Brain Tumor Detection using Machine Learning

Gabriel Goh Sze Wei A0280490E [1] **Low Xing Yi A0280521N** [1] **Muhammad Bahrul Wusto A0280510U** [1]
Wong Sook Kwan A0280530N [1] **Xin Yuhe A0280451L** [1]

Abstract

The primary objective of our project is to train an image detection model capable of augmenting medical professionals' diagnostic accuracy for the early and precise detection of brain tumours. To achieve this, we intend to evaluate and refine serveral machine learning models such as Vision Transformer, U-Net & DenseNet-121, and YOLO using a pre-labelled brain tumour MRI scans dataset. This report will provide insights into the methodology, challenges and potential impact of this project, demonstrating its significance in in supporting and augmenting decision-making processes in medical diagnostics.

1. Dataset Overview

1.1. Pre-labelled MRI Brain Tumour Image Dataset

For this project, we will be using compiled pre-labelled MRI brain tumor MRI scans dataset (Rostami, 2024) featuring four tumor classification: pituitary, meningioma, and glioma, along with non-tumorous scans. The MRI scans have been labeled by medical experts using a standardised labeling protocol and include the type of tumor and the bounding box coordinates of the tumor. The dataset includes 2,443 MRI images and the images are resized to 640x640 (Stretch).

Class 0 Glioma tumor: Shows tumors originating in brain support cells, varying from slow-growing to aggressive, with symptoms like headaches and cognitive changes.

Class 1 Meningioma tumor: Depicts tumors arising from brain membrane, often benign and may cause headaches or neurological symptoms.

Class 2 No tumor: Images without any tumors, used for comparison and evaluating model performance.

Class 3 Pituitary tumor: Represents benign tumors in the pituitary gland causing symptoms like headaches and hormonal imbalances.

1.2. Bounding Box Coordinates Conversion

As some of our models do not support bounding box coordinates, we need to use these coordinates to manually convert them into a rectangular masks, to enable us to maintain the tumor locations across all models.

1.3. Data Preprocessing

In the Pre-labelled MRI Brain Tumour Image Dataset (*Refer to Figure 1*), it includes 1,695 training images, 502 validation images, and 246 test images, each annotated with the tumor type and location. As the training set contains an inadequate number of samples to effectively train a deep CNN architecture, data augmentation is employed to address this limitation.

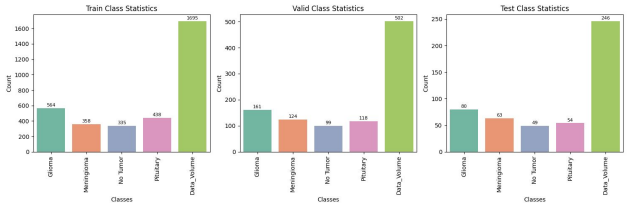

Figure 1. Overview of Classes in Train-Test-Split Datasets

1.4. Data Augmentation

By using this approach, a larger dataset can be generated without the need for additional data collection. In this project, we carried out the following data augmentation techniques:

- **Rotate:** Random rotations between -10° and 10°, to mimic head tilting and rotation.
- **ShiftScaleRotate:** A combination of shifting and scaling (without rotate), for robustness against positional and size variations.
- **HorizontalFlip:** Flips the image horizontally (left to right).
- **ElasticTransform:** Elastic transformations, introducing non-linear deformations for mimicking real-world

scan inconsistencies.

Table 1 illustrates the distribution of images across classes within the training and validation set before and after the application of data augmentation. We initially observed a class imbalance, with Class 0: Glioma tumor having more samples compared to Class 2: No tumor. Despite the data augmentation, we chose to maintain the imbalance because detection of Glioma tumor (Class 0) requires urgent clinical intervention due to its aggressive nature, and we need to ensure our model is especially sensitive in detecting them.

Table 1. Distribution of classes within the dataset before and after implementation of data augmentation within the training and validation set.

	Train Set		Validation Set		Test set
	Before aug.	After aug.	Before aug.	After aug.	
Class 0	564	1128	161	322	80
Class 1	358	716	124	248	63
Class 2	335	670	99	198	49
Class 3	438	876	118	236	54
TOTAL	1695	3390	502	1004	246

2. Machine Learning Models for Brain Tumor Classification in MRI Scans

This section explores the application of machine learning models for classifying brain tumors in MRI scans highlighting the strengths of these models for medical image analysis. Specifically, we have chosen a set of diverse models, namely, "Vision Transformer (ViT)", combined "U-Net & DenseNet-121", as well as "You Only Look Once (YOLO)".

2.1. Image Recognition Expertise

Deep learning models, including Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), excel at image recognition tasks due to their architectures designed for analysing visual data. These models can identify crucial details within MRI scans and distinguish healthy tissue from tumors. Studies have demonstrated their effectiveness in disease detection and classification across various imaging modalities, including MRI scans. This established success highlights their potential for brain tumor classification.

2.2. Automated Feature Extraction

Both CNNs and deep learning models like ViTs can automatically learn relevant features from MRI data, eliminating the need for manual feature engineering, a complex and time-consuming process. This allows them to adapt to the specific characteristics of brain tumors in scans.

To ensure a fair comparison, we will evaluate all models on a common set of metrics commonly used in medical image analysis (Hicks et al., 2022) for classification tasks.

By evaluating these diverse models on this common set of metrics, we aim to identify the one that achieves the best overall performance in classifying brain tumors using MRI scans. This model will then be further explored for potential enhancements in tumor detection and classification capabilities.

- **Precision:** The proportion of correctly identified tumors among all predicted tumors.
- **Recall:** The proportion of actual tumors correctly identified by the model.
- **F1 Score:** A balanced measure of precision and recall, providing a single score for model performance.

3. Model 1: Vision Transformer

Vision Transformer (ViT) is a deep learning model for image recognition tasks (*Refer to Figure 3.1*). Unlike traditional Convolutional Neural Networks (CNNs) that process images through convolutions, ViT breaks an image into smaller patches, converts them into vectors, and feeds them into a Transformer encoder.

In our task, MRI scans often show subtle variations in tissue contrast that can be crucial for tumour identification. ViT's self-attention mechanism and transformer architecture allow the model to learn long-range dependencies between different parts of the image, which is crucial for accurate image classification and object detection. The original architecture of the Vision Transformer, however, does not have the specific capability to conduct image segmentation task. Despite that, due to the flexibility given by the ViT structure as well as the unique trait in understanding distant part of the image, we believe that this model has a great potential to locate and identify tumour effectively.

Therefore, to achieve our classification and segmentation goals, we designed and added 2 linear layers—classification head and bounding box head to output the class labels and coordinates. Through this modification, we empower this model to predict the location of the tumour, without the configuration of detailed image segmentation.

3.1. ViT Architecture

Image Patching: The input image is split into fixed-size patches (in our case, 224x224 pixels each).

Embedding: Each patch is flattened and projected into a higher-dimensional space via a trainable linear transformation. Positional embeddings are added to retain positional information of the patches.

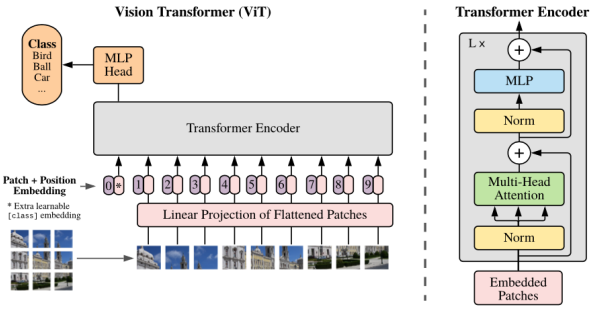

Figure 2. Vision Transformer ViT Architecture https://viso.ai/deep-learning/vision-transformer-vit/

Transformer Encoder: The sequence of embedded patches passes through multiple layers of the transformer encoder. This encoder uses a self-attention mechanism to learn relationships between different parts of the image, captured in the patch vectors. Through multiple encoder layers, the model progressively builds a global understanding of the image.

Output Layers: Depending on the task, the final layers of the ViT can vary. For image classification, a linear layer takes the final encoder output and predicts class probabilities. In our project, because the given data also provides the information of the tumour location in each image, we designed our own ViT model and extended this concept by adding another linear layer to predict bounding box coordinates for object detection tasks.

3.2. Evaluation of Vision Transformer

To explore and achieve a satisfying performance of ViT model in this task, we trained the model with different settings. We started by utilizing a pre-trained vit-base-patch16-224 model. The pre-training on a large image dataset allows the model to learn powerful image feature extraction capabilities. These capabilities can then be adapted to the specific task of tumor segmentation and classification in MRI scans, even with a potentially smaller medical image dataset.

We first want to demonstrate the transfer learning capability of this model, and after training with 10 epochs and learning rate 1e-3, we achieved F1 score 0.409, precision score 0.428 and recall 0.400 *(Refer to Table 2)*. We noticed that the performance was not good as expected, considering the advantages discussed above. Thus, a detailed finetuning process was conducted to this model. We freezed all layers in the model, and re-trained our model by updating the parameters of the two newly designed heads for classification and bounding box identification. After the same 10 epochs training, we achieved F1 score 0.791, precision score 0.824 and recall 0.772. The finetuning process greatly improve the model performance, and demonstrate the transfer learning capabilities of the ViT model. To further improve the performance of the ViT model, we also explore the option of replacing Adam optimizer with AdamW, an optimizer that is often recommended for fine-tuning pre-trained models because of its improved handling of weight decay. After implementing the new optimizer, we acheived F1 score 0.837, precision score 0.859 and recall 0.822.

Table 2. Evaluation of Vision Transformer

Model	F1 Score	Precision	Recall
ViT, 10 epochs, Adam	0.409	0.428	0.400
ViT, 10 epochs, frozen layers, Adam	0.791	0.824	0.772
ViT, 10 epochs, frozen layers, AdamW	0.837	0.859	0.822

While a great improvement was achieved by freezing layers, it is worth noting that the training precision is less than test precision. Therefore, there is reason to believe that our model is overfitted. Despite that, there is gradual improvement of the model performance after the modifications of the training process were implememnted as listed in the table above.

The results indicate a significant improvement in all metrics when the backbone of the Vision Transformer is frozen during the initial training phase. It might indicate that freezing the pre-trained layers helps in preserving the learned features, which are generally useful for visual recognition tasks. It seems that allowing these features to remain stable while only the newly added heads (classification and bounding box) are trained helps the model to adapt better without being overwhelmed by too many changes at once.

Changing the optimizer from Adam to AdamW leads to an improvement in all performance metrics (F1 score, precision, and recall), albeit the loss remains nearly unchanged. AdamW modifies the way weight decay is handled, which can lead to better generalization by decoupling the weight decay from the learning rate schedule. This often results in more stable and effective training, particularly for complex models like transformers.

4. Model 2: U-Net and DenseNet-121

Given the nature of our project, we also explored two neural network models which are widely used in the medical imaging field.

In medical imaging workflows, image segmentation typically precedes classification, where potentially tumorous areas are first delineated and isolated. Once the MRI images have been segmented, classification is then performed to

identify and classify specific tumours within the areas of interest. This approach ensures that the focus is narrowed down to only the most relevant areas of an MRI image, thereby allowing for more accurate diagnoses.

For our project, we utilized U-Net for image segmentation and DenseNet-121 for image classification, and implemented these models using the Medical Open Network for AI (MONAI) library. MONAI is a PyTorch-based framework that provides tools and pre-trained models which are robust and specifically tailored for medical imaging tasks. By leveraging MONAI, we hope to leverage field specific tools and reduce the need for additional configuration.

U-Net was originally designed for medical imaging segmentation and is known for its exceptional ability to capture fine details, contexts, and patterns. On the other hand, DenseNet-121 is highly efficient in classification tasks due to its dense connectivity, feature propagation and reuse. These models are particularly well suited, well adopted and well liked for medical imaging.

4.1. U-Net Architecture

U-Net is an encoder-decoder convolutional neural network that has a symmetric architecture (*Refer to Figure 4.1*). It utilizes a series of contracting (encoder) and expanding (decoder) paths to capture context and enable precise localization respectively. The contracting path consists of repeated applications of two 3x3 convolutions (each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step, the number of feature channels is doubled. The expansive path is done in reverse sequential order and reconstructs the original resolution. Put together, the features captured in the contracting path are integrated during the expansive path, thereby enabling accurate segmentation of medical images.

For our model, we used Parametric Rectified Linear Unit (PReLU) as our activation function, as it introduces non-linearity in order to learn more complex mappings between input and output segmentation masks, which is beneficial for medical imaging. PReLU is also known to mitigate the vanishing gradient issue and to improve convergence. In addition to Binary Cross Entropy loss function, we also tested the Dice loss function due to its prominence in medical settings, its ability to handle class imbalances and small objection detection tasks.

4.2. DenseNet-121 Architecture

DenseNet-121 comprises 121 feed-forward layers which have high information flow thereby creating very deep and dense connections in the network (*Refer to Figure 4.2*). This dense connection encourages enhanced feature reuse and

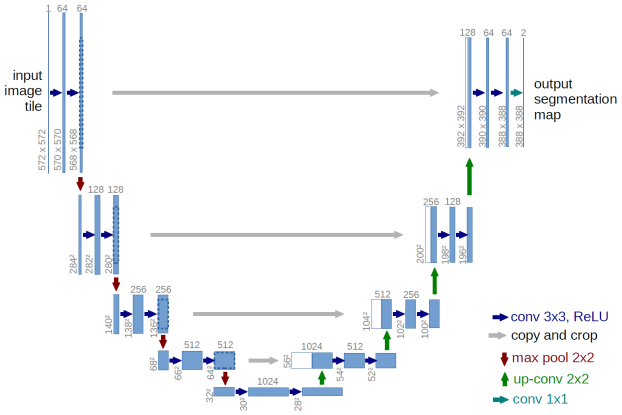

Figure 3. U-Net Architecture

mitigates the vanishing gradient problem. This enables DenseNet-121 to capture intricate details and patterns in MRI scans even with limited datasets (as is typical in medical settings).

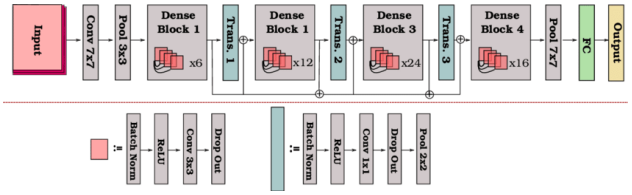

Figure 4. DenseNet-121 Architecture

For our model, we used Cross Entropy Loss as the standard default loss function and also for its ability to handle class imbalances.

4.3. Evaluation of U-Net and DenseNet-121 Models

To establish a baseline, we trained both models for 10 epochs at a fixed learning rate of 1e-3, no further tuning beyond the above-mentioned was performed. The results indicated that U-Net's performance was consistent and balanced across all three metrics F1, Precision and Recall around 0.819, but IoU was only moderately good at 0.693. DenseNet-121's results were slightly lower than U-Net except Precision *(Refer to Table 3)*.

Both false positive and false negative diagnoses have significant consequences for patients, thus it is important to minimize false negatives for segmentation and minimize false positives for classification. Considering our baseline results and their implications for medical imaging workflows, we can infer one of two things.

- U-Net's higher recall suggests that it is better at minimizing false negatives i.e. minimizing cases which are tumorous but marked as benign. This makes U-Net more suitable for applications like image segmentation where, due to its role as a preliminary step before classification, makes it even more crucial to accurately delineate all potential areas of interest.

- DenseNet-121 demonstrated slightly stronger results in Precision, suggesting that it is more effective in minimizing false positives. This attribute makes DenseNet-121 more suitable for applications like image classifications where it can better distinguish between tumor cases e.g. glioma, pituitary, meningioma and benign. This in turn can help to improve the efficacy of the treatment plan and reduce over-diagnose or unnecessary medical interventions.

4.4. Optimising UNet and DenseNet-121

Due to computational resource constraints, it was important that we strategically prioritized and implemented only high impact optimization techniques. The following fine-tuning techniques were implemented:

- **Optimizer change from Adam to AdamW:** We opted for AdamW over other optimizers like Adam (our baseline) and Stochastic Gradient Descent (SGD), due to its adaptive learning capabilities even for complex models, and its ability to self regularize; thereby reducing overfitting more effectively. This was particularly beneficial to us due to our fairly limited dataset and gave us a chance to build-train models that realistically simulate real-world medical imaging workflows where excellent generalization is expected despite limited training data.

- **Learning rate scheduler with gamma = 0.9:** e implemented an exponential learning rate scheduler to allow us to efficiently and systematically test the effects of various learning rates on the training process and model convergence. The decay rate was set at 0.9 to allow for a modest and gradual reduction in learning rate, given that we only ran 10 epochs for model training.

- **Minority class weighting in BCE loss function for U-Net:** During fine-tuning, we observed that despite implementing the above two changes, the U-Net model continued to fail at learning during initial epochs e.g. no validation scores for epochs 0 to 2. This could have been due to the high class imbalance of benign classes (¡ 20% of total samples) in our train dataset. Hence, we weighted the "no tumor" class in order to counteract this class imbalance and increase U-Net's sensitivity to tumor segmentation.

- **Warmup scheduler for U-Net for epochs 0 and 1:** To further aid the learning process during the initial training phases, we implemented a custom warmup scheduler. The custom warmup scheduler prevents the model from "settling down" too quickly at the start, and when paired with the learning rate scheduler, it gradually increases the learning rate from a significantly lower initial value to the intended learning rate over the first two epochs. Starting with a lower learning rate helps to stabilize the training process and prevents premature convergence (overfit); this is another feature that is well-suited for medical imaging since input images can vary in quality and structure.

We also evaluated how the best models were identified and saved from the training epochs by looking at the *minimum validation loss* and the *maximum validation F1 score* over all epochs. Once again, given our relatively small and highly unbalanced dataset, we determined that prioritizing maximum validation F1 score would be more effective for U-Net and DenseNet121. F1 is a balanced measure of recall and precision and is well suited for medical imaging as a high F1 score indicates that the model adequately recognizes even rare but critical conditions without punishing false positives. Conversely, low validation loss can be skewed by class imbalances and majority classes (as is the case for our dataset); in real life this means that less common tumors might not be detected.

After fine-tuning, we observe that DenseNet-121 responded extremely well just by changing the optimizer and implementing a learning rate scheduler, achieving almost a near perfect label agreement with an F1 score of 0.961 *(Refer to Table 3)*. Similarly, U-Net also performed at its best by changing the optimizer and implementing a learning rate scheduler, achieving an F1 score 0.841 and IoU of 0.725. Also, U-Net's F1 and Precision decreased with complex fine-tuning strategies, but Recall benefitted greatly; this could suggest that the weighted BCE loss and warm-up strategies potentially caused overfitting to the minority class causing highly accurate identification of the tumor cases but at the expense of misclassifying benign classes (reduced Precision and IoU).

If we were to adopt the two current best versions of our models as part of a consolidated segmentation-classification pipeline for medical imaging, the consolidated model will likely not be sufficiently fit for use due to the slightly inferior segmentation model. Additional iterative fine-tuning strategies are required to improve U-Net, also bearing in mind that amongst other things, our dataset can be further augmented, and our training depth can be increased.

Table 3. Evaluation of U-Net and DenseNet-121 Models

Model with Test Dataset	F1[1]	Precision	Recall	IoU
U-Net[2], baseline	0.819	0.831	0.806	0.693
U-Net, finetuned				
• AdamW with learning rate scheduler	0.841	0.841	0.840	0.725
• AdamW with learning rate scheduler and weighted BCE loss	0.795	0.717	0.893	0.660
• AdamW with learning rate scheduler, warm-up scheduler and weighted BCE loss	0.764	0.653	0.919	0.618
DenseNet-121, baseline	0.797	0.844	0.792	NA
DenseNet-121, finetuned with AdamW and learning rate scheduler	0.961	0.961	0.962	NA

[1]Equivalent to Dice Coefficient. [2]Using Binary Cross Entropy loss function.

5. Model 3: YOLO

YOLO (You Only Look Once) is a powerful deep learning model that stands out for its object detection capabilities. Unlike traditional image classification models that analyse the entire image, YOLO excels at pinpointing and classifying specific objects within an image.

Furthermore, YOLOv8 achieves good accuracy while maintaining processing speed. This efficiency translates to faster analysis of images, which can be crucial in real-time clinical settings in our context. Additionally, with minor adjustments, YOLOv8 can be adapted for classification tasks as well. This potential allows for a single model that can not only detect but also classify objects, potentially streamlining the analysis process.

Beyond its core functionality, YOLOv8 boasts user-friendly features like a command-line interface and a well-structured Python package. This user-friendliness facilitates implementation and integration into existing workflows. Moreover, YOLOv8 is backed by a large and active community, providing readily available support and resources for troubleshooting and further development.

These qualities make YOLOv8 a valuable tool for advancing medical image analysis and potentially improving medical diagnosis and could be a compelling choice for brain tumor detection in MRI scans and.

5.1. YOLO Architecture

YOLO's architecture typically consists of three key components such as backbone, neck and head (*Refer to Figure 5*).

Backbone: This initial stage extracts high-level features from the input image. These features capture essential details about shapes, textures, and patterns crucial for object identification.

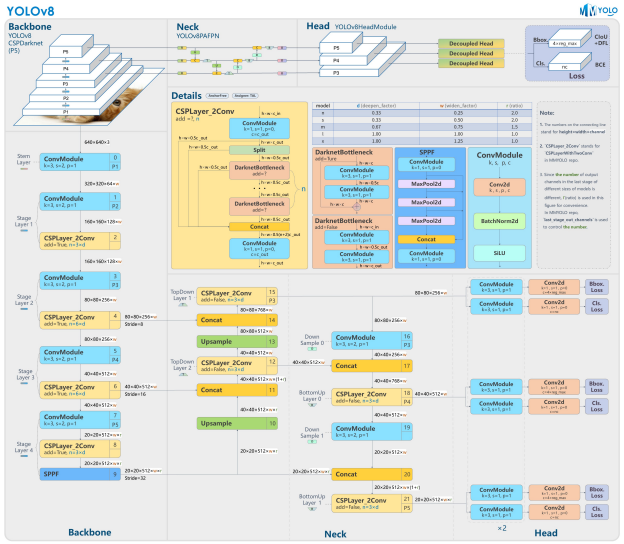

Figure 5. YOLOv8 Architecture by GitHub user RangeKing

Neck: This module combines feature maps from various backbone levels, allowing the model to capture features at different resolutions for richer analysis. Unlike some previous models, YOLOv8 might concatenate these features without forcing them to have the same number of channels, potentially leading to a more compact model size.

Head: The final stage predicts bounding boxes and class probabilities for detected objects. YOLOv8's key distinction lies in its anchor-free approach, directly predicting bounding box coordinates and class probabilities, potentially simplifying the process and improving accuracy.

5.2. Evaluating Different Versions of YOLO

YOLO offers flexibility in model size that caters to different deployment needs. Larger YOLO versions, like YOLOv5-X, prioritize high accuracy but require more computational resources. Conversely, smaller versions like YOLOv5-Tiny prioritise speed and efficiency, making them suitable for real-time applications on devices with limited processing power. This range of sizes allows you to choose the best YOLO model to strike a balance between speed and accuracy requirements.

For this project, in order to assess the trade-off between accuracy and efficiency in state-of-the-art object detection models, we compared YOLOv5-X (i.e. the largest model from YOLOv5) with YOLOv8-N (i.e. the smallest model from YOLOv8). While YOLOv5-X is known for its speed and robustness, our initial evaluation suggests YOLOv8-N offers more superior performance. As expected, YOLOv5-X has the potential for higher accuracy due to its larger size,

however, YOLOv8-N, despite being a smaller model, appears to demonstrate comparable or even better accuracy *(Refer to Table 6)*. These improvements are likely to stem from YOLOv8-N's architectural refinements, including optimised layer configurations and the adoption of Automatic Mixed Precision (AMP).

Table 4. Evaluation of YOLOv5-X and YOLOv8-N Models

Model	F1 Score	Precision	Recall	mAP@0.50[1]
YOLOv5-X	0.597	0.621	0.574	0.660
YOLOv8-N	0.783	0.797	0.770	0.831

[1] In object detection, mAP50 is a common metric that summarizes a model's overall performance across various detection difficulties by considering both precision and recall at this specific IoU 50% threshold.

Overall, YOLOv8-N seems to perform better than YOLOv5-X in terms of both precision and recall. It also has a higher mAP@0.5 score (0.660 vs 0.831), indicating its overall detection accuracy might be better. From this initial evaluation, it would appear that there is definitely more room for improvement especially for YOLOv8-N and it would be beneficial to finetune the model's hyperparameters to look at enhancing its performance.

5.3. Optimising YOLOv8-N

For this finetuning, we focus on 3 key components comprising optimiser, learning rate and batch sizes to further refine YOLOv8-N's performance for brain tumor classification.

Table 5. Optimising YOLOv8-N for the Best Model Performance

Model	F1 Score	Precision	Recall
Adam, LR1 e-2,4Batch	0.764	0.785	0.745
SGD, LR1 e-2,4Batch	0.884	0.901	0.868
AdamW,LR1 e-2,4Batch	0.903	0.906	0.901

Based on the Table 5, the most optimal set of configuration was identified to be: Auto optimizer (AdamW), batch size of 4, and a learning rate of 0.02. This configuration achieved the best F1-score of 0.903, indicating fairly good performance and the model also showed a favorable balance between precision and recall. The optimised model demonstrated promising capabilities in predicting brain tumor class with varying confidence levels, highlighting its ability to detect and classify tumors in MRI scans.

6. Assessment of Models and Recommendation

6.1. Summary of the 3 Selected Models

The project aimed at enhancing the accuracy of brain tumor detection through machine learning has culminated in the evaluation of three CNN models: Vision Transformer (ViT), U-Net combined with DenseNet-121, and YOLO. Table 6 shows the comparative summary of their performances based on precision, recall, and F1 scores:

Table 6. Summary of 3 selected models

Model	F1 Score	Precision	Recall
Vision Transformer (ViT)	0.837	0.859	0.822
U-Net + DenseNet-121	0.901	0.901	0.901
YOLOv8n	0.903	0.906	0.901

The Vision Transformer (ViT) is commendable for its detailed contextual image analysis, achieving an F1 score of 0.837. This model is particularly suited to research-driven applications where comprehensive image examination is paramount, although its deployment requires significant computational investment.

The combined U-Net and DenseNet-121 model, with its F1 score of 0.901, offers an integrated approach to brain tumor detection, balancing the strengths of both architectures to provide precise image segmentation and classification. If we integrate the top-performing versions of our models into a combined medical imaging pipeline, the overall model may not reach the required standard due to the segmentation model's limitations. To address this, we must refine U-Net through additional fine-tuning, dataset expansion, and extended training to ensure clinical viability.

YOLOv8n, with its agility and efficiency, c aters w ell to the pressing demands of clinical scenarios, evidenced by an F1 score of 0.903, slightly above U-Net and DenseNet-121's score. It is optimal for environments where time is a factor, and rapid diagnosis can significantly influence patient outcomes .

6.2. Assessment and Recommendation

Each model reviewed brings distinct strengths to medical imaging applications. The Vision Transformer excels in detailed image analysis due to its ability to process contextual relationships within the image, making it suitable for complex diagnostic tasks where deep insights are critical. However, it requires substantial computational resources and extensive training times, which may not be feasible in all clinical settings. The combination of U-Net and DenseNet-121 is highly effective for tasks requiring precise image segmentation followed by accurate classification, ideal for structured diagnostic environments. Yet, like ViT, they demand significant computational power and processing time.

YOLOv8n, known for its speed and efficiency in real-time detection, offers a viable option for emergency situations where rapid diagnosis is essential. Although it generally provides less precision than the other models, its ability to quickly process images makes it invaluable in time-sensitive

scenarios. When recommending models, YOLOv8n is best for fast-paced environments, U-Net with DenseNet-121 for accuracy-critical settings, and ViT for research-focused applications where computational resources are abundant.

The primary limitations of these models are their computational demands and the operational costs associated with them. Implementing ViT and U-Net with DenseNet-121 in resource-limited settings could be challenging due to their requirements for advanced hardware and lengthy training periods. Additionally, while YOLOv8n's faster processing capability is advantageous, the potential compromise on accuracy must be carefully managed, particularly in critical medical diagnostics.

6.3. Conclusion

The deployment of machine learning models has significantly advanced the field of medical imaging and diagnostics, including the detection and classification of brain tumors. The evaluation of three sophisticated models - Vision Transformer (ViT), U-Net combined with DenseNet-121, and YOLOv8n - highlights their distinct strengths and suitability for different diagnostic requirements and settings.

The Vision Transformer excels in complex diagnostic scenarios where deep contextual analysis of imaging data is crucial, making it well-suited for research environments and specialized medical applications (Xie et al., 2022). The combined approach of U-Net and DenseNet-121 is ideal for applications where precise segmentation and detailed classification are paramount, ensuring high diagnostic accuracy essential for effective treatment planning (Anaya-Isaza & Mera-Jiménez, 2022). YOLO, with its remarkable speed and efficiency, addresses the urgent needs of clinical environments where rapid tumor detection can significantly impact patient outcomes (Soomro TA, 2022).

The adoption of machine learning in medical imaging presents both opportunities and challenges for healthcare businesses. While machine learning algorithms can assist in making more accurate and efficient diagnoses, leading to improved patient outcomes and reduced costs, the implementation also comes with risks, such as the potential for biased or unreliable predictions. When selecting a machine learning model, healthcare institutions should carefully evaluate the tradeoffs and ensure appropriate governance and oversight to mitigate these risks

Overall, the application of these models can lead to earlier disease detection and more accurate diagnoses, with the added benefits of streamlined workflows and cost efficiencies. Nonetheless, this technological leap demands careful consideration of biases, ethical standards, and the need for transparent model predictions. A balanced, multidisciplinary approach, encompassing clinical, imaging, data science, and regulatory expertise, is imperative to ensure these models are used responsibly and effectively within the medical field, mitigating risks while maximizing patient care and operational benefits.

Code and Data

The code used for this project is available on GitHub. For access to the source code, additional resources, and documentation, please visit our GitHub repository: github.com/gabigarms/BT5153-Final-Project.

References

Anaya-Isaza, A. and Mera-Jiménez, L. Data augmentation and transfer learning for brain tumor detection in magnetic resonance imaging. *IEEE Access*, 10:23217–23233, 2022.

Hicks, S. A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M. A., Halvorsen, P., and Parasa, S. On evaluation metrics for medical applications of artificial intelligence. *Scientific Reports*, 12(1):5979, 2022. ISSN 2045-2322.

Rostami, A. Labeled mri brain tumor dataset. https://universe.roboflow.com/ali-rostami/labeled-mri-brain-tumor-dataset, Feb 2024. Accessed: 2024-04-22.

Soomro TA, Zheng L, A. A. A. A. S. S. Y. M. G. J. Image segmentation for mr brain tumor detection using machine learning: A review. *IEEE Reviews in Biomedical Engineering*, 16:70–90, 2022.

Xie, Y., Zaccagna, F., Rundo, L., Testa, C., Agati, R., Lodi, R., Manners, D., and Tonon, C. Convolutional neural network techniques for brain tumor classification (from 2015 to 2022): Review, challenges, and future perspectives. *Frontiers in Oncology*, 12:9406354, 2022. doi: 10.3389/fonc.2022.9406354.