Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1625~1632
e-ISSN: 2548-6861

Detection and Localization of Brain Tumors on MRI Images Using the YOLO Algorithm

Zaky Indra Bayu Satria [1]*, Catur Supriyanto [2]*
* Faculty of Computer Science, Universitas Dian Nuswantoro
111202012905@mhs.dinus.ac.id[1], catur.supriyanto@dsn.dinus.ac.id[2]

Article Info

Article history:

Received 2025-07-02
Revised 2025-07-12
Accepted 2025-07-19

Keyword:

Brain Tumor Detection,
Deep Learning,
Object Detection,
MRI,
YOLO.

ABSTRACT

This study addresses the critical need for early and accurate brain tumor diagnosis on MRI images by comparing five versions of the YOLO algorithm (YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv12) with consistent parameters. Utilizing a pre-annotated Kaggle MRI brain dataset, the research meticulously verified annotations and employed data augmentation (flipping, rotation, blurring, noise) to expand the dataset from 801 to approximately 1362 images, enhancing model generalization and robustness. Models were trained and evaluated on metrics including precision, recall, mAP@0.5, mAP@0.5:0.95, and inference time. YOLOv12 demonstrated superior overall performance, achieving the highest recall (97.32%), mAP@0.5 (92.2%), and mAP@0.5:0.95 (76.57%), establishing its robustness for accurate detection and object localization. While YOLOv7 achieved the highest precision (96.89%) and excellent inference speed, its overall mAP and recall were surpassed by other iterations. YOLOv9 and YOLOv8 also showed strong competitive performance, indicating significant advancements in the newer YOLO generations. The findings confirm the efficacy of the YOLO algorithm for brain tumor detection and localization in MRI images, with YOLOv12 proving to be the most effective variant in this comparative analysis.

I. INTRODUCTION

Brain tumors are a disease with an extremely high mortality rate. Brain tumors are ranked 2.6% in global cancer mortality, and the occurrence increases by a mean of 1.2% annually, according to the World Health Organization (WHO) [1]. Early and accurate diagnosis is therefore essential for an effective cure. Brain tumor diagnosis is currently primarily carried out using Magnetic Resonance Imaging (MRI) scans, and diagnosis is performed by radiologists [2]. The duration of this diagnostic procedure typically depends on the radiologist's skill level, as it involves manual detection.

In recent years, computer vision and Artificial Intelligence (AI) have significantly advanced, especially with deep learning, leading to their application across various professional fields [3]. Medical image analysis is one such area significantly impacted by these technologies. Convolutional Neural Networks (CNNs) and Region-based CNNs (R-CNNs) have been employed to segment brain tumors, achieving processing speeds comparable to human analysis. However, the computational complexity of CNN algorithms can make them time-consuming for real-time detection [4].

You Only Look Once (YOLO) is a single-stage detector algorithm architecture that offers a solution by providing inference speeds up to 10 times faster than R-CNNs, without significant compromise in accuracy [5]. YOLO treats object detection as a single regression problem, integrating bounding box prediction and object classification into one stage [6]. This algorithm has gained considerable popularity and widespread use in various object detection applications due to its efficiency and lightweight architecture [7].

Previous research has explored the application of various YOLO versions, including YOLOv3, YOLOv4, and YOLOv5 [8]. The YOLO algorithm has demonstrated

superior performance compared to other algorithms such as R-CNN, CNN, or Single Shot Detectors (SSDs) [9], [10]. Based on these prior findings, YOLO methods have proven effective in detecting brain tumor objects in medical images[11], [12]. Considering the aforementioned points, this study aims to perform brain tumor object detection on MRI images by comparing several versions of the YOLO algorithm (YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv12) and focuses on the binary detection of brain tumors, aiming to identify the presence and localization of a single 'tumor' class on MRI images.

II. Related Works

The advancement of MRI-based brain tumor detection has been significantly propelled by deep learning algorithms, with YOLO (You Only Look Once) emerging as a prominent technique due to its real-time object detection and localization capabilities. Numerous studies underscore the effectiveness of YOLO for this critical medical application.

Researchers have explored various YOLO iterations and modifications to enhance brain tumor detection. For instance, Mercaldo et al. [13] employed YOLOv8s, achieving a notable mAP@0.5 of 0.941 and a precision of 0.943. While promising, their study was limited by a relatively small dataset of 300 MRI images and lacked comparisons with other algorithms, hyperparameter tuning, or advanced optimization. Addressing the challenge of small tumor detection, Abdusalomov et al. [14] introduced architectural revisions to YOLOv7, integrating CBAM, SPPF+, and BiFPN modules. This innovative approach yielded an impressive accuracy of up to 99.5%, demonstrating the significant impact of architectural enhancements on detecting complex tumor areas. Similarly, Muksimova et al. [15] refined YOLOv5m by incorporating an Enhanced Spatial Attention (ESA) layer, which effectively reduced false positives and sharpened the detection focus on critical regions within MRI images. Beyond specific architectural improvements, comparative analyses have further elucidated the capabilities of the latest YOLO algorithms. Taha et al. [16] conducted a comparative study between YOLOv8 and YOLOv11 for tumor classification, differentiating between glioma, meningioma, pituitary, and non-tumor cases. Through transfer learning and fine-tuning, they achieved high accuracies of 99.49% with YOLOv8 and 99.56% with YOLOv11, solidifying the efficacy of these contemporary YOLO versions. Furthermore, recognizing the common challenge of limited datasets in medical imaging, Safdar et al. [17] emphasized the crucial role of data augmentation when training YOLOv3 models. Their findings highlighted rotation techniques (90° and 180°) as the most effective methods for improving model generalization, thereby addressing data scarcity and enhancing the robustness of the models. These collective efforts demonstrate a continuous evolution in leveraging and adapting YOLO for more accurate and robust brain tumor detection.

III. Methods

This research aims to detect brain tumors using various versions of the You Only Look Once (YOLO) algorithm, specifically comparing YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv12. This research will use a pre-annotated MRI brain dataset from Kaggle. To ascertain data accuracy, pre-existing annotations will be thoroughly checked and, where required, refined using the Roboflow platform. After this crucial step of data preparation, the entire dataset will be meticulously split into training, validation, and testing sets in a classic proportion of 70:20:10, respectively.

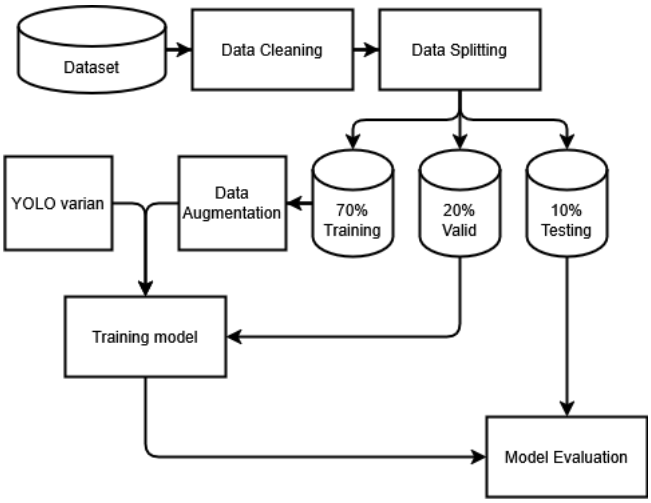


Figure 1 Method Workflow for YOLO

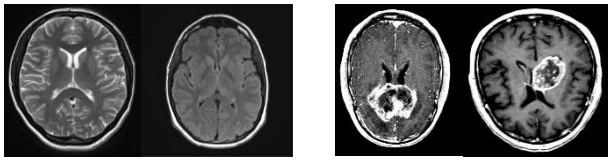
This meticulous division process guarantees that the models are trained on varied data and tested on entirely new samples, thus giving a comprehensive test of their generalization ability. The last stage of the research will be a comprehensive assessment of the performance exhibited by all YOLO models, as well as a side-by-side comparison using results achieved under similar parameters and in a comparable computational setup. This methodical process will enable the identification of the most effective and efficient YOLO variant for brain tumor detection. Figure 1 will be the visualization of the Workflow that is explained above.

A. Datasets

The data for this study was collected from Kaggle, a widely used public data repository for researchers in need of readily accessible datasets. The primary advantage of this dataset is the availability of existing annotations, eliminating the need for time-consuming and resource-intensive manual annotation from scratch.

Despite pre-annotated data being available, a significant aspect of our methodology was a strict verification and

fortification of such annotations. This rigorous process was required to generate quality and precise data, which overcame potential mistakes such as misaligned bounding boxes or incorrect labels. This laborious scrutiny ensures that the dataset is robust and reliable for model training.



(A) (B)

Figure 2 Example of brain axial MRI scan Dataset images without tumor (A) and with tumor (B)

Figure 2 represents an example of datasets that will be used in this research topic. This dataset contains a mixed sample of gliomas, meningiomas, and pituitary, but in this study the datasets specifically prepared for a single-class object detection task, focusing solely on the presence of a 'tumor' rather than differentiating between specific tumor types like glioma or meningioma. The dataset comprises 801 axial images of the healthy brain (A) and the diseased brain (B), with the original 500 (62%): 201 (25%): 100 (13%) splitting set. We modify these split set into 561 (70%): 160 (20%): 80 (10%). This number of samples is considered adequate to train deep learning models efficiently without the need for high-end computing facilities. Its adequate, yet manageable size makes the training process efficient and accessible to researchers with varying computational capabilities.

B. Preprocessing data

To enrich the dataset's variety and size, and to improve model generalization, data augmentation was performed using the Roboflow platform. The augmentation methods applied as shown in Table I:

TABLE I
AUGMENTATION METHODS FOR THE DATASETS

Method	Parameter Value
Flip	Horizontal
Rotation	Random ±15° range
Blur	15 pixels radius
Noise	0.5% Image Pixels

After the augmentation process, the total dataset used for training, validation, and testing amounted to approximately 1362 images. This data variation increased the amount of the original data by 561 compared to the original dataset 801. Originally, this dataset was divided into training, validation, and testing subsets with a 70:20:10 ratio, but after the augmentation the splitting becomes 82:12:6, resulting in approximately 1112 images for training, 160 for validation, and 80 for testing. All images were normalized and resized to the input dimensions appropriate for the YOLO architecture used, which is 640×640 pixels. This data augmentation will hopefully boost the resulting trained model into more robust and accurate for detection.

C. Model Architecture

At its core, a standard YOLO architecture consists of three primary parts: The Backbone extracts hierarchical features from the input image through the utilization of a Convolutional Neural Network (CNN). The Neck, sandwiched between the backbone and the head, combines and fuses feature maps at various scales, enhancing their representational capability. Lastly, the Head handles bounding box prediction, objectness scores, and class probabilities.

1) *YOLOv5*: by Ultralytics, is renowned for its native PyTorch code and versatile architecture, with a CSPDarknet backbone and a PANet + FPN neck mainly to facilitate the effective fusion of features at a range of scales. Its coupled head, which is anchor-based, predicts detection on three scales, with the key improvements aimed at improving engineering efficiency, accommodating different model sizes, and adding the essential C3 module for improved learning while removing duplicate gradients, thereby improving inference-constrained vs. computational performance trade-off [7], [15], [18].

2) *YOLOv7*: this model stands out by prioritizing "trainable bag-of-freebies" that achieves maximum accuracy without sacrificing inference cost. It is coupled with an Extended Efficient Layer Aggregation Network (ELAN) in the backbone and neck for enhancing learning efficiency and robustness. A significant enhancement is the incorporation of RepConv (Reparameterized Convolution), which enables the merging of a sophisticated multi-branch training network into one efficient convolutional layer for inference, thus decreasing latency without affecting accuracy[8].

3) *YOLOv8*: another from Ultralytics, is a significant shift towards a more universal, unified vision AI model by forgoing the use of explicit anchor boxes. Its backbone and neck feature a new C2f module replacing YOLOv5's C3 to be more efficient and perform better with richer gradient flow. A key innovation is the introduction of an anchor-free (decoupled) prediction head that simplifies the process by directly estimating object center offsets and bounding box dimension offsets, foregoing pre-defined anchor boxes[13], [16], [18].

4) *YOLOv9*: YOLOv9 introduces novel concepts to minimize information loss when propagating deep networks for enhanced accuracy and efficiency. Its backbone features the Generalized Efficient Layer Aggregation Network (GELAN), a flexible and efficient architecture that leverages ideas from CSPNet and ELAN for scalable flexibility and optimal computation utilization. A key innovation is the Programmable Gradient Information (PGI), in which a parallel branch is employed to generate stable gradient information to guide the learning of the main network and

prevent information degradation, especially in down-sampling [19].

5) *YOLOv12*: one of the latest developments in the series, brings an improved balance between operational speed and accuracy by consolidating advanced architectural components and training techniques. It presents novel improvements in the context of efficient layer aggregation networks and spatial pyramid pooling for enhanced feature extraction and fusion, along with optimized parameters with regard to network depth and width to maintain hardware efficiency. Improvements also involve advanced loss functions and incremental training techniques, which integrate modern data augmentation and regularization techniques to enhance more robust feature learning and enhance scalability to different implementation scenarios [20].

The selection of the medium variant (m) is based on a consideration of the balance between accuracy and computational complexity, making it suitable for various real-world applications without requiring extensive resources. Furthermore, this study specifically compares variants from several YOLO generations (v5 to v12) to evaluate the performance advancements achieved by each version chronologically.

Experiments were conducted in a Google Colaboratory environment utilizing an NVIDIA Tesla T4 Graphics Processing Unit (GPU) that is specified in Table IV. All models were trained using standard hyperparameters as follows:

TABLE II
MODEL PARAMETERS

Parameters	Parameter Value
Optimizer	AdamW
Batch Size	16
Epochs	100
Momentum	0.937
Image Size	640 x 640

To ensure both reproducibility and a consistent evaluation framework across different YOLO models, the training process for each version adhered strictly to its default configuration as provided by its respective official repository. Specifically, YOLOv5, YOLOv8, YOLOv9, and YOLOv12 were trained using the configurations from the official Ultralytics repository, which is their primary development source. Conversely, YOLOv7 utilized the default settings found in the repository maintained by WongKinYiu. Also, no additional architectural modifications were introduced during the training of any of these models. This approach ensures that the observed performance metrics directly reflect the inherent capabilities and design strengths of each YOLO variant under standard, out-of-the-box conditions, allowing for a fair comparison of their intrinsic object detection prowess.

D. Performance Evaluation

To observe whether the trained models show improvement from YOLOv5 to YOLOv12, this research will incorporate various evaluation techniques. The evaluation techniques will utilize performance metrics based on the confusion matrix, namely precision, recall, and mAP.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$Intersection\ over\ union\ (IoU) = \frac{Overlapped\ area}{area\ of\ union} \quad (3)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (4)$$

Note :
mAP = metrics average precision
N = number of classes
AP = Average Precision

The validation procedure involves testing the models on a testing dataset that is entirely separate from the training data, ensuring that the models have not seen this data previously. This aims to evaluate the model's generalization ability to new data. The detection results are evaluated using an Intersection over Union (IoU) example in Figure 3, with threshold of 0.5, meaning a prediction is considered correct if the predicted area and the ground truth area have at least 50% overlap.

IV. RESULTS AND DISCUSSIONS

This study conducted a comparative evaluation of five versions of the YOLO model (YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv12) to detect brain tumors. All models were trained using augmented datasets from Kaggle and Roboflow.

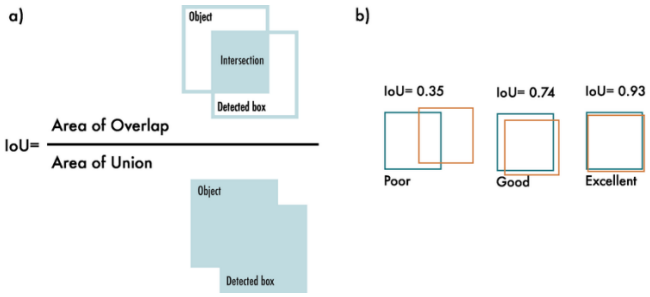


Figure 3 Representation of Intersection over Union (IoU)

TABEL III YOLO TRAINING VARIAN BEST RESULT OVER 100 EPOCHS

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Inference (ms/image)
YOLOv5	84.38	94.74	89.6	74.07	9.5
YOLOv7	96.89	90.23	76.1	62.59	11.9
YOLOv8	82.12	94.96	89.96	75.0	19.9
YOLOv9	84.01	96.24	91.78	76.5	14.2
YOLOv12	84.76	97.32	92.2	76.57	15.7
YOLOv12 (Original dataset)	78.79	89.47	81.46	64.62	17.2

The reason why the data augmentation was to increase the variety of data that would likely appear when using Magnetic Resonance Imaging (MRI), and hoping to boost the result which resulting in better performing model. We also implement a uniform hyperparameter configuration to ensure a fair comparison. The discussion included the performance of each model based on precision metrics, recall, mean Average Precision (mAP) at the IOU (Intersection Over Union) threshold of 0.5 (mAP50), and mean Average Precision at the IOU threshold range of 0.5 to 0.95 (mAP@0.5:0.95), inference time, as well as analysis of performance trends during the training process.

Image processing and classification are known to have high computational demands, such as large amounts of RAM and a powerful GPU. To ensure reproducibility and provide transparency regarding the computational resources used, all models were trained on Google Colaboratory. The specific hardware configuration utilized was a Tesla T4 GPU with 15.1 GB VRAM, 16 GB of RAM, and 120 GB of disk space, as detailed in Table IV.

TABLE III
EQUIPMENT SETUP

Environment	GPU	Ram	Disk
Google collab	NVIDIA Tesla – T4 (15.1 GB VRAM)	16 GB	120 GB

For hyperparameter Tuning, we set up 5 different parameters. The parameters are Optimizer, Batch Size, Epoch Count, Momentum, and image size. All images are resized to 640 x 640 for all models input image size. Also, to make the comparison more accurate, we set all YOLO models to have 100 epochs for each model.

The training process of the five versions of the YOLO model (YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv12) was conducted using the same dataset, consisting of 1362 images for a single class of objects, with each model trained for 100 epochs. We also use YOLOv12 to train with the original dataset for comparison to the augmented model. Table III summarizes the best performance (maximum value) achieved by each model for key metrics during the validation process.

As we can see from the data that is served in Table III, it is evident that no single model definitively excels across all metrics. YOLOv7 exhibits significantly higher performance in terms of Precision (96.89%), which indicates that the vast majority of its detections are correct. However, interestingly enough, at other aspects such as Recall, mAP@0.5, and mAP@0.5:0.95 it got outperformed by the older generation YOLOv5. This can happen possibly because the source code that is in the repository from WonKinYiu's GitHub repository hasn't been adapted to ultralytics methods. The other model that uses the ultralytics repository, most likely has already received some changes and improvements. The other probable cause is that the pre trained weight that is used in the model training is unoptimized, since when the researcher did the model training, YOLOv7 took the longest time to train compared to the others. However, in terms of inference time YOLOv7 excel with second best speed compared to the others. This inference time can be useful for faster tumor detection, while also have more room in terms of hardware leniency.

Conversely, YOLOv12 with the data augmentation achieved the highest recall, demonstrating its exceptional ability to detect almost all target objects present in images. YOLOv12's superiority is also apparent in the mAP metrics, where it records the highest values for both mAP@0.5 (92.2%) and mAP@0.5:0.95 (76.57%). YOLOv12's highest recall (97.32%) is particularly significant in a medical context as it indicates the model's exceptional ability to detect nearly all existing tumors, minimizing the critical risk of false negatives. This highlights YOLOv12's superior capability in detection accuracy and object localization across various IOU thresholds, making it accurately localizing the tumors across different precision requirements. In terms of detecting time, YOLOv12 lag behind by few millisecond compared to YOLOv5,v7, and v9 with it's 15,7 ms, this few millisecond sacrifice is still

acceptable, while given that YOLOv12 model is the best performing model out of all tested model.

YOLOv9 and YOLOv8 also show a promising performance against the newer YOLOv12 version in mAP@0.5:0.95 metrics, closely approaching the performance of the newer version by a small margin, but still slightly below the YOLOv12 model in other aspects. However unlike YOLOv8, YOLOv9 excel in terms of inference time compared to YOLOv12 with 14,2 ms. While YOLOv8 lag so far behind with 19.9 ms for it's inference time. For benchmark comparison to the YOLO model that is using the data augmentation, we train YOLOv12 model that is performing best in our test result without data augmentation for an example. The resulting of the model training is that, the model without the data augmentation resulting in lower score compared to the other model that is using the data augmentation. However it is still showing higher score in mAP@0.5 and mAP@0.5:0.95 compared to the YOLOv7 model.

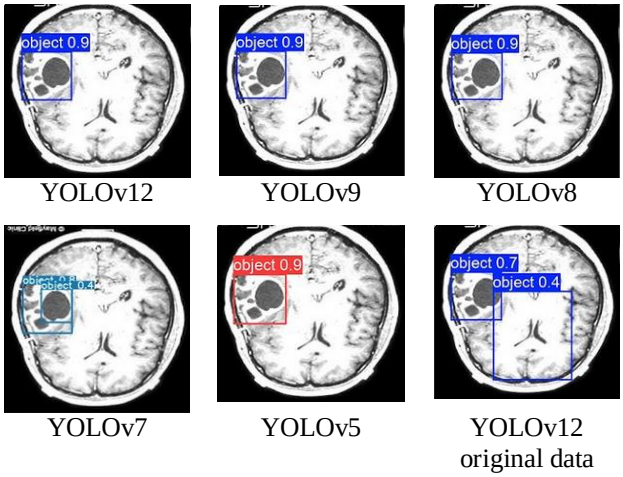


Figure 4 Detection and localization of Brain Tumor with the same image of each YOLO model shown in Figure 4

Figure 4 illustrates the precise localization capabilities of each model, demonstrating a direct correlation between high precision and recall and increased confidence in tumor detection. Notably, YOLOv5, YOLOv8, YOLOv9, and YOLOv12 consistently achieve a superior confidence rate of 0.9. In stark contrast, YOLOv7 and YOLOv12 (trained on the original dataset) exhibit significantly diminished confidence when analyzing identical images. This deficiency manifests as the generation of erroneous bounding boxes by both YOLOv7 and YOLOv12 (original data), a critical flaw absent in the other models. Specifically, YOLOv5, YOLOv8, YOLOv9, and YOLOv12 accurately delineate the true tumor locations. Conversely, YOLOv7 and YOLOv12 (original dataset) not only misidentify two distinct tumor areas but also fail to produce consistent or overlapping bounding box predictions, highlighting their unreliability in accurate tumor localization.

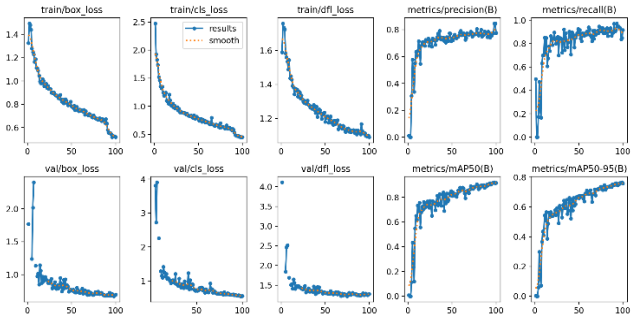


Figure 5 YOLOv12 Training Performance

The resulting data from Figure 5, shows a very interesting result, where in the Training Losses graph we can conclude that all training losses show a clear and consistent downward trend. Especially in train/cls_loss, it shows that it decreases quickly and then remains low. For a single-class detection problem like this brain tumor detection, this is an expected result. Where this means the model efficiently learns to classify the presence of the single 'tumor' class. This indicates that the YOLOv12 model is effectively learning and fitting the training data, This shows a positive sign of training.

Conversely, in the validation losses we can see an initial instability where in the first few epochs that we trained, we got NaN as the returned value. This suggests some initial numerical instability or issues in processing the validation set at the very beginning of training. This can suggest that the initial learning rate might have been slightly aggressive, but the model adjusted. However, these issues quickly resolved by itself, and the losses became stable and continue to decrease, which is a good recovery.

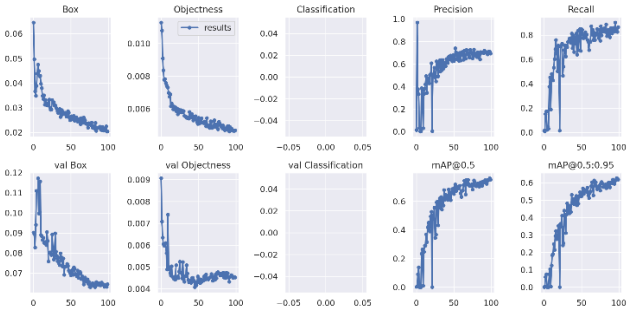


Figure 6 YOLOv7 Training Performance

Contrary to YOLOv12, YOLOv7 our worst performing model has a different result, as seen in Figure 6 which we can see in the validation objectness graph shows a clear downward trend, mirroring the training objectness, this indicates that the model is generalizing well to unseen data. There were still a couple of NaN results at the very beginning of training, but these were successfully handled and do not affect the overall trend. These Curves Indicate that the losses are still decreasing, Suggesting that the model could potentially benefit from having more training epochs to minimize these values. There's also a classification graph

and val classification graph that returned 0, this was normal since the dataset only contain one class of object 'Tumor', the model doesn't need to distinguish between multiple object categories. This type of graph is also similarly achieved by YOLOv5 version since they use classification loss instead of Distribution Focal Loss (dfl_loss) like the newer generation.

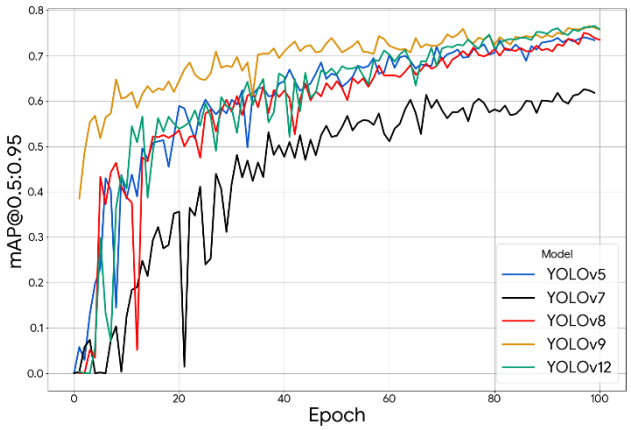


Figure 7 YOLO mAP@0.5:0.95 comparison over epoch

In Figure 7, the training trend for mAP@0.5:0.95 of each model is achieving a similar result, where it keeps increasing until the last epoch. The mAP@0.5:0.95 graph clearly shows that YOLOv12 and YOLOv9 consistently achieve higher mAP@0.5:0.95 values across epochs, with their lines closely intertwined at the top. However, interestingly enough, YOLOv9 starts its training from a higher mAP, the model reaches higher result in early training, this can be useful if we train the YOLO model with a lower count of epochs. YOLOv8 and YOLOv5 follow, while YOLOv7 lags significantly behind throughout the training process. This graph is crucial as mAP@0.5:0.95 gives a comprehensive view of detection accuracy across various IoU thresholds, reflecting the model's ability to localize objects precisely.

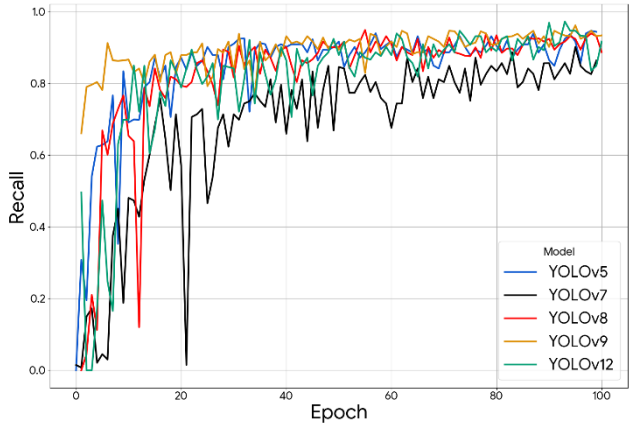


Figure 8 Recall over epoch comparison for each YOLO version

The recall graph shown in Figure 8, shows that YOLOv9 and YOLOv12 tend to have higher recall values compared to other models. The high recall rates observed, can be partially attributed to the extensive data augmentation applied, which significantly increased the diversity of the training data and improved the model's ability to recognize various tumor presentations under different conditions this indicating that they were better at detecting all actual objects present in the images, resulting in fewer false negatives in the confusion matrix. However YOLOv8 and YOLOv5 also perform very well in recall, with a slight instability at the beginning of the epochs. On the contrary, YOLOv7 performs worst with the lowest recall, which indicates that the model missed a significant number of the actual objects compared to the other YOLO models.

In Figure 9, which contain a precision graph, we can conclude that it was similar to recall, where YOLOv12 and

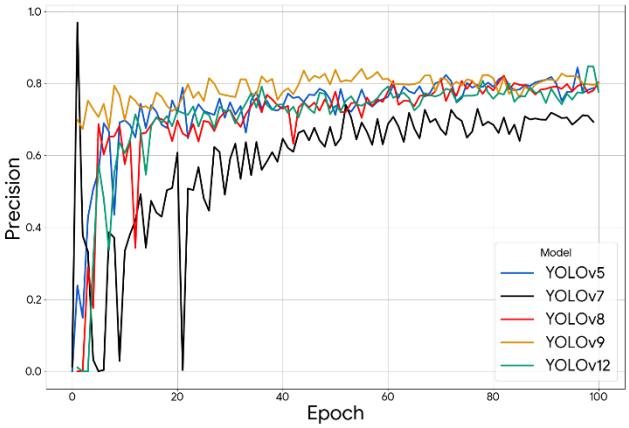


Figure 9 Precision over epoch comparison for each YOLO version

YOLOv9 have higher precision values throughout the training, which means that the model results in a lower rate of false positives. YOLOv8 and YOLOv5 shows a slightly lower precision over epochs but still comparable to the other model, while maintaining good generalization of the data that is seen. An interesting phenomenon observed in the training logs of YOLOv7 involves its precision metric during the initial epochs. At Epoch 1, the model exhibited an exceptionally high precision of 0.9689, a significant jump from Epoch 0's 0.01333, before subsequently decreasing and fluctuating in later epochs. This initial jump can be attributed to the inherent dynamics of early-stage deep learning training. Precision, defined as TP/(TP+FP), might temporarily spike if the model is being overly conservative, making very few predictions but getting most of those correct. As training progresses and the model endeavors to increase recall (detect more objects), it may inevitably introduce more false positives, causing a frequent dip in precision as it attempts to generalize across a wider, more complex range of data.

V. CONCLUSION

Results demonstrate that even though YOLOv12 sacrifice some miliseconds of it's inference speed, the model performed consistently, with the highest recall rate of 97.32%, mAP@0.5 of 92.2%, and mAP@0.5:0.95 of 76.57%. This reveals the strong ability of YOLOv12 in detecting brain tumors accurately and locating them correctly at various Intersection Over Union (IoU) thresholds. While YOLOv7, albeit an anomaly achieved the best precision (96.89%) while also having the second best speed compared to others, its overall performance in mAP and recall was outmatched by other iterations, which means that its precision comes at the cost of losing some true positives. YOLOv9 and YOLOv8 also posted strong performances, getting really close to YOLOv12, particularly in the mAP@0.5:0.95 category, which is indicative of huge improvement in the newer generations of YOLO algorithms.

The examination of training performance plots showed the consistent reduction of training and validation losses for the majority of the models, indicating good learning and generalization. Despite the early occurrence of numerical instabilities (NaN values) in the validation losses of YOLOv12 and YOLOv7, these were rapidly resolved, allowing the models to regain efficient learning capabilities.Single-class detection classification loss, i.e., 'Tumor', always converged to zero, an expected and hoped-for result.

In summary, the findings of this study confirm the efficacy of the YOLO algorithm for brain tumor detection and localization in MRI images with a significant performance boost from the previous to the latest models. Surprisingly, YOLOv12 is the best model for this particular task. It is important to note that this study was conducted entirely using a public dataset from Kaggle and did not involve real-world testing in a clinical medical environment. Future research can consider additional hyperparameter optimization, more aggressive data augmentation techniques, application of novel architecture changes, or exploring the application of these models into real clinical setups.

REFERENCES

[1] WHO., "Global Cancer Observatory: Brain Tumour Statistics," 2023.

[2] M. F. Almufareh, M. Imran, A. Khan, M. Humayun, and M. Asim, "Automated Brain Tumor Segmentation and Classification in MRI Using YOLO-Based Deep Learning," *IEEE Access*, vol. 12, pp. 16189–16207, 2024, doi: 10.1109/ACCESS.2024.3359418.

[3] M. Puttagunta and S. Ravi, "Medical image analysis based on deep learning approach," *Multimed Tools Appl*, vol. 80, no. 16, pp. 24365–24398, Jul. 2021, doi: 10.1007/s11042-021-10707-4.

[4] M. I. Mahmud, M. Mamun, and A. Abdelgawad, "A Deep Analysis of Brain Tumor Detection from MR Images Using Deep Learning Networks," *Algorithms*, vol. 16, no. 4, Apr. 2023, doi: 10.3390/a16040176.

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934

[6] J. Huang, W. Ding, T. Zhong, and G. Yu, "YOLO-TumorNet: An innovative model for enhancing brain tumor detection performance," *Alexandria Engineering Journal*, vol. 119, pp. 211–221, Apr. 2025, doi: 10.1016/j.aej.2025.01.062.

[7] A. Ishtaiwi *et al.*, "Impact of Data-Augmentation on Brain Tumor Detection Using Different YOLO Versions Models," *International Arab Journal of Information Technology*, vol. 21, no. 3, pp. 466–482, May 2024, doi: 10.34028/iajit/21/3/10.

[8] R. S. Passa *et al.*, "Deteksi Tumor Otak pada Magnetic Resonance Imaging menggunakan YOLOv7," *Jurnal Ilmiah MATRIK*, vol. 25, no. 2, Aug. 2023.

[9] P. Cinantya, S. Catur, Amalia, and R. P. Khalivio, "Comparative Analysis of YOLOv5 and YOLOv8 Cigarette Detection in Social Media Content," *Scientific Journal of Informatics*, vol. 11, no. 2, pp. 341–352, May 2024, doi: 10.15294/sji.v11i2.2808.

[10] S. Shinde, A. Kothari, and V. Gupta, "YOLO based Human Action Recognition and Localization," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 831–838. doi: 10.1016/j.procs.2018.07.112.

[11] N. S. Kumar and A. K. Goel, "Detection, Localization and Classification of Fetal Brain Abnormalities using YOLO v4 Architecture," *International Journal of Performability Engineering*, vol. 18, no. 10, pp. 720–729, Oct. 2022, doi: 10.23940/ijpe.22.10.p5.720-729.

[12] N. Iriawan *et al.*, "YOLO-UNet Architecture for Detecting and Segmenting the Localized MRI Brain Tumor Image," *Applied Computational Intelligence and Soft Computing*, vol. 2024, 2024, doi: 10.1155/2024/3819801.

[13] F. Mercaldo, L. Brunese, F. Martinelli, A. Santone, and M. Cesarelli, "Object Detection for Brain Cancer Detection and Localization," *Applied Sciences (Switzerland)*, vol. 13, no. 16, Aug. 2023, doi: 10.3390/app13169158.

[14] A. B. Abdusalomov, M. Mukhiddinov, and T. K. Whangbo, "Brain Tumor Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging," *Cancers (Basel)*, vol. 15, no. 16, Aug. 2023, doi: 10.3390/cancers15164172.

[15] S. Muksimova, S. Umirzakova, S. Mardieva, N. Iskhakova, M. Sultanov, and Y. I. Cho, "A lightweight attention-driven YOLOv5m model for improved brain tumor detection," *Comput Biol Med*, vol. 188, Apr. 2025, doi: 10.1016/j.compbiomed.2025.109893.

[16] A. M. Taha, S. A. Aly, and M. F. Darwish, "Detecting Glioma, Meningioma, and Pituitary Tumors, and Normal Brain Tissues based on Yolov11 and Yolov8 Deep Learning Models," Mar. 2025, [Online]. Available: http://arxiv.org/abs/2504.00189

[17] M. F. Safdar, S. S. Alkobaisi, and F. T. Zahra, "A comparative analysis of data augmentation approaches for magnetic resonance imaging (MRI) scan images of brain tumor," *Acta Informatica Medica*, vol. 28, no. 1, pp. 29–36, Mar. 2020, doi: 10.5455/AIM.2020.28.29-36.

[18] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.02988

[19] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.13616

[20] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.12524