An Image Processing-based and Deep Learning Model to Classify Brain Cancer

Amal Al-Shahrani
College of Computing, Umm Al-Qura University, Saudi Arabia
amshahrani@uqu.edu.sa (corresponding author)

Wusaylah Al-Amoudi
College of Computing, Umm Al-Qura University, Saudi Arabia
s442002795@st.uqu.edu.sa

Raghad Bazaraah
College of Computing, Umm Al-Qura University, Saudi Arabia
s442002867@st.uqu.edu.sa

Atheer Al-Sharief
College of Computing, Umm Al-Qura University, Saudi Arabia
s442003205@st.uqu.edu.sa

Hanan Farouquee
College of Computing, Umm Al-Qura University, Saudi Arabia
s442017249@st.uqu.edu.sa

Received: 12 May 2024 | Revised: 26 May 2024 and 30 May 2024 | Accepted: 31 May 2024

ABSTRACT

In recent years, the prevalence of cancer has increased significantly around the world. Cancer is considered one of the most dangerous diseases in humans. Cancer screening devices, such as Magnetic Resonance Imaging (MRI), X-ray imaging, ultrasound imaging, and others, play an important role in its early detection. This study aims to facilitate cancer tumor detection on mobile phones with high accuracy in a short period of time using deep learning techniques. A brain tumor dataset was used, consisting of 4,489 images and 14 classified types, and experiments were carried out using ResNet 12, DenseNet, YOLOv8, and MobileNet to evaluate them in terms of accuracy, speed, and model size. ResNet12, DenseNet, YOLOv8, and MobileNet results indicated satisfactory accuracy ranging from 88% to 97.3%. YOLOv8 was the most suitable model, as its fastest inference time of 1.8 ms, preprocessing time of 0.1 ms, highest accuracy of 97.3%, and compact model size make it ideal for real-time mobile applications.

Keywords-brain cancer; deep learning, cancer detection; You-Only-Look-Once (YOLO); MRI; image processing; diagnosis

I. INTRODUCTION

The anatomy of the brain is very complex, with different parts responsible for different nervous system functions. Brain tumors can develop in any part of the brain or skull. There are more than 120 different tumor types that can develop in the brain [1]. Doctors face problems in reading MRI images due to inability to determine fine details, insufficient practical training, and other problems that may cause incorrect diagnoses. Advances in Artificial Intelligence (AI) have made a large contribution to data processing [2] and have transformed healthcare by considering medical treatments and diagnostics as data problems. Deep learning (DL) is a subdivision of Machine Learning (ML) that extracts detailed features from data and has proven to be very effective in many data-processing tasks [3]. DL plays an important role in healthcare. Before the advent of deep learning, medical errors and misdiagnoses were widespread compared to today. Convolutional Neural Networks (CNNs) analyze images and extract features to classify images. Classification is crucial in areas such as medical imaging [4]. Transfer learning (TL) is an ML technique where knowledge learned from a task is reused to increase performance on a related task. Pre-trained methods are built on optimal procedures, depending on the chosen method and data, to ensure greater accuracy in achieving the expected results [5].

II. DATA COLLECTION AND RESEARCH METHOD

A. Dataset Collection

High-quality datasets are important to achieve high accuracy when training models [6, 7]. Since MRI scans are among the most crucial ways to identify brain cancer, an MRI dataset was selected, comprising T1 and T2 contrast-enhanced MRI images, divided by brain tumor type [8]. This dataset was chosen for its diversity and excellent image quality. It contains 14 different types of brain cancer, divided by astrocytoma, carcinoma, and ependymoma, including ganglion glioma, germinoma, glioblastoma, granuloma, medulloblastoma, meningioma, neuroma, oligodendroglioma, papilloma, schwannoma, and tuberculoma. The total number of images was 4479, divided into 44 categories. In contrast to more common types, there was a discrepancy in the number of images of some rare types of brain cancer due to the small number of people affected by them, such as ependymoma, ganglioglioma, and germinal tumors [9].

B. Image Preprocessing

Image processing is a crucial step in preparing data for deep learning, as it increases the generalization capacity and accuracy of models [10]. Additionally, it increases memory efficiency, which expedites the model training process. Image data processing was performed in the Google Colab environment using the Keras ImageDataGenerator module. The pre-processing steps included:

- Data gathering: A designated directory was used to gather file paths and the labels that accompany them.
- Image loading: The Keras ImageDataGenerator flow_from_dataframe function was used to load images from file paths.
- Resize photos: The images were shrunk to fit the typical deep learning model scale of 224×224 pixels.
- Image grouping: Since the typical size for training DL models is 32 images, the images were sorted into groups of 32 images.
- Data division: Data were split into 80% for training, 10% for validation, and 10% for testing.
- Data visualization: A subset of the images is shown to ensure that the loading and processing of the data were performed correctly. Figure 1 shows the appearance of a group of pre-processed and classified MRI images.

C. ResNet12

The ResNet architecture can resolve network depth issues in CNNs [11]. It is a significant advancement in ML methods for high-dimensional data analysis, including image processing. Research in the domain of image recognition has indicated that the proper operation of a CNN requires the use of multiple layers. Some approaches utilize more layers than usual to get better results. However, this quickly saturates the layers and decreases the accuracy. ResNet is a residual learning framework to address the degradation issue [12]. This approach uses identity shortcut connections to resolve degradation, allowing the layers to pick up the remaining functions. This means that the network learns complicated patterns in record time without requiring appreciable improvements in resources [13]. ResNet models generally address a wide range of deep learning problems, including handling more complicated data formats and training deep networks.

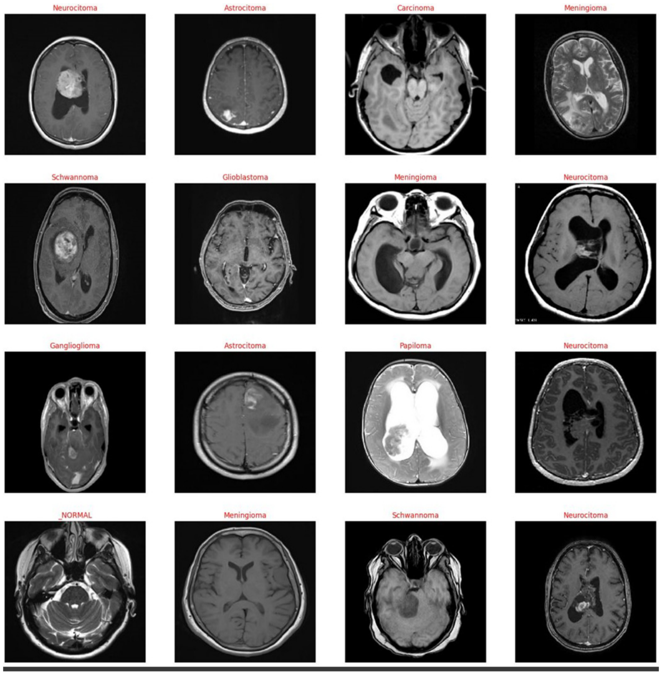

Fig. 1. MRI images of brain cancer.

D. MobileNet

MobileNet is a CNN architecture that was created with portable and embedded devices in mind [14, 15]. One feature that sets MobileNet apart is its use of deep separable convolutions. Compared to other CNNs, this technique breaks the typical convolution into two distinct steps: pointwise convolution and deep convolution. This reduces computational cost [15] and requires less computational energy, making the model less weighty [14, 15]. MobileNet involves two other hyperparameters, the width multiplier and the precision multiplier, which weigh the model size and accuracy, respectively. This makes MobileNet ideal for real-time applications on mobile devices with limited processing resources.

E. DenseNet169

DenseNet is a CNN architecture designed to improve image recognition. It addresses the limitations of traditional CNNs, such as vanishing gradients and redundant layers [16]. The basic points that distinguish DenseNet are the following:

- Dense connections: Unlike traditional CNNs, DenseNet connects each layer to all subsequent layers. This improves information flow and feature reuse [16].

- Fewer parameters: DenseNet achieves high performance with fewer parameters compared to other CNNs, such as ResNet [14, 18]. This makes it more efficient for tasks that require less computational power [17].
- Variants: DenseNet comes in different configurations, such as DenseNet-169, which has 169 efficient communication layers for complex data processing [13].

Its efficient architecture makes it a valuable tool for various computer vision tasks, especially those requiring real-time performance on resource-constrained hardware [17].

F. YOLOv8

The YOLO architecture provides remarkably accurate real-time picture classification, making a substantial contribution to the evolution of computer vision [18, 19]. YOLO has improved significantly throughout its iterations, reaching a new peak for image classification tasks with YOLOv8. Ultralytics' YOLOv8 brings improvements to its architecture and offers a graphical user interface via a pip package, making it more developer-friendly. Two major improvements in YOLOv8 are the incorporation of new loss functions, which are especially helpful for precisely categorizing complex images, and the use of free anchor boxes for faster processing. Furthermore, YOLOv8 is a valuable option for image classification tasks because of its remarkable accuracy and developer-friendly features, including an easy-to-use CLI and a neatly organized Python package. YOLO's position as a leading computer vision solution is further cemented by the vibrant community that surrounds it and its iterations, guaranteeing ongoing support and improvement [20].

G. Training Methodology

The dataset was divided into 80% for training and 10% for validation and testing, respectively. Several hyperparameters were used for each model to determine which was best for validation and testing. Most models performed best at 50 epochs, 64 batch size, and 0.001 learning rate. Table I shows the hyperparameters used and the accuracy achieved.

TABLE I. HYPERPARAMETERS AND ACCURACY

Model	Epochs	Batch size	Learning Rate	Accuracy
YOLOv8	30	32	0.0001	94.6%
	40	64	0.0001	96.4%
	50	64	0.0001	97.3%
ResNet12	30	32	0.0001	90.62%
	40	64	0.0001	88.94%
	50	64	0.0001	92.36%
Densenet169	30	128	0.0001	96.86%
	40	64	0.0001	96.98%
	50	64	0.0001	97%
MobileNet	30	128	0.0001	87.16%
	40	64	0.0001	88.7%
	50	64	0.0001	85.17%

H. Training Environment

Model training requires powerful computing resources. These models were trained using Google Colab, a cloud platform for running Python programs. One of the free resources provided by Google Colab is a 12GB VRAM T4 GPU [21]. However, as this free version was slow in training, the commercial version was used, as it provided more powerful GPU options, such as GPU V100 and GPU A100, to complete training substantially faster.

I. Evaluation Metrics

The detection performance of the models was evaluated using precision (P) and recall (R), as they can determine the models' accuracy and dependability in identifying different brain cancer types. A confusion matrix with four components is used to calculate these metrics:

- True Positives (TP): The number of positive-class cases that the model accurately classifies as such.
- False Positives (FP): The number of cases that fall into the negative category but are mistakenly recorded by the model as positive.
- True Negatives (TN): The number of cases that the model accurately classifies as negatives.
- False Negatives (FN): The number of cases that fall inside the positive category but the model mistakenly categorized as negative.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

III. RESULTS AND DISCUSSION

The most important things to focus on when training a model are speed, accuracy, and model size. However, this study placed more emphasis on accuracy because it deals with sensitive data that can affect diagnostic results positively or negatively. A series of experiments were conducted to achieve the best-desired result.

A. Confusion Matrix

The confusion matrix is an essential tool for evaluating the performance of classification models. Figures 2-5 show the confusion matrices of the different models. The more samples on the diagonal, the more accurate the model is at correctly classifying the class. Cells outside the diagonal indicate samples that were misclassified into another category.

B. Mean Average Precision and Model Size

Table II shows the measurements and the corresponding sizes for each model. In terms of accuracy, as shown in Figure 6, YOLOv8 achieved the highest accuracy of 97.3%, indicating its ability to correctly identify and classify objects in the given dataset. DenseNet169 followed closely with an accuracy of 97%, demonstrating its strong performance as well. MobileNet achieved an accuracy of 85.17%, which is comparatively lower than the other models. ResNet12 obtained an accuracy of 92.36%, positioning it between the highest and lowest accuracy models. When comparing model sizes, YOLOv8 had the highest model size, and MobileNet had the modest model size.

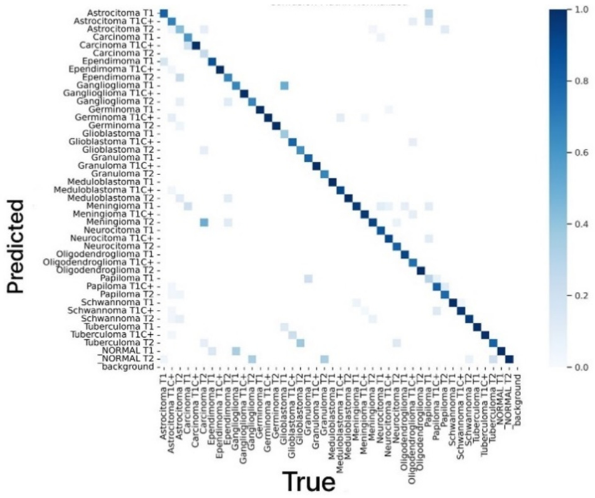

Fig. 2. Confusion matrix for YOLOv8.

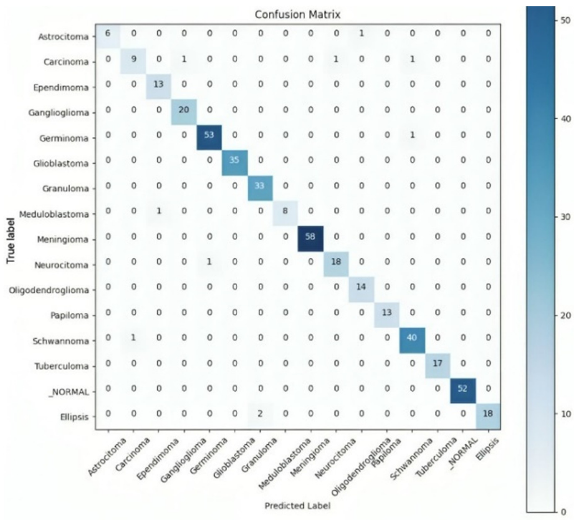

Fig. 3. Confusion matrix for ResNet12.

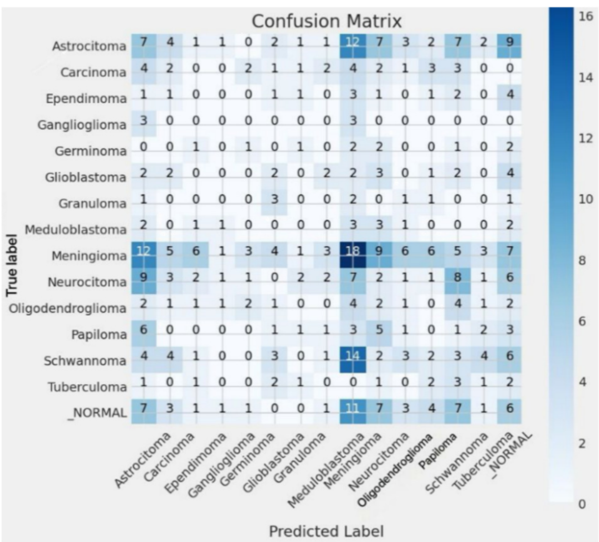

Fig. 4. Confusion matrix for DenseNet169.

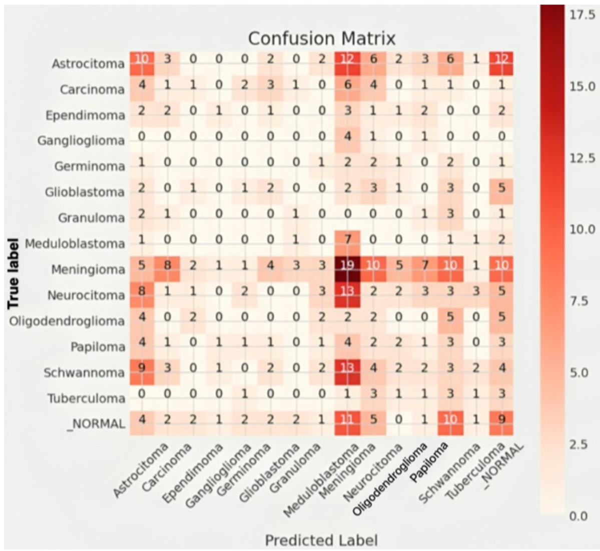

Fig. 5. Confusion matrix for MobileNet.

TABLE II. MODEL EVALUATION METRICS COMPARISON

Model	Model size (MB)	Precision (%)	Recall (%)	Accuracy (%)
YOLOv8	57.39	97.45	96.11	97.3
ResNet12	38.15	90.3	90	92.36
Densenet169	49.89	98.05	95.67	97
MobileNet	18.35	82.92	80	85.17

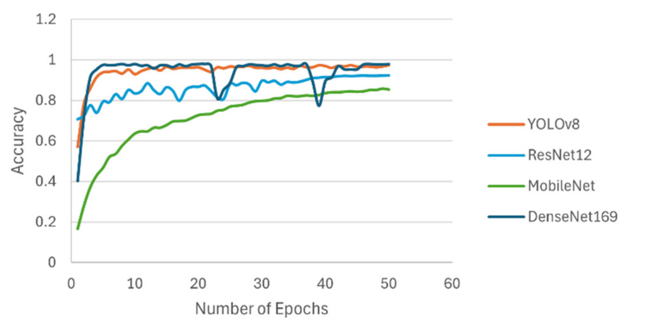

Fig. 6. Models' accuracy during validation.

C. Speed

The average time required to complete each model was determined after numerous trials. YOLOv8 demonstrated the fastest speed, with an average training and testing time of 14 minutes and 49 seconds. Following YOLOv8, ResNet12 had an average time of 36 minutes and 54 seconds, MobileNet took approximately 40 minutes, and DenseNet169 was the slowest, with an average time of 40 minutes and 31 seconds. YOLOv8 outperformed the other models by a significant margin, being approximately 22 minutes and 5 seconds faster than the next best-performing model.

D. Discussion

This study used the dataset in [8], focusing on classifying brain tumors into 14 types, and evaluated the ResNet12, DenseNet, YOLOv8, and MobileNet algorithms. YOLOv8 emerged as the best performer, achieving an accuracy of 97.3%

and having the fastest inference time of 1.8 ms, making it suitable despite its larger model size. The importance of accuracy is due to the sensitivity of the diagnostic data. Future improvements aim to address the model size constraint. Additionally, future improvements will expand the dataset to include other types of cancer to enhance the model's diagnostic capabilities beyond brain tumors to include breast, lung, and liver cancers, and determine malignancy and tumor location.

The size and speed of deep learning models are affected by various factors, including architecture and design choices. In the case of YOLOv8 [22], its larger size of 57.39 MB can be attributed to the complexity of its structure, which includes many layers and parameters. YOLOv8's accuracy and speed are commendable, with an accuracy of 97.3% achieved in 13 minutes and 57 seconds. However, the larger size may be a drawback for mobile applications with limited storage capacity. On the other hand, DenseNet169 offers a smaller model size of 49.98 MB. This size reduction can be attributed to the efficient parameter sharing and feature reuse mechanisms in its architecture. DenseNet169 achieved a high accuracy of 97% in 40 minutes and 22 seconds. DenseNet169 in this study was better than in [23] by 1%. The performance of the model is reasonable, but the longer training time may make it less suitable for applications that require faster inference. MobileNet, despite its low accuracy of 85.1%, stands out due to its small model size of 18.35 MB and relatively faster training time of 37 minutes and 38 seconds. MobileNet achieves this smaller size and higher speed by taking advantage of detachable convolutions and other optimizations designed specifically for mobile and embedded devices. These properties make MobileNet a suitable choice for mobile applications where limited storage capacity and computational resources are critical. ResNet12 achieves a balance between accuracy, model size, and training time, with 92.36%, 38.15 MB, and 33 minutes and 29 seconds, respectively, offering a reasonable performance trade-off. The model size is smaller than that of YOLOv8 and DenseNet169, which makes it relatively more suitable for mobile applications with storage limitations. Furthermore, it was faster than DenseNet169 in training, which can be useful in scenarios that require faster iterations or updates of the model.

In short, each of the four models offers different trade-offs in terms of size, speed, and accuracy. YOLOv8 and DenseNet169 provide higher resolution but at larger sizes, making them less suitable for mobile applications with limited storage capacity. MobileNet, with its smaller size and faster training time, is well-suited for resource-constrained environments but sacrifices some accuracy. ResNet12 strikes a balance between size, speed, and accuracy, making it a reasonable choice for mobile applications where a trade-off between these factors is desirable. However, this study aimed to choose the model with the highest accuracy, as it deals with sensitive diagnostic data. In addition, the diversity of the dataset contributed to improving the results, as it contains images of 14 different types of brain cancer, common and uncommon, while previous studies used a dataset with 3 types of brain cancers [24-26]. This diversity improves the chances of discovering more brain cancer types using DL.

Human specialists evaluated the YOLOv8 application and verified its effectiveness, usefulness, and efficiency in identifying brain cancer. Scientists are excited about the potential of YOLOv8 because it has many advantages over traditional techniques. By using an AI model, the accuracy of brain cancer detection can be significantly increased, leading to earlier diagnosis and better treatment outcomes for patients. Furthermore, YOLOv8 can interpret brain MRI images more quickly than traditional techniques and hospital routines, leading to faster diagnosis and shorter waiting times. Possible errors arising from the model and inaccurate image recognition can be addressed by increasing the datasets and further classifying them. This can contribute to system efficiency, improving diagnoses and the safety of patients. Therefore, this can increase community confidence in brain cancer detection algorithms.

IV. CONCLUSION AND FUTURE WORK

Taking advantage of classification and DL techniques, this study presented a method to detect brain cancer and its type. To achieve this, a dataset consisting of 4489 images of 14 types of brain cancer was used, each classified as astrocytoma, carcinoma, ependymoma, ganglioglioma, germinoma, glioblastoma, granuloma, medulloblastoma, meningioma, neurocytoma, oligodendroglioma, papilloma, schwannoma, or tuberculoma. Subsequently, experiments were carried out, evaluating the ResNet12, DenseNet, YOLOv8, and MobileNet models. Among the four models, YOLOv8 emerged as the best performer, achieving its fastest inference time of 1.8 ms, a pre-processing time of 0.1 ms, and an accuracy of 97.3%. Therefore, this model was selected as the most suitable option, making it ideal for mobile applications. Although the size of the model was larger than the others, more emphasis was placed on accuracy, due to the sensitive data that can affect the diagnostic results. Future improvements would address this constraint. Future research directions involve expanding training datasets to include other types of cancer, such as breast cancer, lung cancer, liver cancer, and others, and determining whether the tumor is malignant or benign and its location. Ultimately, this study emphasizes the importance of advancing technology in healthcare, as it can lead to an easier and more flexible life in terms of providing accurate and faster medical diagnoses in mobile phone applications.

REFERENCES

[1] "Brain Tumors and Brain Cancer." https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor.

[2] P. K. Mall and P. K. Singh, "BoostNet: a method to enhance the performance of deep learning model on musculoskeletal radiographs X-ray images," *International Journal of System Assurance Engineering and Management*, vol. 13, no. 1, pp. 658–672, Jan. 2022, https://doi.org/10.1007/s13198-021-01580-3.

[3] K. C. Kamal, Z. Yin, M. Wu, and Z. Wu, "Evaluation of deep learning-based approaches for COVID-19 classification based on chest X-ray images," *Signal, Image and Video Processing*, vol. 15, no. 5, pp. 959–966, Jan. 2021, https://doi.org/10.1007/s11760-020-01820-2.

[4] "Types of Deep Learning & Their Uses in Healthcare." https://healthitanalytics.com/features/types-of-deep-learning-their-uses-in-healthcare.

[5] "Transfer learning." https://en.wikipedia.org/w/index.php?title=Transfer_learning.

[6] J. Ding and X. Li, "An Approach for Validating Quality of Datasets for Machine Learning," in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 2795–2803, https://doi.org/10.1109/BigData.2018.8622640.

[7] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do, "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?" arXiv, Jan. 2016, https://doi.org/10.48550/arXiv.1511.06348.

[8] "Brain Tumor MRI Images 44 Classes." https://www.kaggle.com/datasets/fernando2rad/brain-tumor-mri-images-44c.

[9] M. Havaei *et al.*, "Brain tumor segmentation with Deep Neural Networks," *Medical Image Analysis*, vol. 35, pp. 18–31, Jan. 2017, https://doi.org/10.1016/j.media.2016.05.004.

[10] "What is the purpose of image preprocessing in deep learning?" https://www.isahit.com/blog/what-is-the-purpose-of-image preprocessing-in-deep-learning.

[11] H. Singh, N. Helian, R. Adams, and Y. Sun, "Sentiment Analysis using BLSTM-ResNet on Textual Images," in *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy, Jul. 2022, pp. 1–8, https://doi.org/10.1109/IJCNN55064.2022.9892883.

[12] H. Choi, S. Ryu, and H. Kim, "Short-Term Load Forecasting based on ResNet and LSTM," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Aalborg, Denmark, Oct. 2018, pp. 1–6, https://doi.org/10.1109/SmartGridComm.2018.8587554.

[13] S. Anwar, S. R. Soomro, S. K. Baloch, A. A. Patoli, and A. R. Kolachi, "Performance Analysis of Deep Transfer Learning Models for the Automated Detection of Cotton Plant Diseases," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11561–11567, Oct. 2023, https://doi.org/10.48084/etasr.6187.

[14] E. Elfatimi, R. Eryigit, and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models," *IEEE Access*, vol. 10, pp. 9471–9482, 2022, https://doi.org/10.1109/ACCESS.2022.3142817.

[15] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York City, NY, USA, Oct. 2019, pp. 0280–0285, https://doi.org/10.1109/UEMCON47517.2019.8993089.

[16] A. Alzahrani, "Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13671–13680, Apr. 2024, https://doi.org/10.48084/etasr.7029.

[17] V. S. Kumar, Dr. S. A. Pearline, and S. R. Bose, "Real-Time Plant Species Recognition Using Non-Averaged Densenet-169 Deep Learning Paradigm," May 2022, https://doi.org/10.2139/ssrn.4111804.

[18] A. Alayed, R. Alidrisi, E. Feras, S. Aboukozzana, and A. Alomayri, "Real-Time Inspection of Fire Safety Equipment using Computer Vision and Deep Learning," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13290–13298, Apr. 2024, https://doi.org/10.48084/etasr.6753.

[19] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *Journal of Physics: Conference Series*, vol. 1004, no. 1, Dec. 2018, Art. no. 012029, https://doi.org/10.1088/1742-6596/1004/1/012029.

[20] "What is YOLOv8? The Ultimate Guide." https://blog.roboflow.com/whats-new-in-yolov8/.

[21] "How to Activate GPU Computing in Google Colab." https://saturncloud.io/blog/how-to-activate-gpu-computing-in-google-colab/.

[22] B. Selcuk and T. Serif, "A Comparison of YOLOv5 and YOLOv8 in the Context of Mobile UI Detection," in *Mobile Web and Intelligent Information Systems*, Aug. 2023, pp. 161–174, https://doi.org/10.1007/978-3-031-39764-6_11.

[23] A. Jabbar, S. Naseem, T. Mahmood, T. Saba, F. S. Alamri, and A. Rehman, "Brain Tumor Detection and Multi-Grade Segmentation Through Hybrid Caps-VGGNet Model," *IEEE Access*, vol. 11, pp. 72518–72536, Jun. 2023, https://doi.org/10.1109/ACCESS.2023.3289224.

[24] H. Mohsen, E. S. A. El-Dahshan, E. S. M. El-Horbaty, and A. B. M. Salem, "Classification using deep learning neural networks for brain tumors," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 68–71, Jun. 2018, https://doi.org/10.1016/j.fcij.2017.12.001.

[25] A. Ari and D. Hanbay, "Deep learning based brain tumor classification and detection system," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 26, no. 5, pp. 2275–2286, Jan. 2018, https://doi.org/10.3906/elk-1801-8.

[26] G. Manogaran, P. M. Shakeel, A. S. Hassanein, P. M. Kumar, and G. C. Babu, "Machine Learning Approach-Based Gamma Distribution for Brain Tumor Detection and Data Sample Imbalance Analysis," in IEEE Access, vol. 7, pp. 12-19, Jan. 2019, https://doi.org/10.1109/ACCESS.2018.2878276.