used by Yolo. For a proper diagnosis of a brain tumor, early detection and treatment planning are crucial. Digital image processing must be carefully considered while analyzing medical images. Segmenting a brain tumor involves dividing it into defective and healthy brain components. The prognosis for many patients is improved by early identification of brain tumors, which lessens the impact of surgery and treatment. The early identification of brain tumors is vital since they may affect cognitive function whether or not they are tumors if they grow to the point where they press against neighboring tissues.



Implementation	Throughput (img/s)
Detectron2	62
mmdetection	53
maskrcnn-benchmark	53
tensorpack	50
simpledet	39
Detectron	19
matterport/Mask_RCNN	14

Fig. 2: Comparison of different model.

E. Design of webapp through Streamlit

Streamlit, an open-source Python framework, makes it simple to create and distribute beautiful, personalized web apps for data science and machine learning. It takes only a few minutes to design and deploy effective data apps. Built on top of Python, Streamlit makes use of the well-liked Flask web framework. This makes it simple to start using Streamlit, even if you have no prior web development knowledge. All you need to do to develop a Streamlit app is write a Python script. You can incorporate widgets, charts, and other interactive components into your programme using Streamlit methods in your script.

F. Dataset Collection

Data is the most important aspect of this project. We need a large number of images because we work with photos. As a result, we must be aware of some critical factors such as image sizes, resolutions, image quality, and the syndrome of brain tumors. Initially, image data is gathered from Roboflow of Brain tumor and annotate themselves through the help of cvat.ai. CVAT.ai is an open annotation platform where we can annotate any custom dataset save it in yolo format. Almost 753 images were gathered and combined. However, not all of the combined images are as beneficial to us. Certain images have insufficient resolution, and some cannot be distinguished as affected or unaffected. As a result, 500 images for the training ambition. Similarly, 20% images are chosen per directory to testing ambition. This is why a data processing system must be completed in order to obtain our preferred modelling data.

G. Building the Model

Computer vision's object detection field is incredibly crucial because it is used in many other fields, combining video monitoring, applications in healthcare, autonomous driving, and many more. We will go over the specifics of our approach in this part, including the weighted feature pyramid network, octave convolution, and overall Retinanet network architecture. The need for Retinanet is for:

Facebook AI Research developed RetinaNet to address the dense detection issue. It was important to compensate for the inconsistencies and irregularities of single-shot detectors for objects like YOLO and SSD when confronting extreme foreground-background classes.

- *Retinanet:*

The feature extraction core network (ResNet), feature pyramid networks (FPN), and both regression and classification smaller networks make up the majority of RetinaNet. The picture features are initially retrieved by ResNet, then reorganized in FPN, and then delivered to the categorization and analysis sub-network in order to arrive at the final detection result.

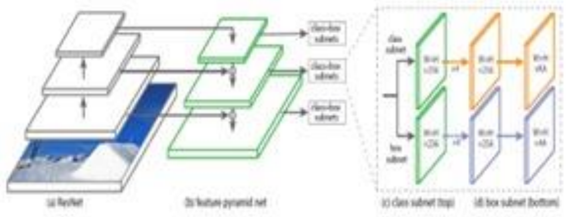


Fig. 3: Retina net architecture

- *OctaveConvmodule*

The image contains both extreme frequency and minimal frequency information. While the high frequency range part primarily gives specific information, the low frequency section frequently reflects the image's overall information. In the standard convolution structure, which works by altering the sliding window, the distribution of high frequency and low frequency is ignored. The solution proposed was for OctaveConv to act as the convolutional layer in a traditional CNN. The octave convolution module divides the characteristic map into an extremely high-frequency and a minimal-frequency channel, and the low-frequency channel's characteristic map is then reduced to half its initial size. Finally, an extremely high and a reduced frequency component are separated from this characteristic map.

Our design's octave convolution structure consists of an initial, transitional, and output layer.

First layer: The task of the first layer is to extract the frequent exposure characteristic map and a low-rate feature map from the input image (XH and XL in Fig 2). The input image is run through a convolutional layer with a 3 X 3 convolution kernel size to produce high-frequency feature maps. The similar process is used to create low-frequency feature maps, but an average pooling layer is added prior to the convolutional layer. The α parameter determines how many channels the high-frequency and low-frequency feature maps have. If the input channel count is given as channels, the low-frequency channel count is channels x α, and the high-frequency channel count is channels x (1- α). The value of in this paper is 0.25; the value of ranges from 0 to 0.5.

Transition layer: Reprocessing the feature map is done using the transition layer. Following a convolutional layer, XH and XL subsequently carry out up- and down-sampling operations, respectively. Finally, summing produces new YH and YL.

Output Layer: The function of the output layer is the polar opposite of that of the starting layer. After passing through a convolution layer, the output of the transition layer, YL, is upsampled to the same size as YH and added to the feature map that is produced after YH passes through a convolution layer. The octave convolution module's output feature map is then obtained.

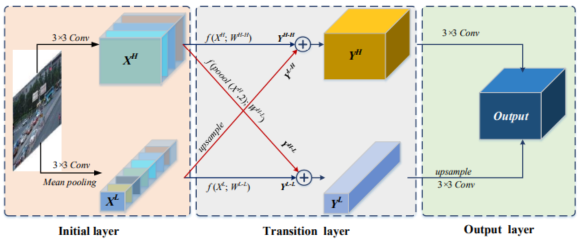


Fig. 4: The working architecture of octave-conv transition.

- Regression Object Subnetwork

Each feature map, within the FPN is linked to both the classification and regression subnetworks. The main difference between these subnetworks lies in the layer, which has a size of 3*3 with 4 filters. This layer generates an output feature map of size W*H*4A. In terms of design, it is this distinction that sets apart the classification and regression subnetworks. The role of the regression subnetwork is to produce four integers for each anchor box. These integers represent the offset, in terms of center coordinates, width and height between the anchor box and the ground truth box. This enables localization of class objects., which results in the final convolution layer having four filters. As a result, the output feature map of the regression subnet includes 4A channels or filters.

H. *Model Compilation*

RetinaNet's default predictor is a class that offers an easy way to use a trained RetinaNet model to make predictions. It is implemented in the package detectron2.

To utilise the default predictor, you must first construct an instance of it by providing the path to a RetinaNet model that has already been trained. The predict () method on the predictor can then be called with an image as a parameter. For each object that is recognised, the predictor will provide the following information in a dictionary of predictions:

- Box coordinates
- Class label
- Confidence score

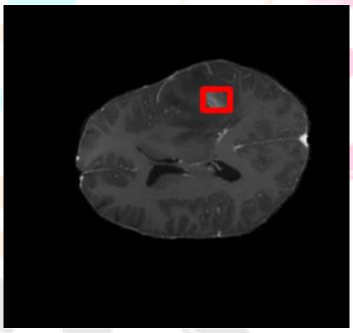
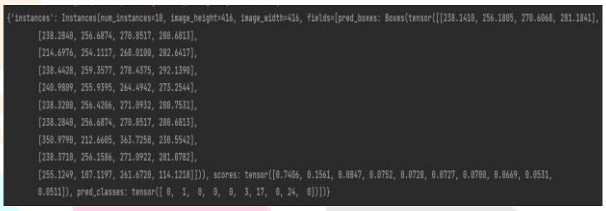


Fig 5: Tumor is shown by bounding boxes.



```
{'instances': Instances(num_instances=10, image_height=416, image_width=416, fields=[pred_boxes: Boxes(tensor([[238.1410, 256.1805, 270.6068, 281.1841],
        [238.2840, 256.6874, 270.8517, 280.6813],
        [214.6976, 254.1117, 268.0100, 282.6417],
        [238.4428, 259.3577, 278.4375, 292.1390],
        [240.9889, 255.9395, 264.4942, 273.2544],
        [238.3280, 256.4206, 271.0932, 280.7531],
        [238.2840, 256.6874, 270.8517, 280.6813],
        [350.9798, 212.6605, 363.7258, 230.5542],
        [238.3718, 256.1586, 271.0922, 281.0782],
        [255.1249, 107.1197, 261.6720, 114.1218]])), scores: tensor([0.7406, 0.1561, 0.0847, 0.0752, 0.0728, 0.0727, 0.0700, 0.0669, 0.0531,
        0.0511]), pred_classes: tensor([ 0,  1,  0,  0,  0,  3, 17,  0, 24,  0])])}
```

Fig 6: Box coordinates, class instances, and confidence score of the following image.

I. *Testing and Training*

To train the model, the function 'util.train()' is used. Validation data testing on the Dataset The number of epochs (6000) is specified in the util.train function, which rhythmically runs the system over the data. Following the completion of the training procedure, the testing procedure is configured. It validates the potential of the trained detectron model. Following is the code to train the model on the validation dataset.

```
util.train(args.output_dir,
           args.data_dir,
           args.class_list,
           device=args.device,
           learning_rate=float(args.learning_rate),
           batch_size=int(args.batch_size),
           iterations=int(args.iterations),
           checkpoint_period=int(args.checkpoint_period),
           model=args.model)
```

Fig. 7: The working code.

Python 3.10 was used for the entire execution, which was carried out on the Google Colaboratory and the resulting model was saved in Google Drive.

IV. RESULTS

After 4000 iterations on the testing data set, the verification efficiency is 82.86% and the validation loss is reduced.

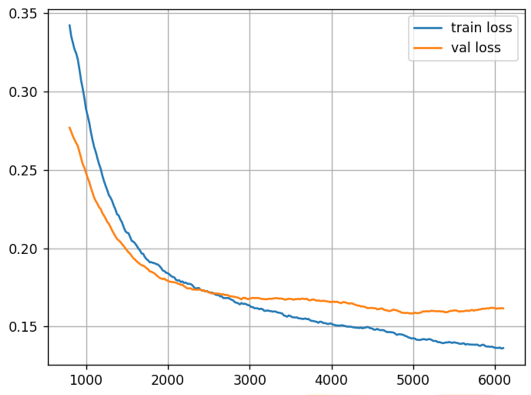


Fig. 8: Model loss

Figure 7 illustrates that when the model is applied to the validation set, a significant loss is produced, but when it is applied to the testing set, the loss slowly decreases as the number of iteration rises.

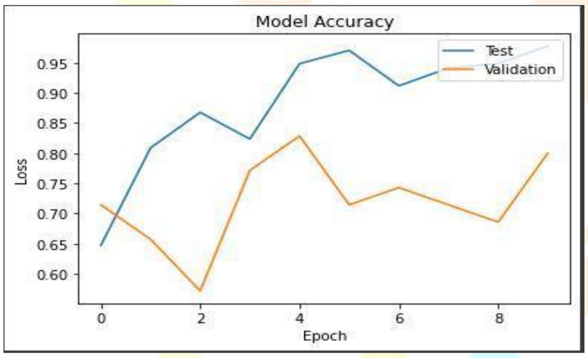


Fig. 9: Model precision

After being applied to the testing set, the convolutional neural network model's precision was 97.79%, with relatively minimal loss as the number of iteration rose. The difference in the model's precision between the training dataset and the validation dataset is shown in Figure 8.

V. CONCLUSION

The purpose of this study is to develop a model that can accurately identify brain cancers using MRI scans. The dataset, which comprises of 753 brain MRI pictures, was sufficient to evaluate the model's performance. The Detectron2 machine learning algorithm serves as the model's foundation. Without sacrificing any crucial information that will be utilised for prediction, it helps to simply reduce and resize the image. When used on the training set, the constructed model achieves an precision of 97.79%, and when used on the validation set, it achieves an precision of 82.86%. The loss progressively starts to go down as the number of epochs rises. When applied to the training set, the model loss is very low; however, when applied to the validation set, it is considerable. To further improve the general accuracy of this model, further datasets will be incorporated in the future.

REFERENCES

[1] N. M. Dipu, S. A. Shohan and K. M. A. Salam, "Deep Learning Based Brain Tumor Detection and Classification," 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498384.

[2] Dixit, A., Singh, P. (2023). Brain Tumor Detection Using Fine-Tuned YOLO Model with Transfer Learning. In: Gupta, M., Ghatak, S., Gupta, A., Mukherjee, A.L. (eds) Artificial Intelligence on Medical Data. Lecture Notes in Computational Vision and Biomechanics, vol 37. Springer, Singapore. https://doi.org/10.1007/978-981-19-0151-5_30.

[3] N. S. Kumar, A. K. Goel and S. Jayanthi, "A Scrupulous Approach to Perform Classification and Detection of Fetal Brain using Darknet YOLO v4," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021,pp.578-581,doi: 10.1109/ICACITE51222.2021.9404656.

[4] Kumar, N. S., & Goel, A. K. (2022). Detection, Localization and Classification of Fetal Brain Abnormalities using YOLO v4 Architecture. International Journal of Performability Engineering, 18(10).

[5] Hamamci, N. Kucuk, K. Karaman, K. Engin and G. Unal, "Tumor-cut: Segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications," IEEE Transactions on Medical Imaging, pp. 790-804, 2012.

[6] H. B. Menze, A. Jakab, S. Bauer, K. Farahani and K. Justin, "The Multimodal Brain Tumor Image Segmentation Benchmark," IEEE Transactions on Medical Imaging, pp. 1993-2024, 2014.

[7] Jin Liu, Min Li, Jianxin Wang, Fangxiang Wu, Tianming Liu and Yi Pan, "A survey of MRI-based brain tumor segmentation methods," Tsinghua Science and Technology, pp. 578-595, 2014.