Jurnal CoreIT, Vol.11, No.2, December 2025
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

Comparison of Various Deep Learning Techniques to Obtain the Best Technique for Detecting Brain Cancer

Febi Yanto[1*], Elvia Budianita[2], Shir Li Wang[3]
[1,3]Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, Malaysia
[1,2]Dept. of Informatics Engineering, Universitas Islam Negeri Sultan Syarif Kasim, Indonesia
[3]Data Intelligence and Knowledge Management (DILIGENT), Universiti Pendidikan Sultan Idris, Malaysia
[3]Robotics and Cyber-Physical System (RACS), Universiti Pendidikan Sultan Idris, Malaysia

febiyanto@uin-suska.ac.id*, elvia.budianita@uin-suska.ac.id, shirli_wang@meta.upsi.edu.my

Abstract. This study aims to address the difficulty of comparing deep learning–based brain cancer detection methods due to differences in datasets and parameter settings, which limits the generalizability of previous findings. The purpose of this research is to evaluate the performance of several convolutional neural network (CNN) architectures using identical datasets and experimental configurations to determine the most effective technique for early brain cancer detection. The study builds a comparative framework using the Keras API on TensorFlow, supported by libraries such as NumPy, Pandas, Matplotlib, and Seaborn. All datasets were split into stratified training, validation, and test sets, and preprocessing included resizing images to 224×224 pixels, converting them to 3-channel RGB, normalizing the inputs, and applying data augmentation. CNN architectures, including VGG16, ResNet50, GoogleNet, and AlexNet, were trained with consistent parameter settings, including epoch count, batch size, learning rate optimization, and training protocols. Performance evaluation using accuracy, precision, recall, and F1-score shows that GoogleNet and ResNet50 achieve the highest results across datasets (average >94%), with GoogleNet slightly outperforming ResNet50. AlexNet performs poorly on the Kaggle dataset but shows potential on the private dataset, while VGG16 demonstrates moderate but less consistent performance. The originality of this study lies in providing a unified evaluation framework that enables fair comparison across CNN models, offering valuable insights for selecting optimal architectures for brain cancer detection.

Keywords: Benchmark, Brain Cancer, CNN, Detection, Deep Learning.

Received November 2025 / **Revised** December 2025 / **Accepted** December 2025

INTRODUCTION

The highest mortality rate in Asia is caused by brain cancer, according to a 2018 WHO report [1]. Symptoms of brain cancer can be identified based on criteria such as frequent headaches, mood swings, difficulty concentrating, seizures, and memory loss [1]. However, if patients are diagnosed too late, it can be fatal in the future [1]. Therefore, a technique is needed that can quickly and inexpensively diagnose patients who are grouped as having brain cancer.

Several conventional techniques for detecting brain cancer are still in use. These techniques are divided into invasive and non-invasive [1]. Invasive techniques involve making an incision in the brain and then taking a specific sample to identify the cancer. After that, a pathologist will analyze the sample. This invasive technique certainly takes a long time and a lot of effort.

In contrast, non-invasive techniques use body and brain scans using Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) equipment. Non-invasive methods are safer and faster than invasive techniques that involve direct physical contact. However, non-invasive techniques are highly dependent on the medical practitioner's analysis of MRI or CT results. Therefore, the medical practitioner's analysis of the MRI or CT results also affects the accuracy of the analysis.

Several researchers are developing accurate, non-invasive techniques for analyzing MRI or CT results using deep learning. Deep learning techniques have advantages, especially in the field of medicine, such as the ability to handle complex data, high precision, the best generalization, automation, and high efficiency and affordability in detecting brain cancer, as done by [2], [3], [4], [5], and [6]. They use CNN techniques to detect brain cancer. However, what distinguishes their research is the use of private and public datasets: [2]

DOI: 10.24014/coreit.v11i2.38599

and [5] use private datasets, while [3], [4], [5], and [6] use public datasets. Reference [6] detects brain cancer using machine learning techniques such as SVM, Random Forest, Decision Tree, Adaptive Boosting (AdaBoost), Gradient Boosting, and relies on CNN architectures such as VGG-16, VGG-19, ResNet50, InceptionRes-NetV2, InceptionV3, Xception, and DenseNet201 as feature extraction techniques. Meanwhile, the performance metrics used by researchers are almost the same: Recall, Precision, F1-score, and Accuracy. However, some studies cannot be compared because they use different datasets and parameters, so the results cannot be generalized as the best technique for detecting brain cancer. Therefore, the performance of various cancer detection techniques on the same dataset and under the same parameters needs to be evaluated.

Reference [2] uses deep-learning-based computer-aided detection (CAD) and achieves 87.1% sensitivity and an ROC value of 0.79 on MRI data from 121 patients with brain metastases, totaling 361 MRI results. [3] reported an accuracy of 99% using a public dataset of 3064 MRI results with three tumor types: meningiomas, gliomas, and pituitary tumors. The deep learning technique used by [2] was Faster region-based convolutional neural network (Faster R-CNN) with VGG16 architecture, while [3] used a convolutional neural network (CNN) with Residual Network (ResNet) architecture.

Reference [4] used a CNN with a VGG-16 architecture to predict Glioma tumors in the 2016 BRAT MRI dataset. This dataset consisted of MRI scan images, namely 188 non-Glioma and 192 Glioma. The results of the [4] study yielded 96.9% sensitivity, 99.3% specificity, and 99.2% accuracy. In contrast, researchers [5] tested CNNs with various architectures: VGG-16, ResNet-50, and Inception-v3. The CNN with the VGG-16 architecture provided the highest accuracy of 0.96. The brain tumor cell dataset used was publicly available on Kaggle and consisted of 233 data points.

Reference [6] did the same as [5], namely comparing various architectures: VGG-16, VGG-19, ResNet50, InceptionResNetV2, InceptionV3, Xception, and DenseNet201. However, the difference lies in the classification techniques used by [6], namely Support Vector Machine, Random Forest, Decision Tree, AdaBoost, and Gradient Boosting, whereas [5] is limited to CNN techniques. Of course, the scientific contribution of [6] is greater than that of [5]. Reference [6] identified the best-performing model, VGG-19-SVM, with an accuracy of 99.39%.

Significantly few researchers have compared various cancer detection techniques, such as [7] and [8]. Reference [7] presents multiple CNN architectures (AlexNet, VGG16, ResNet18, GoogleNet, and ResNet50) on a priv. In contrast, it compares several machine learning techniques (Support Vector Machine, K-Nearest Neighbors, Naïve Bayes, Decision Tree, and Linear Discrimination) and various CNN architectures (AlexNet, VGG16, ResNet18, GoogleNet, and ResNet50). Limitations in the use of datasets, performance evaluation, and parameters make it impossible to compare the two directly

Therefore, many studies cannot be directly compared to determine the best technique. The obstacles encountered include the inability to access publicly available datasets, the use of non-standard or limited research parameters, and the lack of model evaluation by researchers, such as cross-validation. Therefore, this study compares various public datasets using CNN architectures such as AlexNet [9], [10], V, includingsNet18, GoogleNet [11], and ResNet50. CNNs are most widely used by researchers to detect brain cancer across both public and private datasets. This study conducted experiments using CNN techniques on two public datasets, namely Brain MRI Images for Brain Tumor Detection and Figshare. This study used several parameters, including Epochs, Batch Size, Average Iterations, Learning Rate, and Training Protocol. Performance evaluation was carried out using Recall, Precision, F1-score, and Accuracy [12], [13].

METHODS

Collecting and labeling medical data is challenging because it involves data privacy and expert explanations [14]. Since 2012, CNNs have been widely used for image classification, achieving remarkable results [14]. CNNs are widely used by researchers in healthcare image processing, especially for MRI data. CNNs are believed to perform best for working with MRI data. Therefore, many researchers have improved CNNs, especially in various fields of image processing. The CNN architecture is divided into two parts, namely feature extraction and classification [15]. In general, the CNN architecture is divided into five layers, namely input, convolutional, pooling, fully connected, and classification [16], [17]. CNNs consist of several

layers that use distinct functions to convert input volumes into outputs. In essence, deep learning is an adaptation of artificial neural networks where the neurons are stacked on top of each other [18], [19].
This study evaluates the performance of CNN techniques with ed, including ResNet18, GoogleNet, and ResNet50. This study tests CNNs on various public datasets. Very few researchers have conducted studies as significant as this study has. Refer [7] only to CNNs with different architectures (multiple: multipleNet, VGG16, ResNet18, GoogleNet, and ResNet50) on private datasets. The results of the research from [7] cannot be generalized, let alone compared with other related studies. This is because it uses private datasets, so other researchers cannot compare it with the research in [7]. Therefore, the research in [7] can only be used for private environments.

This contrasts with [8], which uses the public REMBRANT dataset. Reference [8] compares machine learning and deep learning techniques. The machine learning techniques used are Support Vector Machine, K-Nearest Neighbors, Naïve Bayes, Decision Tree, and Linear Discriminant Analysis, while a CNN is used as a deep learning technique. The parameters used by [8] are epochs, batch size, average iterations, learning rate, and training protocol. In contrast, [7] uses the parameters: Gradient decay factor, epsilon, initial learning rate, L2 regularization, Gradient threshold, method, threshold (Gradient), maximum epochs, minimum batch size, and frequency (verbose). The similarity between studies [8] and [7] in the use of CNN architecture is AlexNet, VGG16, ResNet18, GoogleNet, and ResNet50. Although the techniques used by [7] and [8] are identical, the results obtained for each CNN architecture differ.

The performance evaluation used by [8] is accuracy (ACC), sensitivity (SE), specificity (SP), positive predictive value (PPV), negative predictive value (NPV), and area under the curve (AUC) of the receiver operator characteristic (ROC). In contrast, [7] uses Precision, Recall, and F-measure. Therefore, the results of studies [8] and [7] cannot be compared.

This study builds a framework to compare each model for detecting brain cancer, as shown in figure 1. This study uses the Keras API on TensorFlow. Several libraries are used, such as NumPy, Pandas, Matplotlib, Seaborn, and several TensorFlow and Keras modules. Each dataset is divided into training, validation, and test sets, ensuring hierarchical separation to maintain class distribution. In data processing, this study employs several techniques, such as resizing to 224x224 pixels, converting images to 3-channel RGB, and applying data augmentation during training. After that, the VGG16, ResNet50, GoogleNet, and AlexNet models were used with several parameters, including Epoch = 100, Batch Size = 35, Learning rate optimization = adam, and Training protocol = categorical_crossentropy. Epoch is the number of times the model checks all data during training, while Batch Size is the number of samples processed at once before updating the model's parameters. Categorical Cross-Entropy is a loss function for measuring errors in predicting in many-class classification, while the Adam Optimizer is an efficient, adaptive optimization algorithm that updates each parameter previously defined for the model.

After that, each model was trained and evaluated using accuracy, precision, recall, and F1-score. The results of each model were assessed for each dataset used, namely the tumor dataset with two classes, yes (155) and no tumor (98), the tumor dataset with three classes meningioma (708), glioma (1426), pituitary tumor (90)), and a tumor dataset wifourh four classes meningioma (306), glioma (300), pituitary tumor (300), and No cancer (405)

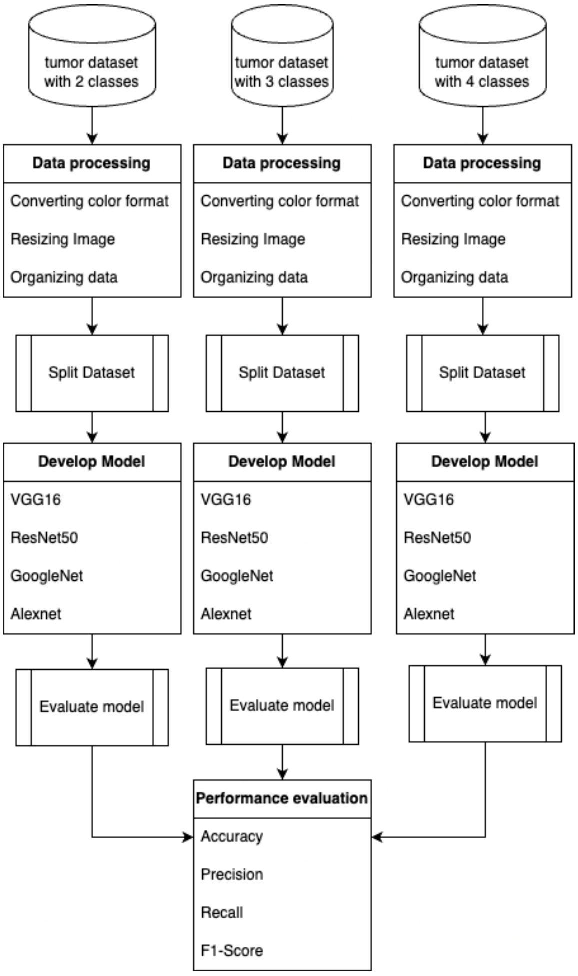

Figure 1. Framework for comparing the best Deep Learning techniques

RESULT AND DISCUSSION

The results of the best technique comparison measurements are explained in each section as follows:

A. VGG16

In figure 1, VGG16 achieved the highest accuracy of 0.9555 on the Kaggle1 dataset. Meanwhile, the accuracy value reported by [7] was 0. This is because [7] did not provide accurate results in their research. The high accuracy of VGG16 is reflected in the highest precision and recall values of 0.9572 and 0.9555, compared to the values reported by [7] of 0.55 and 0.5. The performance of VGG16 is fully supported by an F1-score of 0.952, which is higher than that reported by [7]. ThF1-score1-score in VGG16 indicates a good balance between precision and recall.

In general, as shown in figure 1, VGG16 achieved the highest performance across the dataset, especially on kaggle1. Meanwhile, on the Kaggle2 dataset, performance was poor, especially in terms of accuracy and precision. This is in contrast to [7], which obtain precision and recall values, but not purity values. Meanwhile, on the Kaggle2 dataset, VGG16 achieved low accuracy and precision, indicating that the model produced many false positives. This could be due to problematic data quality and balance. Therefore, future research could explore this aspect. In the Kaggle3 dataset, VGG16 achieved high recall but low precision, indicating that its predictions produced many false positives. This occurred due to overfitting; to address it, cross-validation or regularization techniques can be used.

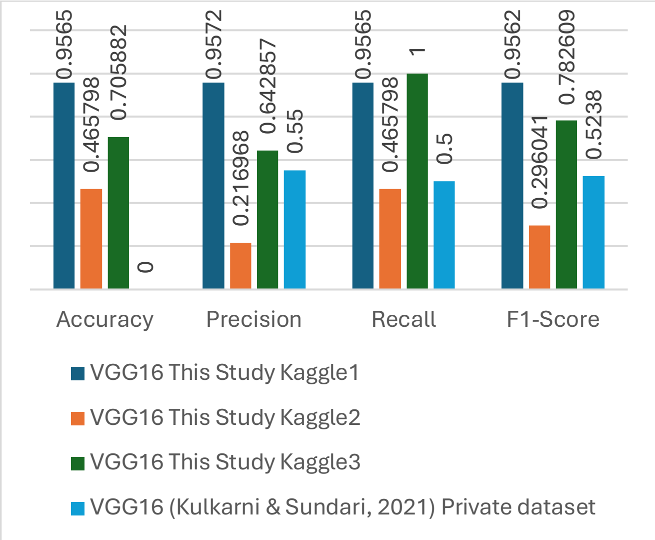

Figure 2. Performance evaluation of the VGG16 model on various datasets

B. ResNet50

In figure 2, the f1-score values for the kaggle1 and kaggle2 datasets are very high, namely 0.9855 and 0.954847, indicating a good balance between precision and recall. In contrast, the Kaggle3 dataset shows a low F1 score (0.862069), indicating lower precision and recall. Meanwhile, the low f1-score in the dataset [7] precision and recall are not optimal. Therefore, the performance of ResNet50 on kaggle1 and kaggle2 reached its maximum at the largest sizes. This shows that ResNet50 is powerful and reliable.

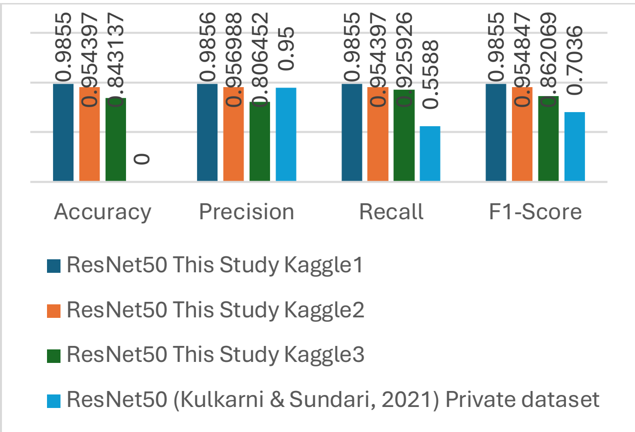

Figure 3. Performance evaluation of the ResNet50 model on various datasets

C. GoogleNet

In figure 3, GoogleNet on the kaggle1 and kaggle2 datasets achieved high accuracy, particularly on kaggle1, with a score of 0.9901. Similarly, GoogleNet achieved the highest precision scores, particularly on kaggle1 with a score of 0.9901. Perfect recall was completed on the Kaggle3 dataset [7]. However, the recall values for the kaggle1 and kaggle2 datasets remain high at 0.9901 and 0.977199, respectively. Meanwhile, the F1-score values for kaggle1 and kaggle2 are high, especially for the kaggle1 dataset at 0.9901. Therefore, GoogleNet performs best across all measures, indicating that the model's validation and training processes are highly reliable.

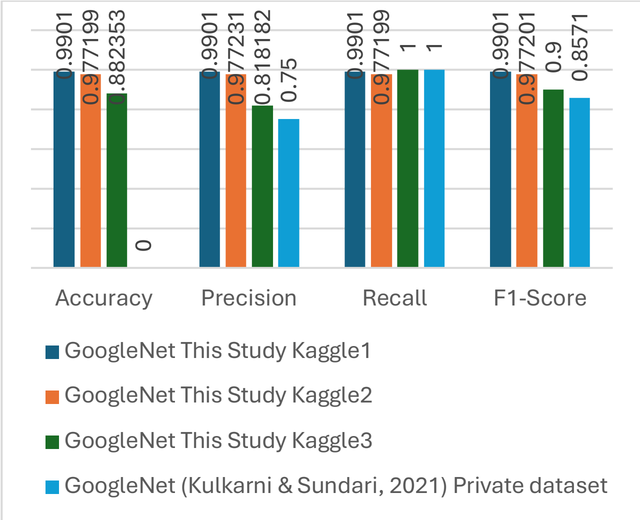

Figure 4. Evaluation of the GoogleNet model performance on various datasets

D. AlexNet

In figure 4, AlexNet achieved the highest accuracy on the kaggle3 dataset, namely 0.803922. However, the highest precision was found in the dataset [7] (0.937), followed by the kaggle3 dataset (0.72973). Meanwhile, the highest recall was obtained by kaggle3 and the private dataset [7].Similarly, the highest F1-score was received by the dataset [7] (0.96774), followed by the Kaggle3 dataset (0.84375). Therefore, AlexNet achieved the best performance on the Kaggle3 dataset, except for precision.

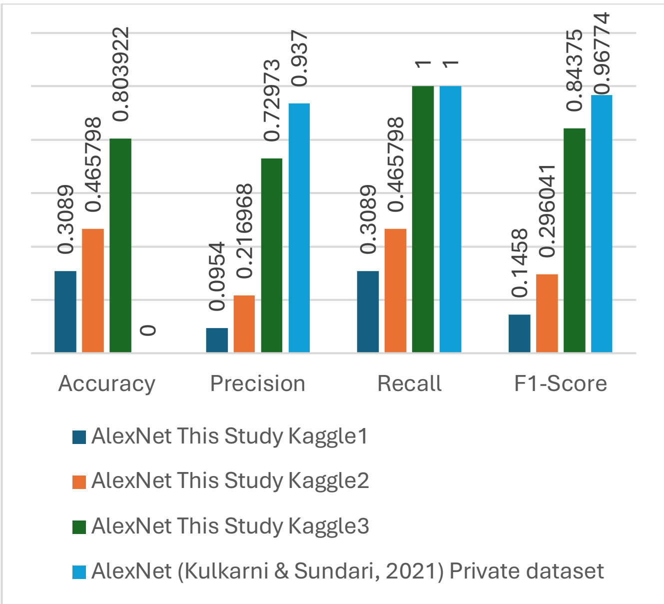

Figure 5. Evaluation of the AlexNet model performance on various datasets

Based on table 1, many models were evaluated on various datasets. Performance evaluation used accuracy, precision, recall, and F1-score. GoogleNet performed best on all datasets, especially on kaggle1 and kaggle2. In addition, ResNet50 also generally performed well on all datasets. On the Kaggle dataset, GoogleNet achieved the highest accuracy of 0.9901, while AlexNet achieved the lowest accuracy of 0.3089 and 0.1458. On the Kaggle2 dataset, GoogleNet achieved the best performance for all sizes, exceeding 0.977. On the Kaggle3 dataset, GoogleNet and ResNet50 achieved accuracy and F1-scores above 0.8, while AlexNet achieved high precision but low accuracy and moderate F1-scores. Unlike the kaggle1, kaggle2, and kaggle3 datasets, AlexNet achieved maximum performance across all sizes, exceeding 0.93, while

GoogleNet achieved balanced performance and a moderate f1-score. Therefore, the model proposed in this study is more reliable across all datasets than that in [7].

Table 1. Performance comparison with current research

Model	Researcher	Dataset	Accuracy	Precision	Recall	F1-Score
GoogleNet	This study	Kaggle1	0.9901	0.9901	0.9901	0.9901
ResNet50	This study	Kaggle1	0.9855	0.9856	0.9855	0.9855
VGG16	This study	Kaggle1	0.9565	0.9572	0.9565	0.9562
AlexNet	This study	Kaggle1	0.3089	0.0954	0.3089	0.1458
GoogleNet	This study	Kaggle2	0.977199	0.977231	0.977199	0.977201
ResNet50	This study	Kaggle2	0.954397	0.956988	0.954397	0.954847
VGG16	This study	Kaggle2	0.465798	0.216968	0.465798	0.296041
AlexNet	This study	Kaggle2	0.465798	0.216968	0.465798	0.296041
GoogleNet	This study	Kaggle3	0.882353	0.818182	1	0.9
ResNet50	This study	Kaggle3	0.843137	0.806452	0.925926	0.862069
VGG16	This study	Kaggle3	0.705882	0.642857	1	0.782609
AlexNet	This study	Kaggle3	0.803922	0.729730	1	0.843750
AlexNet	[7]	Private dataset	-	0.937	1	0.96774
VGG16	[7]	Private dataset	-	0.55	0.5	0.5238
ResNet50	[7]	Private dataset	-	0.95	0.5588	0.7036
GoogleNet	[7]	Private dataset	-	0.75	1	0.8571

Based on table 2, several models performed poorly, such as the AlexNet model on the kaggle1 dataset (accuracy 0.3089, precision 0.0954, recall 0.3089, and F1-Score 0.1458), VGG16 on kaggle2 (accuracy 0.465798, precision 0.216968, recall 0.465798, and F1-Score 0.296041) and AlexNet on kaggle2 (accuracy 0.465798, precision 0.216968, recall 0.465798, and F1-Score 0.296041). Therefore, this study attempts to re-evaluate using the overfitting technique, namely, cross-validation.

Table 2. AlexNet and GoogLeNet models Re-evaluated using Cross-Validation

Model	Dataset	Accuracy		Precision		Recall		F1-Score	
		Before	After	Before	After	Before	After	Before	After
GoogleNet	Kaggle1	0.9901	0.9893	0.9901	0.9894	0.9901	0.9893	0.9901	0.9893
AlexNet	Kaggle1	0.3089	0.3089	0.0954	0.0954	0.3089	0.3089	0.1458	0.1458
GoogleNet	Kaggle2	0.977199	0.964549	0.977231	0.974194	0.977199	0.968094	0.977201	0.970905
ResNet50	Kaggle2	0.954397	0.873098	0.956988	0.967742	0.954397	0.867980	0.954847	0.909836
VGG16	Kaggle2	0.465798	0.763137	0.216968	0.883871	0.465798	0.770337	0.296041	0.819126
AlexNet	Kaggle2	0.465798	0.842353	0.216968	0.864516	0.465798	0.874212	0.296041	0.867903
GoogleNet	Kaggle3	0.882353	0.960187	0.818182	0.9610855	1	0.960187	0.9	0.960329
ResNet50	Kaggle3	0.843137	0.975851	0.806452	0.9759924	0.925926	0.975851	0.862069	0.975852
VGG16	Kaggle3	0.705882	0.518914	0.642857	0.6060496	1	0.518914	0.782609	0.386107
AlexNet	Kaggle3	0.803922	0.922965	0.729730	0.923267	1	0.922965	0.843750	0.922734

In table 2, the GoogleNet model showed insignificant changes with cross-validation, and the AlexNet model showed no changes at all with cross-validation on the Kaggle dataset. In contrast, on the Kaggle 2 and Kaggle 3 datasets, the GoogleNet model still did not experience a significant decline, while AlexNet experienced a very substantial increase in performance in terms of accuracy, precision, recall, and F1-Score on the Kaggle 2 dataset (accuracy 0.842, precision 0.864, recall 0.874, and F1 -Score 0.867) and Kaggle3 (accuracy 0.922, precision 0.923, recall 0.922, and F1-Score 0.922). Therefore, cross-validation can be used to prevent overfitting and achieve better results. In addition, AlexNet is the oldest deep learning technique, so its limitations cannot keep up with developments in research. The limitations of AlexNet include depth, architectural complexity, and model generalization capabilities. Compared with GoogleNet and ResNet50, AlexNet is less effective at predicting brain cancer or similar diseases, though many researchers still use the AlexNet architecture today because it can be optimized.

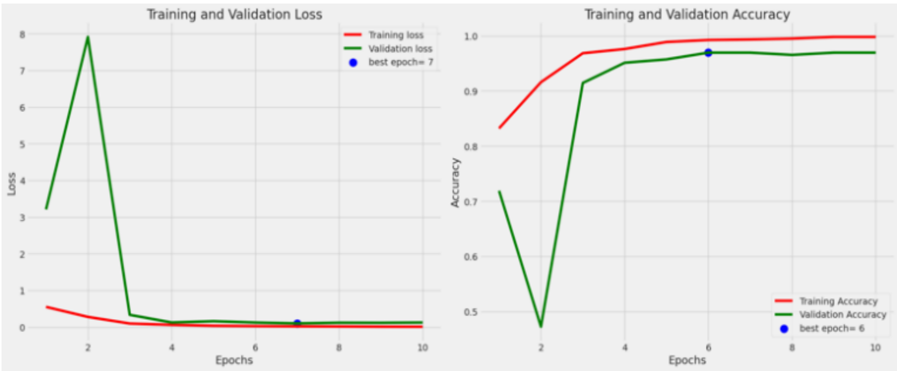

Figure 3. Training and validation on the ResNet50 model

Overall, GoogleNet and ResNet50 excelled in all performance evaluations across various datasets. This is also evident in figure 6, where the ResNet50 model shows progress in learning on both the training and validation sets, with minimal overfitting. Similarly, the GoogleNet model in figure 7, trained on epochs 9 and 10, produced a robust model during the training and validation phases.
Therefore, GoogleNet and ResNet50 have different capabilities: GoogleNet excels in computational efficiency and multi-scale feature capture, while ResNet50 excels in complex data training processes involving deep networks by optimizing residual connections. Additionally, both models can handle overfitting by leveraging features such as dropout, batch normalization, and residual connections, enabling generalization.

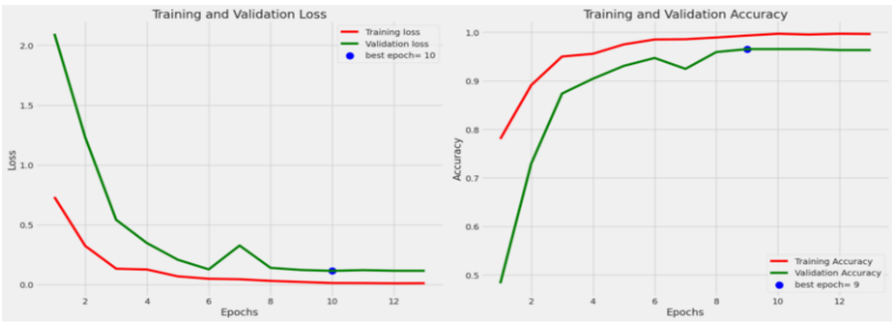

Figure 4. Training and validation on the GoogleNet model

CONCLUSION

Performance evaluation shows that GoogleNet and ResNet50 are superior models across various datasets, with GoogleNet slightly outperforming ResNet50. AlexNet struggles with the Kaggle dataset but shows potential on private datasets. VGG16 shows moderate performance but is less consistent than GoogleNet or ResNet50. Therefore, the best model choice depends on the specific dataset and application requirements. The model's complexity, which requires significant computational resources, and the imbalance in the number of data classes are limitations of this study. The practical implications of GoogleNet or ResNet50 in a clinical context include automating brain cancer detection processes, cancer segmentation, predicting brain cancer severity, monitoring brain cancer progression, and personalized drug administration.

REFERENCES

[1] G. S. Tandel et al., "A Review on a Deep Learning Perspective in Brain Cancer Classification," Cancers, vol. 11, no. 1, p. 111.

[2] M. Zhang et al., "Deep-Learning Detection of Cancer Metastases to the Brain on MRI," J. Magn. Reson. Imaging, vol. 52, no. 4, pp. 1227–1236, Oct. 2020.

[3] S. A. Abdelaziz Ismael, A. Mohammed, and H. Hefny, "An enhanced deep learning approach for brain cancer MRI images classification using residual networks," Artif. Intell. Med., vol. 102, p. 101779, Jan. 2020.

[4] M. Tamilarasi, "Performance Analysis of Glioma Brain Tumor Segmentation Using CNN Deep Learning Approach," IETE J. Res., pp. 1–12, Mar. 2021.

[5] C. Srinivas et al., "Deep Transfer Learning Approaches in Performance Analysis of Brain Tumor Classification Using MRI Images," J. Healthc. Eng., vol. 2022, pp. 1–17, Mar. 2022.

[6] S. Ahmad and P. K. Choudhury, "On the Performance of Deep Transfer Learning Networks for Brain Tumor Detection Using MR Images," IEEE Access, vol. 10, pp. 59099–59114, 2022.

[7] S. M. Kulkarni and G. Sundari, "Comparative Analysis Of Performance Of Deep Cnn Based Framework For Brain Mri Classification Using Transfer Learning," J. Eng. Sci. Technol., vol. 16, no. 4, pp. 2901–2917, 2021.

[8] G. S. Tandel, A. Tiwari, and O. G. Kakde, "Performance optimisation of deep learning models using majority voting algorithm for brain tumour classification," Comput. Biol. Med., vol. 135, p. 104564, Aug. 2021.

[9] S. Anjum et al., "Detecting brain tumors using deep learning convolutional neural network with transfer learning approach," Int. J. Imaging Syst. Technol., vol. 32, no. 1, pp. 307–323, Jan. 2022.

[10] A. DIKER, "A Performance Comparison of Pre-trained Deep Learning Models to Classify Brain Tumor," in IEEE EUROCON 2021 - 19th International Conference on Smart Technologies, IEEE, Jul. 2021.

[11] T. Noguchi et al., "A Fundamental Study Assessing the Diagnostic Performance of Deep Learning for a Brain Metastasis Detection Task," Magn. Reson. Med. Sci., vol. 19, no. 3, pp. 184–194, 2020.

[12] M. Nazir, S. Shakil, and K. Khurshid, "Role of deep learning in brain tumor detection and classification (2015 to 2020): A review," Comput. Med. Imaging Graph., vol. 91, p. 101940, Jul. 2021.

[13] Y. Xie et al., "Convolutional Neural Network Techniques for Brain Tumor Classification (from 2015 to 2022): Review, Challenges, and Future Perspectives," Diagnostics, vol. 12, no. 8, p. 1850, Jul. 2022.

[14] P. Immaculate Rexi Jenifer and S. Kannan, "Deep Learning with Optimal Hierarchical Spiking Neural Network for Medical Image Classification," Comput. Syst. Sci. Eng., vol. 44, no. 2, pp. 1081–1097, 2023.

[15] Y. E. Almalki et al., "Isolated Convolutional-Neural-Network-Based Deep-Feature Extraction for Brain Tumor Classification Using Shallow Classifier," Diagnostics, vol. 12, no. 8, p. 1793, Jul. 2022.

[16] M. Ahmadi, A. Sharifi, M. Jafarian Fard, and N. Soleimani, "Detection of brain lesion location in MRI images using convolutional neural network and robust PCA," Int. J. Neurosci., vol. 133, no. 1, pp. 55–66, Jan. 2023.

[17] M. F. Alanazi et al., "Brain Tumor/Mass Classification Framework Using Magnetic-Resonance-Imaging-Based Isolated and Developed Transfer Deep-Learning Model," Sensors, vol. 22, no. 1, p. 372, Jan. 2022.

[18] O. Özkaraca et al., "Multiple Brain Tumor Classification with Dense CNN Architecture Using Brain MRI Images," Life, vol. 13, no. 2, p. 349, Jan. 2023.

[19] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, "A review on deep learning in medical image analysis," Int. J. Multimed. Inf. Retr., vol. 11, no. 1, pp. 19–38, Mar. 2022.