ASEAN Engineering Journal

Full Paper

IMPROVING THE EFFICIENCY OF DETECTING AND CATEGORIZING BRAIN TUMORS FROM DIVERSE MRI IMAGES THROUGH THE IMPLEMENTATION IN DEEP CONVOLUTION NEURAL NETWORK

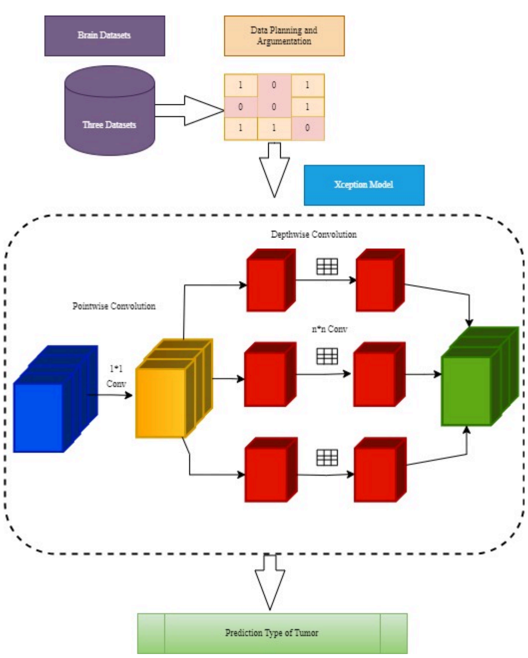
Thein Aung Tin[a]*, Hla Myo Tun[b], Ei Ei Khin[b], Mya Mya Aye[a]

[a]Department of Electronic Engineering, Yangon Technological University, Yangon, Myanmar
[b]Department of Research, Yangon Technological University, Yangon, Myanmar

Article history
Received
19 July 2024
Received in revised form
07 March 2025
Accepted
12 March 2025
Published online
30 November 2025

*Corresponding author
theinaungtin@gmail.com

Graphical abstract



Abstract

Deep learning is a popular and effective approach to medical imaging detection and classification. Various brain tumor classifications are related to an ongoing research topic made possible by the diversity of cancer features. Local brain images testing in the latest brain images datasets BRAT can be challenging due to their processing time, lower accuracy, and the possibility of overfitting and underfitting. In this paper, Xception DCNN model can be implemented to test the local images or real images and apply three datasets in this model, analyse the individual result and improve the training speed and stability of neural networks by normalizing the activations of each layer. The Xception model can significantly reduce the number of parameters and computational complexity without compromising the model's accuracy. Moreover, this system used popular benchmark datasets such as Kaggle and BRATS, the suggested research MRI image datasets were assessed, and the local validation dataset was proposed. In this research work, this model achieved the best performance results with training accuracy and loss 99±0.005%, validation accuracy and loss 98±0.2%. The main objectives are improving the accuracy of classification, avoiding overestimation along with underestimating, and decreasing down on processing time.

Keywords: Brain Tumor, Local Brain images, Xception DCNN model, MRI, Classification accuracy

1.0 INTRODUCTION

Worldwide, brain tumor is considered a crucial component of serious cancers. The term "brain tumor" refers to abnormal cell growth around the brain, which falls under the two categories of no tumor and various types of tumor. The mass or lump produced through this unregulated growth of cells as well as collection could disrupt natural neural activity. It is known that there are two major categories of brain tumors: development types tumor and initial stage tumors. The minuscule details within the body can be found using MRI scanning [1].

It is possible to find brain cancers by diagnostic image processing or biopsy as well. CT and MRI imaging constitute different kinds of brain tumor identification techniques applied for brain cancer treatment [2]. Brain tumor detection is thought to represent the end of the test, and a biopsy observation is required [3].

In literature, it is stated that CNN architectures were modified with genetic algorithms (GA), five convolutional layers and pooling were constructed in the feature extraction, dense layer and classification layer, and an accuracy of 90.9 in gliomas and 94.2 in four tumor types was achieved. This research is based on online accessible datasets for research purposes and was required to improve the CNN architecture due to the lack of accuracy [4]. CNN classifier was modified to compare the activation function and the techniques with and without regulation, and to achieve an overall accuracy of 96 using the Brats (2018) open access dataset. It was necessary to improve the classification accuracy and reduce the computational cost of the classifier[5]. The CapsNets CNN classifier was used for segmentation and optimization by using a routing iteration in the hyperparameters. Two convolutional layers and matching routing were combined and an accuracy of 90.89 was achieved[6]. CNN architectures were proposed in three stages: data acquisition, preprocessing and classification. It was constructed with convolution, activation and fully connected layers and achieved an accuracy of 92%. This model had a lack of accuracy in detection and classification in CNN[7]. A new CNN model for classifying four tumor types was proposed, which was built with convolution, activation, drop out, max pooling and dense layer and achieved an 96% of accuracy[8]. The algorithm was involved in five steps: pre-processing the data and extracting the features, using two pre-trained models, applying the Extreme Learning Machine (ELM) for feature learning, converting to Partial Least Square (PLS) in a matrix and then classifying. 93% accuracy was achieved in the Brats 2018 datasets[9]. Two classifiers such as SVM and ANN were proposed for the extraction in preprocessing and fuzzy C-Means clustering for the determination of tumor location. It achieved 91% accuracy in tumor classification[10]. Two feature extraction experiments were modified by using machine learning classifiers such as ANN, KNN, Rf and LDA. The open access Kaggle dataset was used for training images and local images for testing and achieved an overall accuracy of 95[11].22 CNN layers were implemented for tumor detection and classification built with convolution, Relu-activation, max-pooling, normalization, fully connected layer, and SoftMax layer, and achieved 96% accuracy for four tumor types[12].

In the previous study, both MRI and CT scans are most frequently used to diagnose brain cancers[13], MRI is more useful [14]because it analyzes the anatomical portion of brain images to point out the location and characteristics of tumors in high-resolution structure images [15]. The previous approach experienced two primary challenges as a result of problems with medical images or medication data: scalability up to bigger collections due to computation limits or difficulties and applying to new or unfamiliar information. Moreover, the DCNN model was applied to diverse data, such as real local images. This study required the application of different CNN models to maximize classification accuracy on a variety of datasets. This conventional method is well suited for categorization and analysis of structured data.

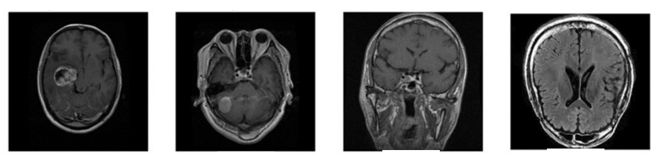
In this model, improved dataset representation helps avoid situations where overfitting or underfitting leads to mismatches between objective and expected outcomes. To improve the training performance and convergence, this study proposes to modify Deep CNN and use the normalization technique. A layer's inputs must be well adapted following the normalization process by scaling and modifying the activations. The data being entered provides a more rapid and accurate evaluation method for the overall system through input normalization. This paper creates the proposed novel DCNN model, applies the three diverse datasets, and adds local validation images to the testing datasets, and minimize the total processing time and total inference time.

2.0 DATA PLANNING AND MODEL SELECTING

Three datasets for data planning are being implemented in this model. The three main stages include collecting datasets, pre-processing and normalizing data, and augmenting data. This approach allows the Xception model to capture spatial features and channel-wise correlations more efficiently.

2.1 Datasets Availability

Tumor classification is widely used in BRATS datasets [16] and local datasets [17] in individual datasets. Kaggle BRATS datasets are easy and open-access datasets are easily available to download on the Kaggle website [18] [19]. Three datasets are applied in this system. Dataset 1 includes 7023 MR pictures of the brain from the collected data, comprising 2000 photos without brain tumors and 1621, 1645, and 1757 images with glioma, meningioma, and pituitary tumors as shown in Figure 1. Brats, Kaggle datasets add additional images regularly, and an especially recent dataset. Dataset 2 comes from the detection of brain tumor images in the MRI Department of the Defense Service of the Orthopedic Hospital. It captured the brain images in three styles: T1-weighted contrast, T2-weighted contrast, and flair-enhanced MRI images applied to the testing dataset[20].



a b c d

Figure 1 Four types of Brain Tumor for this Experiments (a) Glioma Tumor (b) Meningioma Tumor (c) Pituitary Tumor (d) No Tumor[16]

Dataset 3 split local brain tumor images into the validation and testing datasets to achieve the best prediction. This model is used to classify the local MRI image in testing data from the local image collection. It collects images for four types of tumors, like 9 images in the pituitary tumor, 47 images in the no tumor, 10 images in the meningioma tumor, and 6 images in the glioma tumor from 72 patients.

2.2 Data preprocessing and Argumentation

For the purpose of online datasets, T1 and T2 weight MRI images are widely used [21], and some datasets have obtained the flair dataset. Eight techniques can be studied for the detection of MRI images in real life. There are T2 FAT SAT AX, T1 AX, FLAIR AX, DWI AX, FLAIR SAG, T2 COR, T2* AX, and DWI analysis for image preprocessing. These popular algorithms are applied in detection systems and can be selected and extracted from different types of MRI images[22]. After that, collect the type of tumor and images applied to individual testing datasets. Three experiments are applied for this research: the first experiment is using online access training and testing datasets;

the next training dataset is higher-quality images from the research-purpose online datasets, and the testing dataset is local MRI images; the last one is splitting the validation and testing images into testing datasets for the purpose of higher efficiency.

2.3 Xception DCNN model

The Xception model is used to train medical image dataset, which provides it like a robust capability to recognize a wide variety of features from natural images, making it highly effective for transfer learning in diverse image classification tasks in Figure 2.

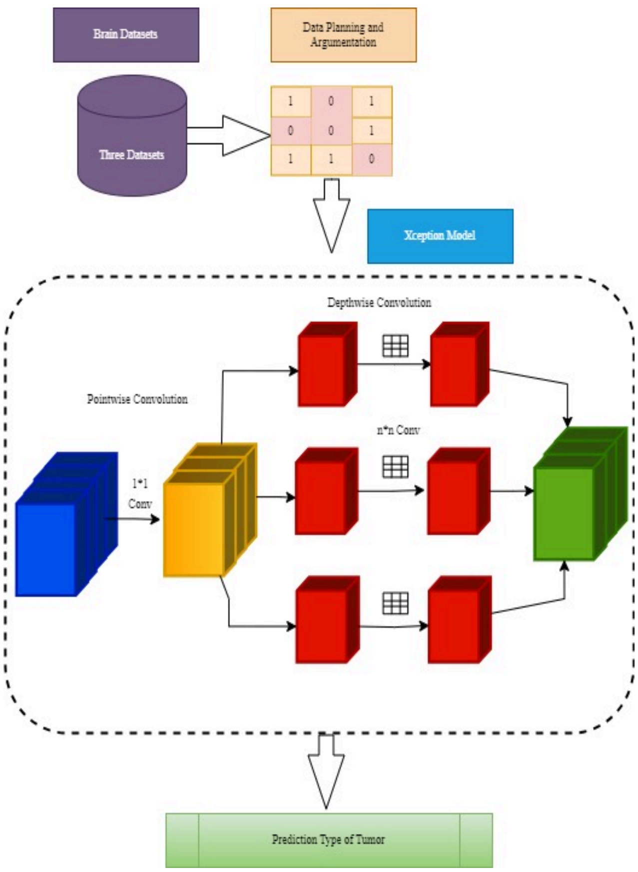


Figure 2 Xception model using Depthwise Separable Convolution algorithms[16][23]

To choose an effective image classification model, several key steps must be undertaken. First, import essential libraries such as TensorFlow, Keras, and relevant Keras layers to provide the necessary functionalities for model construction and training. Next, apply the architecture from the previous datasets, eliminating the unwanted feature, to leverage its powerful feature mapping resources. To modify this architecture for a precise estimation process, calculation is undertaken in every step: a flatten layer to convert the 2D matrix output from Xception into one-dimension path; unnecessary feature subtraction layer to reduce the previous unwanted value to get the zero value in the running condition; after neural network using ReLU activation to introduce non-linearity; and another feature reduction algorithm. The last layer is used SoftMax to estimate the four outputs, corresponding to the four target classes. This architecture is collected with lower cost value, optimizer, and performance metrics to prepare it for training. Finally, the model is trained on the provided dataset, adjusting weights to minimize the loss function and enhance accuracy through iterative learning. This systematic approach ensures the development of a robust and efficient model tailored to the specific classification task. The proposed architectures were constructed step by step, as shown in Figure 3.

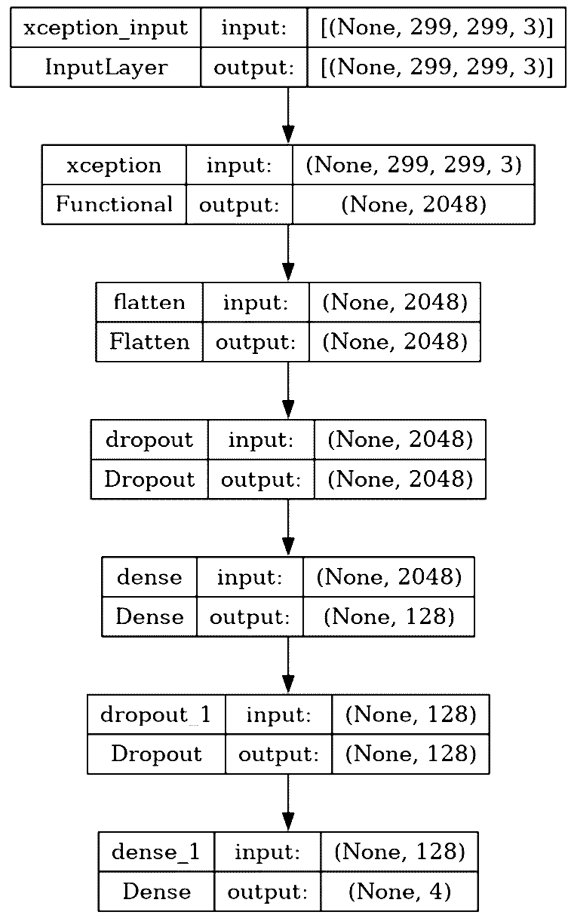


Figure 3 Exception Step by Step in Xception model

The major advantage of this model is to improve efficiency compared with the previous model by using depthwise separable 1*1 convolutions. This can reduce computation cost and processing time and obtain high accuracy. And then the model can extract features and moderate overfitting and underfitting more than the previous CNN model.

3.0 EXPERIMENTAL RESULTS

Three experiments are applied in this classification. The first test was with the latest brain tumor dataset, such as Kaggle, and then the next one was applied to a local image from the local hospital in testing datasets. The last one was splitting the local images into validation and testing datasets. The proposed novel Xception model was tested and demonstrated lower cost and higher accuracy during propagation. Two combination datasets are applied in Colab Notebooks software such as Kaggle BRAT datasets, and local image validation datasets.

3.1 Apply Kaggle BRATS Datasets to The Xception Model

The present study employed the new Xception model by hyperparameter turning, including the number of epochs, to optimize the estimation efficiency and minimize cost in different datasets. The goal of predictions loss was minimized and its optimization efficiency was optimal for this architecture, as shown in Figure 4. In the final results, the optimization loss (0.005) and target outcomes (99.98%) were nearly equivalent. In the estimation results, the optimization loss (0.03) and target outcomes (98.98%) were nearly equivalent.

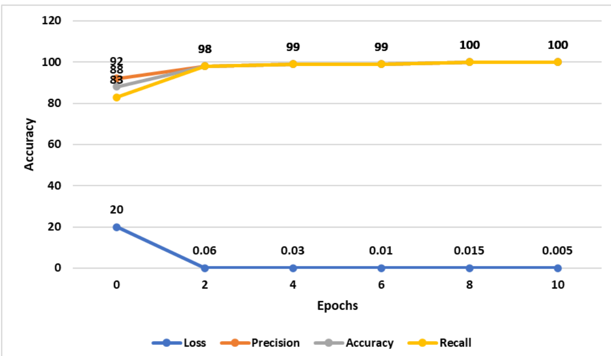


Figure 4 Model Training Metrics Over Epochs

This training dataset shows the performance metrics of a deep learning model over 10 epochs, focusing on loss, precision, accuracy, and recall. Initially, the model starts with a high loss value of 20, which dramatically decreases to 0.0 by the second epoch and continues to decline, reaching a minimal value of 0.005 by the tenth epoch. This significant reduction in loss indicates substantial improvement in the architecture's achievement. Concurrently, four estimation accuracy show a positive estimation results.

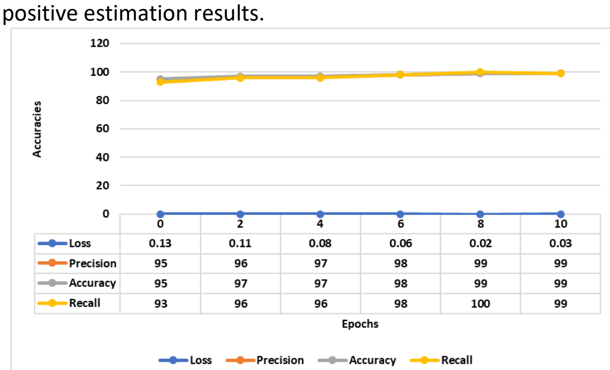


Figure 5 Model Validation Metrics Over Epochs

Precision begins at 85% and quickly rises to 97% by the second epoch, eventually reaching 100% by the tenth epoch. Accuracy follows a similar trajectory, starting at 80% and achieving 100% by the tenth epoch. Recall, which starts at 90%, also reaches 100% by the tenth epoch. As training datasets, the loss consistently decreases from 0.13 at epoch 0 to 0.03 at epoch 10, indicating improving model performance. Precision and accuracy both start at 95% and gradually increase to 99% by epoch 10, demonstrating enhanced model reliability in making correct predictions. Recall shows a similar upward movement, starting at 93% and peaking at 100% by epoch 8 before slightly dipping to 99% at epoch 10 as shown in Figure 5.

These metrics collectively illustrate that the model undergoes significant improvement in its predictive capabilities with continued training, achieving high levels of accuracy, precision, and recall while minimizing loss. This improvement highlights the model's robustness and effectiveness in handling the given dataset, suggesting it is well-tuned and capable of maintaining high performance across various evaluation metrics.

3.2 Apply Local Testing Datasets to The Xception Model

In order to acquire the estimation efficiency and minimize cost in different datasets. the DCNN model was trained using the following hyperparameters: weight and bias, type of filter, and number of parameter. Figure 6 and 7 demonstrates that the last epochs predicted for this DCNN model were 0.005, 99.89%, 4.75, and 62%, respectively.

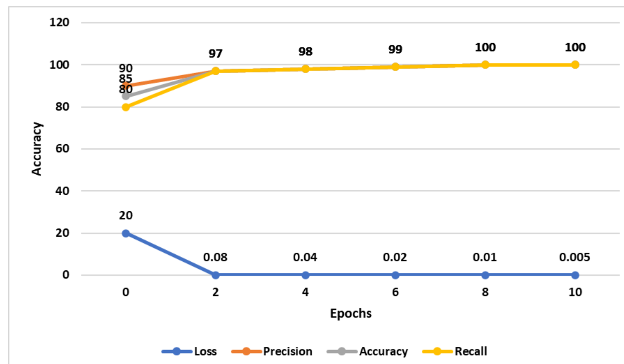
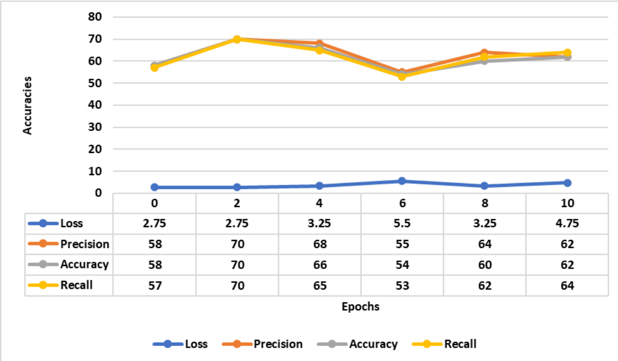


Figure 6 Model Training Metrics Over Epochs

Firstly, the model starts with a high loss value of 20, which dramatically decreases to 0.08 by the second epoch and continues to decline, reaching a minimal value of 0.005 by the tenth epoch. This significant reduction in loss indicates substantial improvement in the model's performance. Precision begins at 85% and quickly rises to 97% by the second epoch, eventually reaching 100% by the tenth epoch.

Accuracy follows a similar trajectory, starting at 80% and achieving 100% by the tenth epoch. Recall, which starts at 90%, also reaches 100% by the tenth epoch. This model becomes increasingly reliable and effective in its predictions as training progresses, achieving perfect scores in precision, accuracy, and recall by the end of the training period. The convergence of these performance metrics towards optimal values demonstrates the model's successful learning and adaptation over the epochs.



	0	2	4	6	8	10
Loss	2.75	2.75	3.25	5.5	3.25	4.75
Precision	58	70	68	55	64	62
Accuracy	58	70	66	54	60	62
Recall	57	70	65	53	62	64

Figure 7 Model Validation Metrics Over Epochs

This testing dataset illustrates the variation in performance metrics—loss, precision, accuracy, and recall—over 10 epochs. Initially, the model starts with a loss of 2.75 and precision, accuracy, and recall all at 58, 58, and 57, respectively. By the second epoch, precision, accuracy, and recall significantly improve to 70, while loss remains constant at 2.75. At the fourth epoch, precision and accuracy slightly decrease to 68 and 66, respectively, while recall drops to 65, and loss climbs to 3.25. A marked decline is observed at the sixth epoch, where precision, accuracy, and recall fall to 55, 54, and 53, respectively, accompanied by an increased loss of 5.5. However, these metrics recover by the eighth epoch, with precision, accuracy, and recall rising to 64, 60, and 62, respectively, and loss decreasing to 3.25. By the tenth epoch, the metrics stabilize, with precision and accuracy both at 62, recall at 64, and loss at 4.75. These fluctuations suggest the model experiences overfitting and underfitting phases but shows a general trend of stabilization towards the latter epochs. The target and prediction results displayed a large overfitting and underfitting in this dataset.

3.3 Apply Splitting Local Images in Validation And Testing Datasets to the Xception model

This study employed the new Xception model by hyperparameter turning, including the number of epochs, to optimize the estimation efficiency and minimize cost in different datasets. The goal of predictions loss was minimized and its optimization efficiency was optimal for this architecture, as shown in Figure 8.

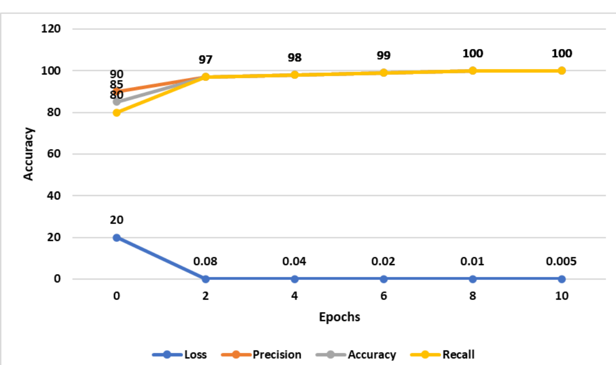
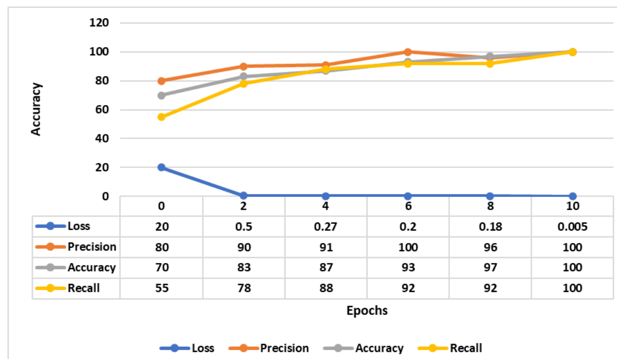


Figure 8. Model Training Metrics Over Epochs

In the estimation results, the optimization loss (0.005) and target outcomes (99.98%) were nearly equivalent. The training dataset presents the performance metrics of a deep learning model over 10 epochs, focusing on loss, precision, accuracy, and recall. Initially, the model exhibits a high loss value of 20, which rapidly decreases to 0.08 by the second epoch and continues to decline, reaching a minimal value of 0.005 by the tenth epoch. This reduction in loss indicates a significant improvement in the estimation process. Precision starts at 85% and quickly rises to 97% by the second epoch, eventually reaching 100% by the tenth epoch. Accuracy follows a similar range, starting at 80% and achieving 100% by the tenth epoch. Recall, which begins at 90%, also reaches 100% by the tenth epoch. These movements suggest that the model becomes increasingly reliable and effective in its predictions as training progresses, achieving perfect scores in precision, accuracy, and recall by the end of the training period. The convergence of these performance metrics towards optimal values demonstrates the model's successful learning and adaptation over the epochs.



	0	2	4	6	8	10
Loss	20	0.5	0.27	0.2	0.18	0.005
Precision	80	90	91	100	96	100
Accuracy	70	83	87	93	97	100
Recall	55	78	88	92	92	100

Figure 9 Model Validation Metrics Over Epochs

This dataset shows the performance metrics of a deep learning model over 10 epochs, highlighting the movements in loss, precision, accuracy, and recall. Initially, the model starts with a high loss value of 20, which significantly decreases to 0.005 by the 10th epoch, indicating improved model performance. Concurrently, precision, accuracy, and recall exhibit a positive trend. Precision starts at 80% and reaches 100% by the 10th epoch, while accuracy improves from 70% to 100%. Recall shows the most substantial increase, starting at 55% and also reaching 100% by the final epoch. These metrics suggest that the model becomes more reliable and effective in its predictions as training progresses, achieving perfect scores in precision, accuracy, and recall by the end of the training period. This convergence of performance metrics towards optimal values demonstrates the model's successful learning and adaptation over the epochs. The goal of predictions loss was minimized and its optimization efficiency was optimal for this architecture, as shown in Figure 9. In the prediction findings, the optimization loss (0.005) and target outcomes (99.98%) were nearly equivalent.

3.4 Comparison and Discussion With Previous Work

This model used several hidden layer types and hyperparameter adaptations to apply Kaggle and local datasets in a proposed architecture. Within testing experiment in Kaggle datasets, this architecture achieved the best results in training

(99%) and testing (98%) accuracy. After the testing experiment on local validation datasets, this model obtained acceptable results in training (100%) and testing (100%) accuracy as shown in Figure 10. The model's reliance on weight initialization was reduced by the use of normalization techniques. Higher learning rates are possible without reducing the ability to adjust during training. It can eliminate the need for an additional regularization procedure and overfitting.

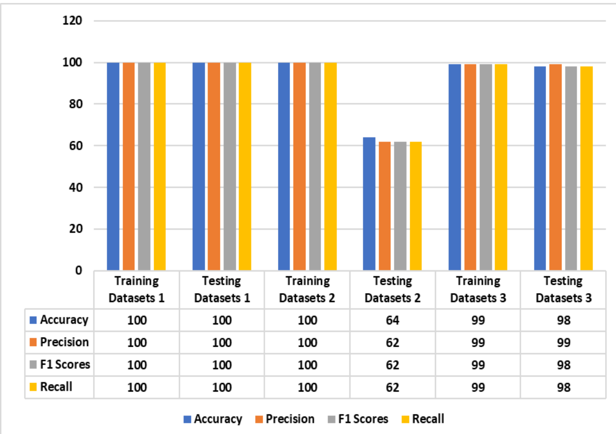


Figure 10 Performance loss in Xception Model

This research approach is to obtain higher accuracy in local image predictions, like that standard dataset. The first dataset can easily be downloaded from the BRAT website, constructed with training and testing datasets. The second dataset, Dataset 2, is included with standard datasets in training and local images in testing for the purpose of AI to predict any type of raw images, but the testing result achieved lower accuracy. After that, changing the splitting algorithm in dataset 2 and adding validation images in training datasets, formed the new dataset which is also called dataset 3. The main reason is to achieve higher prediction performance in local images.

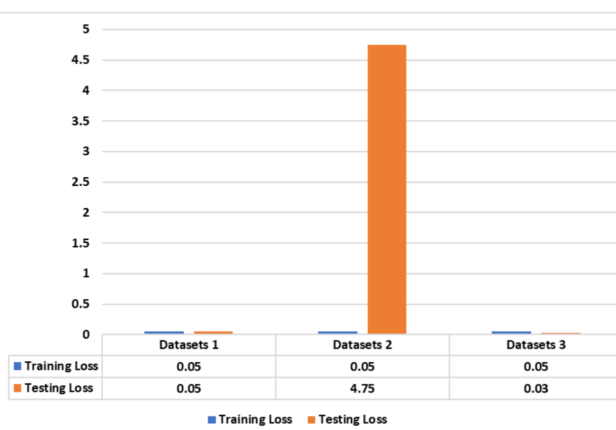


Figure 11 Total processing time and inference time in Kaggle and local datasets

Three dataset illustrates the performance metrics, including accuracy, precision, F1 scores, and recall, across three different training and testing datasets. For Training Datasets 1, the performance metrics are perfect, with all values at 100. Testing Datasets 1 also exhibit ideal performance, maintaining 100 across all metrics. In contrast, Training Datasets 2 display consistent perfect scores of 100, while Testing Datasets 2 show a significant drop, with accuracy at 64, precision at 62, F1 scores at 62, and recall at 62. This testing datasets based on local images only. Training Datasets 3 show high performance with accuracy, precision, F1 scores, and recall at 99. Testing Datasets 3 demonstrate a slight decrease in performance metrics, with accuracy at 98, precision at 99, F1 scores at 98, and recall at 98. This analysis indicates a notable decline in performance from the training to the testing phase, especially for Datasets 2, highlighting potential issues with overfitting or dataset variability.

This model presents the training and testing loss values across three different datasets. For Datasets 1 and 3, both the training and testing losses are relatively low and comparable, each with a training loss of 0.05 and testing losses of 0.05 and 0.03, respectively. In stark contrast, Datasets 2 demonstrates a significant disparity between training and testing losses while the training loss remains low at 0.05, the testing loss escalates dramatically to 4.75 as shown in Figure 11. This substantial increase in testing loss for Datasets 2 suggests a severe overfitting issue, designating that this architecture has misclassified in unseen images and real local datasets. Such a discrepancy underscores the necessity for better regularization or more robust validation techniques to enhance model generalization. The Xception model can deliver real-time insights by minimizing this time, which is crucial for applications such as image recognition and object detection. This model can optimize the results and processing time by applying it to different datasets, but it needs to add validation images.

Table 1 Accuracy Comparison between previous and proposed model

Reference	Classifier/Algorithm	MRI Protocol	Accuracy %
[4]	CNN architecture	Brain	**94**
[5]	CNN classifier	Brain	**96**
[6]	CapsNets CNN classifier	Brain	**91**
[7]	CNN architectures	Brain	**92**
[8]	New CNN model	Brain	**96**
[9]	Two pre-trained models (ELM and PLS)	Brain	**93**
[10]	fuzzy C-Means clustering	Brain	**91**
[11]	Machine learning classifiers	Brain	**95**
[12]	22 CNN layers in CNN classifier	Brain	**96**
Proposed Xception model in Kaggle dataset 1		Brain	**99**
Proposed Xception model in Local dataset 3		**Brain**	**99**

Table 1 compares the accuracy of the previous classifier and the proposed classifier. The previous classifier achieved (81 - 96%) accuracy, while the proposed classifier achieved 99% accuracy.

The table shows the results of dataset 1 and dataset 3. The experiment of dataset 2 achieved lower prediction performance in testing images. Dataset 2 and Dataset 3 are the same training and testing images except for the validation images in Dataset 3.

The previous traditional CNN model used a larger layer, a high number of parameters, skip connections, and residual blocks. The performance of previous models has lower efficiency, complex feature extraction, and support in simple image classification. The proposed model has fewer parameters, more efficient convolution, and support in medical image classification.

In this research, the new Xception model is discussed and applied to three datasets, such as open-access BRAT datasets for research and local validation datasets. During the same 10-epoch time frame, model achieved the best optimization possible in terms of accuracy and loss between the target and prediction outcomes in the two datasets. Furthermore, reduced processing time optimizes resource utilization and facilitates deployment on edge devices with limited computational capacity. This proposed Xception model achieves higher accuracy and lower computation costs because it is constructed with optimization techniques in the hidden layer.

4.0 CONCLUSION

This research uses advanced new convolution neural network Xception architecture with an optimal normalization strategy to automatically identify various brain cancers from collected MRI images. Individual blocks are used in this architecture. Convolution, normalization, pooling, Dense, and SoftMax layers that make up this model. ReLU stands for boundary function. With 99.89% validation accuracy, 0.2 validation loss, 0.05 training loss, and 99.89% training accuracy, this model has produced the best results for the upgraded dataset. This suggested model outperforms the conventional CNN model since it is committed to identifying brain tumors with the greatest convenience and the lowest loss. This research constructed larger datasets by utilizing the two resources, Kaggle BRATS datasets and local validation datasets. With the consultation of the physician, only the patient IDs were used considering patient privacy concerns and ethical issues in medical diagnostics. This research helps neurologist by easily classifying different types of tumors. Better medical diagnostic results and quicker patient treatment support are the goals of this research. These plans are to build more extensive and varied training datasets in the future using both more local images and open-access images. Then, the models on the local images will be tested to ensure real-time performance.

Acknowledgement

To be able to complete this work—a collaborative investigation with the MRI Department of 500-bedded Orthopedic Defense Service hospital in Myanmar—the author is thankful for numerous classmates from the Biomedical Electronics Engineering Research Group within the Faculty of Electronic Engineering at Yangon Technological University.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] Z. Zulkoffli and T. A. Shariff, 2019. "Detection of Brain Tumor and Extraction of Features in MRI Images Using K-means Clustering and Morphological Operations," *2019 2019 IEEE International Conference on Automatic Control and Intelligent Systems. I2CACIS 2019 - Proceeding.* 1–5, DOI: 10.1109/I2CACIS.2019.8825094.

[2] D. Suresha, N. Jagadisha, H. S. Shrisha, and K. S. Kaushik, 2020, "Detection of Brain Tumor Using Image Processing," *Proceedings of the 4th International Conference on Computing Methodologies and Communication. ICCMC 2020*. 844–848, DOI: 10.1109/ICCMC48092.2020.ICCMC-000156.

[3] W. Wang, F. Bu, Z. Lin, and S. Zhai, 2020. "Learning Methods of Convolutional Neural Network Combined with Image Feature Extraction in Brain Tumor Detection," *IEEE Access*, 8: 152659–152668, DOI: 10.1109/ACCESS.2020.3016282.

[4] A. Kabir Anaraki, M. Ayati, and F. Kazemi, 2019. "Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms," *Biocybernetics and Biomedical Engineering*, 39(1): 63–74, DOI: 10.1016/j.bbe.2018.10.004.

[5] H. Mzoughi *et al.*, 2020. "Deep Multi-Scale 3D Convolutional Neural Network (CNN) for MRI Gliomas Brain Tumor Classification," *Journal of Digital Imaging*, 33(4): 903–915, DOI: 10.1007/s10278-020-00347-9.

[6] Parnian Afshar, K. Plataniotis, Arash Mohammadi. 2018. "Capsule Networks for Brain Tumor Classification Based on MRI Images and Coarse Tumor Boundaries," *IEEE International Conference on Acoustics, Speech, and Signal Processing*

[7] A. H. Khan *et al.*, 2022, "Intelligent Model for Brain Tumor Identification Using Deep Learning," *Applied Computational Intelligence and Soft Computing*, 2022, Article ID 8104054. DOI: 10.1155/2022/8104054.

[8] M. M. Badža, 2020. "Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network," *Applied sciences 10(6): 1999. DOI: https://doi.org/10.3390/app10061999*

[9] M. A. Khan *et al.*, 2020, "Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists," *Diagnostics*. 10(8): 1–19, DOI: 10.3390/diagnostics10080565.

[10] T. P, B. Bhat, and K. Prakash, 2019, "Detection and Classification of Tumour in Brain MRI," *International Journal of Engineering and Manufacturing*, 9(1): 11–20, DOI: 10.5815/ijem.2019.01.02.

[11] K. Kaplan, Y. Kaya, M. Kuncan, and H. M. Ertunç, 2020, "Brain tumor classification using modified local binary patterns (LBP) feature extraction methods," *Medical Hypotheses*, 139 DOI: 10.1016/j.mehy.2020.109696.

[12] M. F. Alanazi *et al.*, 2022, "Brain Tumor/Mass Classification Framework Using Magnetic-Resonance-Imaging-Based Isolated and Developed Transfer Deep-Learning Model," *Sensors*, 22(1), DOI: 10.3390/s22010372.

[13] M. I. Sharif, J. P. Li, J. Naz, and I. Rashid, 2020, "A comprehensive review on multi-organs tumor detection based on machine learning," *Pattern Recognition Letters* 131: 30–37, DOI: 10.1016/j.patrec.2019.12.006.

[14] V. K. Waghmare and M. H. Kolekar, 2021. Brain Tumor Classification Using Deep Learning, *Internet of Things for Healthcare Technologies* 73: 155–175. Springer Singapore, DOI: 10.1007/978-981-15-4112-4_8.

[15] M. Alshayeji, J. Al-buloushi, A. Ashkanani, and S. Abed, 2021. "Enhanced brain tumor classification using an optimized multi-layered convolutional neural network architecture," *Multimedia Tools and Applications*,19: 28897-28917 DOI: 10.1007/s11042-021-10927-8 Springer Nature

[16] G. Karayegen and M. F. Aksahin, 2021, "Brain tumor prediction on MR images with semantic segmentation by using deep learning network and 3D imaging of tumor region," *Biomedical Signal*

Processing and Control, 66, DOI: 10.1016/j.bspc.2021.102458.

[17] R. Hao, K. Namdar, L. Liu, and F. Khalvati, 2021, "A Transfer Learning–Based Active Learning Framework for Brain Tumor Classification," *Frontiers in Artificial Intelligence*, 4: 1–11, DOI: 10.3389/frai.2021.635766.

[18] S. B. Fayyadh and A. A. Ibrahim, 2020, "Brain Tumor Detection and Classifiaction Using CNN Algorithm and Deep Learning Techniques," *3rd International Conference on Advanced Science and Engineering, ICOASE 2020*, 157–161, DOI: 10.1109/ICOASE51841.2020.9436599.

[19] "Brain Tumor MRI Dataset." Accessed: Apr. 25, 2024. [Online]. Available: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

[20] Thein Aung Tin, 2024. "BrainU." Accessed: Jun. 18, 2024. [Online]. Available: https://www.kaggle.com/datasets/minthein/brainu

[21] T. Sadad *et al.*, 2021, "Brain tumor detection and multi-classification using advanced deep learning techniques," *Microscopy Research and Technique*, 84(6): 1296–1308, DOI: 10.1002/jemt.23688.

[22] D. Rammurthy and P. K. Mahesh, 2022, "Whale Harris hawks optimization based deep learning classifier for brain tumor detection using MRI images," *Journal of King Saud University - Computer and Information Sciences*, 34(6): 3259–3272, DOI: 10.1016/j.jksuci.2020.08.006.

[23] Sik-Ho Tsang. 2024. "Review: Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification) | by | Towards Data Science." Accessed: Jun. 16, 2024. [Online]. Available: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568