Tumor, Cancer, and Aneurysm Classification on Brain CT Image using Convolutional Neural Network

1st Bintang Vieshe Mone
2301222036
School of Computing

2nd Ghina Khoerunnisa
2301222033
School of Computing

3rd Medina Diani Nastiti
2301222038
School of Computing

***Abstract*—Diagnosing brain disease poses challenges for neurologist, particularly following Computed Tomography (CT) scans, as the brain's neuron system is complicated. Other contributing factors include the unavailability of exact causes and markers for each disease type, as well as the limited number of patient diagnosed with brain diseases, making precise diagnosis challenging. Advances in technology have led to increased focus on studying brain CT images, particularly within the computer vision field. The most popular method used in these studies is Convolutional Neural Network (CNN). Various CNN architectures have been tested, and it has been proven that CNNs give the best results in image processing-related domains, including image classification, object detection, and image segmentation. This study proposes the creation of a classifier system using CNN architecture to process brain CT images and classify them into three types of brain disease: aneurysm, tumor, and cancer. Two experiments were conducted on the created classifier system, applying different hyperparameter settings to analyze the impacts resulting from changes in specific CNN architecture. The results indicate that increasing the learning rate yields significant changes in accuracy. Additionally, increasing the convolution strides leads to a notable decrease in accuracy on certain points compared to the baseline model's performance.**

***Index Terms*—Computed Tomography, Computer vision, Convolutional Neural Network, Brain imaging, Image processing, Classifier systems, Aneurysms, Tumors, Cancer**

I. Introduction

Computed tomography (CT) images are known in the medical field to help medical personnel detect illnesses or abnormalities of various human organs; one of them is an illness related to the brain [1]. Usually, after the scan is complete, the radiologist requires time to thoroughly review and interpret the images, crafting a detailed report outlining their findings. Typically taking a few hours, this process may also extend the time spent if the CT scan proves complex or necessitates collaboration with other medical professionals or experts. On the other hand, these experts are not always available [2]. Even to become experts, medical personnel need to do years of study and handle lots of cases to gain enough experience. With technological advances in image processing and combining them with human knowledge, these CT scan images can be processed through intelligent systems to help detect diseases, thus shortening the time spent analyzing CT scan results.

There are lots of studies related to processing CT images or classification of brain diseases that have happened for many years with various proposed novel approaches such as RELM [3], KELM [4], Convolutional Neural Network (CNN) [5]–[8], and various modified CNN frameworks such as Capsule Network (CapsNet) [9], Residual Network (ResNet) [10], Wavelet Convolutional Neural Network (WCNN) [11], etc. In 2016, Gao and Hui conducted a study on CT scan images with the focus on classifying Alzheimer's disease (AD), lesions, or normal aging. They use CNN as the method for classifying those diseases, and the results show that the CNN model was effective in predicting those diseases with an average accuracy of 86.8% [13]. Another study about the classification of CT scan images occurred in 2021 by Gautam and Raman. They conducted a study to classify brain CT scan images into hemorrhagic stroke, ischemic stroke, and normal brain conditions using the CNN method. The results show that the used model can achieve the highest accuracy of 98% [14].

Based on the background and several previous studies, we try to implement a popular image processing method called CNN as the classification system based on CT images to classify the three illnesses: aneurysm, cancer, and tumor. There are three problems that we discuss in this paper.

1) How is the effectiveness of the proposed CNN architecture in classifying brain CT scan images into aneurysm, cancer, and tumor classes?
2) How is the effect of the learning rate on the performance of the CNN model?
3) How is the effect of the convolution stride on the performance of the CNN model?

II. Material and Methods

The CT scan image classifier system is built using the CNN method. It consists of several stages, namely data preprocessing, model development with CNN, and model evaluation. Figure 1 displays all the stages of the classifier system.

A. Dataset Overview

The "Computed Tomography (CT) of the Brain" dataset provided in Kaggle [15] is a comprehensive collection of CT brain scans, providing detailed images of patients brains captured through computed tomography imaging. The dataset

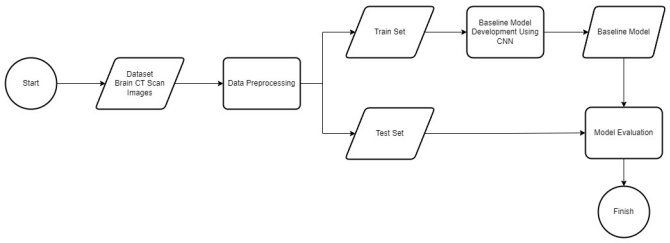

Fig. 1. System design flow

encompasses individuals diagnosed with one of three distinct brain diseases, including aneurysms, cancer, or tumors. Within the ”files” directory, each disease category divided into a specific folder and each of these folder contains CT scans of individuals diagnosed with the corresponding illness. Additionally, metadata stored in .csv format accompanies the dataset. This metadata includes hyperlinks to access files in both .dcm and .jpg formats, along with a corresponding ”type” field denoting the specific brain disease represented in the scan. The dataset consists of 256 CT images, with 83 CT images representing brain aneurysm, 83 CT images represent brain tumor, and 90 CT images represent brain cancer. Each image has the same size of 512x512 pixels. An example of these data can be seen in Figure 2

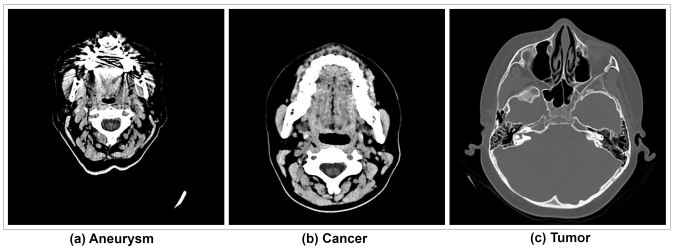

Fig. 2. Examples of the dataset for each classes (from left to right: Aneurysm, Cancer, and Tumor)

B. Data Pre-processing

The collected data was then subjected to data pre-processing. The data pre-processing step is executed to improve the quality of the data for research purposes. The first process is image normalization, which is applied to change the intensity range of image pixels to a standard range compatible with the CNN model. For instance, image pixels are generally measured in the range of 0-255. To standardize the image size, the pixels can be converted to a range of 0–1 by dividing each value by 255. The next step involves extracting grayscale data by collecting the first channel of each image in the dataset, assuming those images are grayscale channels. This simplifies the data for further processing. Afterwards, labels from the 'diseases type' column in the metadata file are extracted to be used as the classification target for each CT image. Then, the data obtained from pre-processing steps is split into a train set and a test set with an 8:2 ratio.

C. Convolutional Neural Network Architecture

A CNN is a specific form of neural network structure, and it has demonstrated its effectiveness in the field of computer vision many times. CNN utilizes several convolutional layers and pooling layers to extract hierarchical information from the input [12]. The convolution layers act as filters, identifying visual patterns, such as edges, corners, and textures, at various levels. The pooling layer is used to downsample the convolution results, reducing the data size and capturing the most important information [16]. CNN architecture excels in the domains of image classification, object detection, and picture segmentation [17].

The main advantages of CNN include their capacity to maintain spatial characteristics despite shifts and size variations in the image, as well as their ability to enable the model to recognize certain patterns regardless of their relative position and size [18]. Moreover, the incorporation of convolutional features enables the CNN to reduce the number of parameters involved in the learning process. This leads to more efficient models that can effectively address complex problems, especially those involving large datasets. CNN's advancements have rendered it a remarkably efficient method in numerous applications, such as facial recognition and medical image classification.

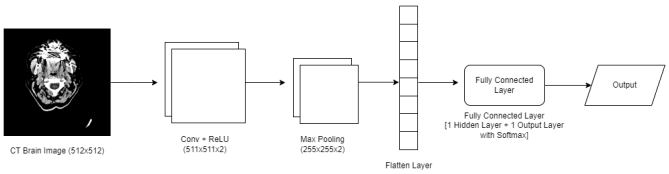

Fig. 3. The proposed architecture for classification of CT Brain images

Our system architecture, which uses CNN as the baseline method, consists of multiple layers that together form a model structure. Figure 3 illustrates our CNN architecture. Initially, the 2D convolutional layer extracts important features from the input image using two filters and a 2x2 convolution kernel with ReLU activation. The 2D convolutional layer produces an output matrix with dimensions of 511x511. Subsequently, a 2D MaxPooling layer is applied to reduce the output matrix dimension to 255x255, aiming to eliminate less important details.

Next, a dropout layer with a rate of 0.3 is applied to prevent overfitting in model performance by randomly ignoring some units during training. The results from this process are converted into one-dimensional vectors through the flatten layer, preparing them for the dense or fully connected layers. In the fully connected layers, the first hidden layer consists of 4 neurons and utilizes the ReLU activation function, generating an increasingly abstract representation of the input features. Finally, the output layer consists of three neurons corresponding to the three brain disease categories: class 0 for aneurysm, class 1 for cancer, and class 2 for tumor. Overall, this architecture is trained with a learning rate of 0.001, batch

size of 32, and epoch of 30 into a model that can classify brain CT images into the three disease categories. The details of the parameters used in convolution, max pooling, dropout, fully connected, and output layers are shown in Table I.

TABLE I
THE DETAILS OF THE PARAMETERS USED IN THE CNN ARCHITECTURE

Layer	Parameter	Value
Convolution	Filter	2
	Kernel Size	(2,2)
	Stride	(1,1)
	Activation	ReLU
Max Pooling	Pool Size	(2,2)
	Stride	(2,2)
Dropout	Rate	0.3
Fully Connected (Hidden Layer 1)	Units	4
	Activation	ReLU
Output Layer	Units	3
	Activation	Softmax

With the established CNN architecture, a series of experiments are conducted to explore the impact of two hyperparameter settings on the classifier's performance. The first hyperparameter setting is named Model with Learning Rate Variation (MLR), which adjusts the baseline model's architecture by adjusting the learning rate during the optimizer process. The baseline architecture utilizes a learning rate of 0.001, whereas the experiment investigates several learning rates, including 10, 1, 0.1, 0.01, and 0.0001. The objective of this exploration is to assess the impact of different learning rates on the model's ability to effectively train the input data while minimizing the loss function. This will provide valuable insights into identifying the most suitable learning rate for the specific task at hand.

The second hyperparameter setting is named Model with Convolution Stride Variation (MCS), which modifies the baseline architecture by adjusting the convolution stride during the convolutional process. The baseline model employs a stride of (1,1), while the experiment explores different strides, including (5,5), (10,10), (25,25), (50,50), and (100,100). This experiment seeks to examine the effect of modifying the convolution stride on the model's performance by analyzing how the spatial sampling of the input data is affected. The details of the CNN hyperparameter experiments that were conducted are provided in Table II

TABLE II
CNN HYPERPARAMETERS EXPERIMENTS

Abbr.	Description	Hyperparameter	Values
MLR	Model with an architecture similar to the baseline (learning rate = 0.001), but the learning rate is changed to analyze the effect of the learning rate on model performance	Learning Rate	[0.0001, 0.001, 0.01, 0.1, 1, and 10]
MCS	Model with an architecture similar to the baseline (convolution stride = 1,1), but the convolution stride are changed to analyze the effect of convolution stride on model performance	Convolution Stride	[(1,1), (5,5), (10,10), (25,25), (50,50), (100,100)]

D. Model Evaluation

Evaluation is the stage used to compare the obtained results from the implementation with existing criteria and standards to determine the success rate of the implementation. Model performance is evaluated using a confusion matrix, which summarizes the correct and incorrect predictions on the test set. The confusion matrix contains positive and negative values, as illustrated in Table III, which can be used to evaluate accuracy, precision, recall, and F1-score values. True Positive (TP) represents data that is actually positive and is classified as positive. True Negative (TN) represents data that is actually negative and is classified as negative. False Positive (FP) represents data that is actually negative but is classified as positive. False Negative (FN) represents data that is actually positive but is classified as negative.

TABLE III
CONFUSION MATRIX

Class		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative(FN)
	Negative	False Positive (FP)	True Negative (TN)

Accuracy (Q) measures the degree of accuracy between the predicted value (generated from the test set on the trained model) and the actual value (from the target data of the test set). It is used to evaluate the number of label predictions that match with the actual labels. The accuracy value directly correlates with the system's classification performance; a higher accuracy signifies superior performance.

Recall or Sensitivity (SE), represents the success rate of the system in retrieving information. Precision (PR) represents the system's accuracy in providing user-desired information compared to the system's generated responses. F1-Score is a parameter used to calculate the performance metric by combining the precision result with the recall results.

The formula used for accuracy, precision, recall, and F1-score can be found in equations (1), (2), (3), and (4).

$$Q = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$SE = \frac{TP}{TP+FN} \quad (2)$$

$$PR = \frac{TP}{TP+FP} \quad (3)$$

$$F1-Score = 2 \times \frac{PR \times SE}{PR+SE} \quad (4)$$

III. RESULT AND DISCUSSION

A. Result

According to the experiments conducted on the baseline architecture, the model achieves a high level of accuracy, with 99.02% on the train set and 98.08% on the test set. The precision, recall, and F1-score metrics from the classification report for each brain disease class in the test set are also displayed in Table IV. The classification report provides an assessment of the model's performance on the test set, specifically for three distinct classes.

For the aneurysm class, the model achieved a precision score of 0.93, meaning that 93% of the data from the test set predicted as class aneurysm were accurately classified. It also achieves recall scores of 1.00 for the aneurysm class, indicating that the model effectively captured all actual instances of this class. The model achieves a F1-score of 0.97 for the aneurysm class, indicating a strong overall performance.

TABLE IV
THE PERFORMANCE RESULT OF THE BASELINE MODEL

Model	Class	Q	PR	SE	F1-score
Baseline	0 (Aneurysm)	0.98	0.93	1.00	0.97
	1 (Cancer)		1.00	0.94	0.97
	2 (Tumor)		1.00	1.00	1.00

Regarding the cancer class, the baseline model achieved precision scores of 1.00, indicating that all instances from the test set predicted as cancer class were accurately classified. However, the achieved recall score is 0.94, signifying that the model failed to correctly identify certain cancer cases. The model achieved an F1-score of 0.97 for the cancer class, indicating a well-balanced performance in terms of precision and recall.

For both precision and recall obtained by the model for the tumor class, optimal values of 1.00 were achieved. This indicates that the model correctly categorized all cases belonging to the tumor class without any incorrect positive or negative classifications. The model achieved an F1-score of 1.00 for the tumor class, indicating its outstanding performance in this category.

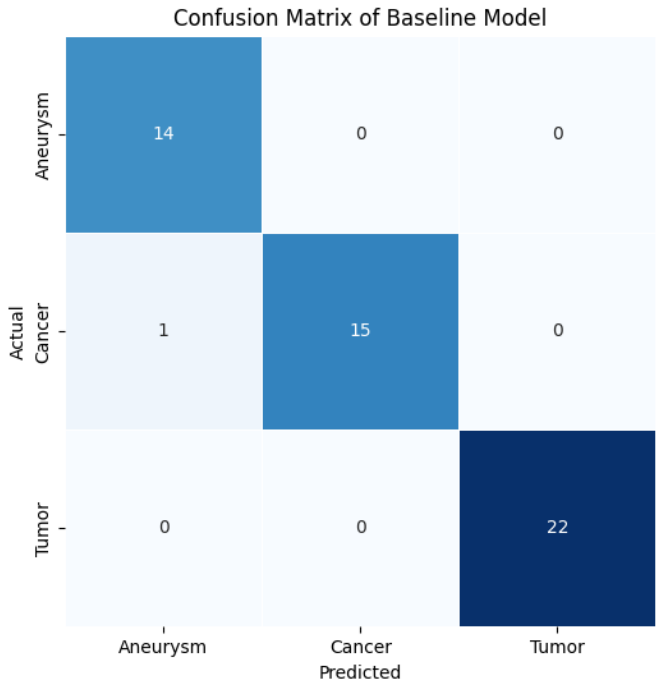

Fig. 4. The confusion matrix of baseline model

Figure 4 represents the confusion matrix of the results from the baseline model on three classes, with each row corresponding to the actual class and each column corresponding to the predicted class. Based on the figure, it can be observed that all instances of the aneurysm and tumor classes were predicted correctly. As for the cancer class, 15 data points were correctly predicted, and one data point was predicted as another class, namely the aneurysm class.

TABLE V
THE PERFORMANCE RESULT OF THE HYPERPARAMETER OBSERVATIONS

Parameter	Value	Q
Learning Rate (MLR)	0.0001	0.71
	0.001	**0.98**
	0.01	0.96
	0.1	0.27
	1	0.31
	10	0.27
Convolution Stride (MCS)	(1,1)	**0.98**
	(5, 5)	**0.98**
	(10, 10)	**0.98**
	(25, 25)	0.69
	(50, 50)	0.64
	(100, 100)	0.56

Table V outlines the impact of different hyperparameter values on model performance, focusing on learning rates (MLR) and convolution stride (MCS). When considering the learning rates of 0.0001, 0.001, 0.01, 0.1, 1, and 10, the model's performance varied inconsistently. The model achieved the best accuracy on the train set with a learning rate of 0.01 and 0.001, approximately 0.99 or 99%, while the best accuracy on the test set was obtained with a learning rate of 0.001, approximately 0.98 or 98%. Regarding the convolution stride, the stride values of 1 and 10 exhibited the best performances, achieving the same accuracy on both the train and test sets, with 0.99 or 99% for the train set and 0.98 or 98% for the test set.

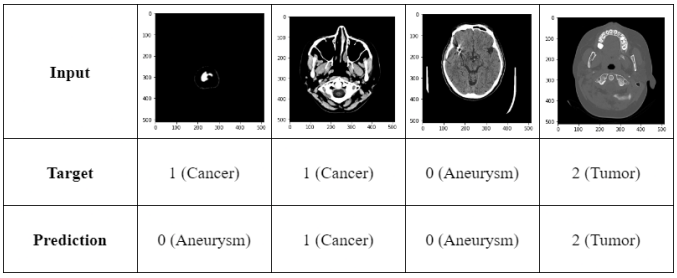

Fig. 5. Output samples of the classification results

Based on the confusion matrix in Figure 4, there is one image that is misclassified, namely a CT scan image with an actual class 1 (cancer). The model classifies the image into class 0 (aneurysm). The output sample of classification results can be seen in Figure 5.

B. Discussion

Based on the earlier results, the CNN model achieves high accuracy in classifying CT images by optimizing key

hyperparameters: learning rate and convolution stride. In the learning rate experiment, the accuracy of the test set decreased when using values of 0.0001, 0.01, 0.1, 1, and 10, compared to the initial model accuracy of 98%. This suggests that increasing the learning rate does not always lead to improved CNN accuracy. Even when the learning rate is lower than the ones used on the baseline model, it only achieves an accuracy less than 80%, and this may worsen with further reductions in learning rate below 0.0001. This shows that selecting the right learning rate is important to obtain an optimal model. The learning rate itself is the step size at each training iteration when moving towards the optimal sparse categorical crossentropy loss function. From the experiment, it can be seen that when the step size is too large, the model fails to achieve optimal results because it may pass the optimal point due to the large step. In addition, when the step size is reduced, it also experiences a decrease in performance. This happens because when the step size is too small, the model fails to reach the optimal value because it moves too slowly during training, so the model has not reached the optimal point. However, since it has already finished its iteration, the training process stops and produces a model that has not reached the optimal value.

Similarly, in the convolutional stride case, increasing the hyperparameter values does not consistently give a result with higher accuracy. Convolution strides of (1,1), (5,5), and (10,10) are able to achieve accuracy rates up to 98%, but when they are increased to (25,25), the model's performance starts to decrease. The decline becomes more visible when the stride value increases to (100,100), decreasing the model's performance to 56%. This shows that increasing the stride value in the convolution layer can have a significant impact on model performance. Stride controls how far the filter moves each time it convolves the input. A larger stride can reduce the output dimension of the convolution layer, which can result in the loss of important spatial information. On the other hand, a smaller stride can lead to a more precise model but requires more computation. Next, according to Figure 5, the image that the model misclassifies is one that is not as clear as the other images. This may confuse the model. Further research on improving the performance of the model can be done by improving the quality of training data.

This study has limitations, mainly due to the near-perfect accuracy achieved, which is attributed to a limited dataset. To address this, image data augmentation techniques can be applied to the available image dataset to generate new image data. Common techniques such as random rotation, random scaling, blurring, and noise additions are often used in studies related to medical images, including CT scan images and MR images [18]. By applying these techniques, the study may yield different and better results compared to those obtained in the current analysis.

IV. Conclusions

This study aimed to analyze a dataset containing CT images of patients with brain diseases using a classifier model with a CNN architecture. The baseline model achieved an accuracy of 0.98, indicating strong classification performance from the baseline model. The study further involved modifying key CNN hyperparameters, such as learning rate and convolution stride, to assess their impact on model performance and determine optimal values. The experiments revealed that the model's performance significantly improved with a learning rate of 0.001. Similarly, a model with a convolution stride of 1 or 10 demonstrated the same improvement, with both values outperforming other stride variations.

In summary, the experiments show that adjusting either the learning rate or convolution stride can improve the model's performance. These findings highlight the importance of hyperparameter tuning in optimizing the CNN model for the classification of brain diseases from CT images.

References

[1] S. Hussain et al., "Modern Diagnostic Imaging Technique Applications and Risk Factors in the Medical Field: A Review," BioMed Research International, vol. 2022, no. 5164970, pp. 1–19, Jun. 2022, doi: https://doi.org/10.1155/2022/5164970.

[2] M. D. Hssayeni, M. S. Croock, A. D. Salman, H. F. Al-khafaji, Z. A. Yahya, and B. Ghoraani, "Intracranial Hemorrhage Segmentation Using a Deep Convolutional Model," Data, vol. 5, no. 1, p. 14, Feb. 2020, doi: https://doi.org/10.3390/data5010014.

[3] A. Gumaei, M. M. Hassan, M. R. Hassan, A. Alelaiwi, and G. Fortino, "A Hybrid Feature Extraction Method With Regularized Extreme Learning Machine for Brain Tumor Classification," IEEE Access, vol. 7, pp. 36266–36273, 2019, doi: https://doi.org/10.1109/access.2019.2904145.

[4] A. Pashaei, Hedieh Sajedi, and N. Jazayeri, "Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines," International Conference on Computer and Knowledge Engineering, Oct. 2018, doi: https://doi.org/10.1109/iccke.2018.8566571.

[5] Ö. Polat and C. Güngen, "Classification of brain tumors from MR images using deep transfer learning," The Journal of Supercomputing, Jan. 2021, doi: https://doi.org/10.1007/s11227-020-03572-9.

[6] X. W. Gao, R. Hui, and Z. Tian, "Classification of CT brain images based on deep learning networks," Computer Methods and Programs in Biomedicine, vol. 138, pp. 49–56, Jan. 2017, doi: https://doi.org/10.1016/j.cmpb.2016.10.007.

[7] 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, doi: https://doi.org/10.3390/diagnostics12081850.

[8] A. Basu, K. H. Sheikh, E. Cuevas, and R. Sarkar, "COVID-19 detection from CT scans using a two-stage framework," Expert Systems with Applications, vol. 193, p. 116377, May 2022, doi: https://doi.org/10.1016/j.eswa.2021.116377.

[9] P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Capsule Networks for Brain Tumor Classification Based on MRI Images and Coarse Tumor Boundaries," IEEE Xplore, pp. 1368–1372, May 2019, doi: https://doi.org/10.1109/ICASSP.2019.8683759.

[10] A. Foysal, A. Hossain, A. Yassine, and M. Shamim Hossain, "Detection of COVID-19 Case from Chest CT Images Using Deformable Deep Convolutional Neural Network," Journal of Healthcare Engineering, vol. 2023, pp. 1–12, Feb. 2023, doi: https://doi.org/10.1155/2023/4301745.

[11] P. Moisés et al., "A new model for classification of medical CT images using CNN: a COVID-19 case study," Multimedia Tools and Applications, vol. 82, no. 16, pp. 25327–25355, Dec. 2022, doi: https://doi.org/10.1007/s11042-022-14316-7.

[12] X. W. Gao and R. Hui, "A deep learning based approach to classification of CT brain images," 2016 SAI Computing Conference (SAI), Jul. 2016, doi: https://doi.org/10.1109/sai.2016.7555958.

[13] X. W. Gao and R. Hui, "A deep learning based approach to classification of CT brain images," 2016 SAI Computing Conference (SAI), Jul. 2016, doi: https://doi.org/10.1109/sai.2016.7555958.

[14] A. Gautam and B. Raman, “Towards effective classification of brain hemorrhagic and ischemic stroke using CNN,” Biomedical Signal Processing and Control, vol. 63, p. 102178, Jan. 2021, doi: https://doi.org/10.1016/j.bspc.2020.102178.
[15] “Computed Tomography (CT) of the Brain,” www.kaggle.com. https://www.kaggle.com/datasets/trainingdatapro/computed-tomography-ct-of-the-brain/ (accessed Dec. 29, 2023).
[16] D. Raval and J. N. Undavia, “A Comprehensive assessment of Convolutional Neural Networks for skin and oral cancer detection using medical images,” Healthcare analytics, vol. 3, pp. 100199–100199, Nov. 2023, doi: https://doi.org/10.1016/j.health.2023.100199.
[17] M. M. Taye, “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,” Computation, vol. 11, no. 3, p. 52, Mar. 2023, doi: https://doi.org/10.3390/computation11030052.
[18] Moez Krichen, “Convolutional Neural Networks: A Survey,” Computers, vol. 12, no. 8, pp. 151–151, Jul. 2023, doi: https://doi.org/10.3390/computers12080151.
[19] Saeedi, Soheila, et al. ”MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques.” BMC Medical Informatics and Decision Making 23.1 (2023): 16.
[20] Goceri, Evgin. ”Medical image data augmentation: techniques, comparisons and interpretations.” Artificial Intelligence Review (2023): 1-45.