

Review

Open Access



# Toward the detection of intracranial hemorrhage: a transfer learning approach

Jayesh Soni 

Applied Research Center, Florida International University, Miami, FL 33172, USA.

**Correspondence to:** Dr. Jayesh Soni, Applied Research Center, Florida International University, 10555 W Flagler St, Miami, FL 33172, USA. E-mail: jsoni@fiu.edu

**How to cite this article:** Soni J. Toward the detection of intracranial hemorrhage: a transfer learning approach. *Art Int Surg*. 2025;5:221-38. <https://dx.doi.org/10.20517/ais.2024.46>

**Received:** 20 Jun 2024 **First Decision:** 20 Jan 2025 **Revised:** 22 Mar 2025 **Accepted:** 27 Mar 2025 **Published:** 8 May 2025

**Academic Editor:** Andrew Gumbs **Copy Editor:** Pei-Yun Wang **Production Editor:** Pei-Yun Wang

## Abstract

Traumatic brain injury can cause intracranial hemorrhages, which, if not diagnosed and treated early, may lead to fatal outcomes due to excessive bleeding inside the cranium. In the United States, stroke is the fifth-leading cause of death, and approximately 10% of strokes result from intracranial hemorrhages. Identifying the presence, location, and type of hemorrhage is a critical step in treating emergency room patients. To determine the position and size of hemorrhages, X-ray computed tomography (CT) scans are commonly employed. While radiologists are highly skilled in analyzing CT scan images, the process is time-consuming. This study deals with intracranial hemorrhage detection using a deep-learning approach. First, we introduce convolutional neural networks (CNNs), a type of neural network designed for image-based datasets. Subsequently, we discuss various hyperparameter optimization techniques to enhance CNN training efficiency. As CNN training can be computationally expensive and time-intensive in many instances, we address this challenge by leveraging transfer learning with pre-trained models. We explore different transfer learning architectures, including VGGNet, AlexNet, EfficientNetB2, ResNet, MobileNet, and InceptionNet. The proposed framework for intracranial hemorrhage detection is implemented using transfer learning on the Radiological Society of North America Intracranial Hemorrhage Detection dataset, which is publicly available on Kaggle. Specifically, VGGNet is employed within this framework using powerful deep-learning libraries such as TensorFlow. This methodology can also be generalized to classify CT scan images in other biomedical domains.

**Keywords:** Intracranial hemorrhage detection, transfer learning, VGGNet, CTScan, Tensorflow



© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



## INTRODUCTION

Intracranial hemorrhage refers to bleeding inside the skull and is one of the significant causes of morbidity, stroke, and mortality<sup>[1]</sup>. It can transpire as a result of trauma or spontaneously<sup>[2-4]</sup>. Spontaneous intracranial hemorrhage can be connected to cerebral amyloid angiopathy, hypertension, broken aneurysms, tumors, and hemorrhagic strokes<sup>[5,6]</sup>. Traumatic intracranial hemorrhage can affect anyone who experiences head trauma, but patients on anticoagulation are at considerably more risk. Intracranial hemorrhages are categorized into four types: epidural, subdural, subarachnoid, and intraparenchymal. Brain computed tomography (CT) scans are frequently used for evaluation, as changes in X-ray attenuation and the position of hemorrhages on these scans enable the detection and differentiation of diverse types of intracranial hemorrhage<sup>[7]</sup>. Hemorrhages may be subtle for several reasons, including their small size and the pooling of blood with similar attenuation to surrounding normal brain tissue. Additionally, conditions like calcifications in the choroid plexus or pineal glands can sometimes be mistaken for intracranial hemorrhages. Every year, numerous CT scans are performed at trauma centers to decide if head trauma has triggered an intracranial hemorrhage. While radiologists are typically able to detect hemorrhages accurately, an “additional set of eyes” to assist in identifying subtle cases of intracranial hemorrhages could be beneficial. Marking images with likely substantial hemorrhages could help prioritize cases for the radiologist’s review, potentially reducing the time required for diagnosis. Automated detection and quantification of hemorrhages could facilitate rapid assessment across multiple scans. This work applies a transfer learning technique to detect intracranial hemorrhages based on CT scan images.

The subsequent sections are organized as follows: Section “RELATED WORK” discusses related work. Section “IMAGE PROCESSING TECHNIQUES” provides an overview of image processing techniques and a deep learning-based convolution neural network. Section “TRANSFER LEARNING” explains transfer learning with various pre-trained models. Section “IMPLEMENTATION” presents practical real-time use cases of the VGG-16 for detecting intracranial hemorrhage. Finally, Section “CONCLUSION” concludes the paper.

## RELATED WORK

Learning-based algorithms are very efficient at solving complex problems<sup>[8-11]</sup>. Convolutional neural networks (CNNs) are commonly employed for medical image segmentation tasks. Hussain *et al.* proposed a method involving two such networks to segment glioma tumors<sup>[12]</sup>. This new stacked architecture achieved results comparable to those of other leading architectures in the field. Arslan *et al.* applied CNNs to assist individuals with spinal cord disabilities<sup>[13]</sup>. Their proposed human-machine interface uses eye movements to control the mouse and the network outperformed various other models in the literature when validated and tested against a handcrafted dataset.

Zhang *et al.* introduced a lesion synthesis approach to generate data aimed at improving the robustness of deep learning models for detecting intracranial hemorrhages<sup>[14]</sup>. The results demonstrated that synthetic data augmentation techniques can effectively improve classification accuracy in medical imaging. Sengupta *et al.* focused on using 3D CNNs for intracranial hemorrhage detection from CT images<sup>[15]</sup>. They emphasized the benefits of volumetric image analysis and showed that 3D CNNs outperform 2D models in capturing spatial dependencies, leading to better diagnostic accuracy.

A hybrid CNN-LSTM model for diagnosis in successive CT scans was proposed by Nguyen *et al.*<sup>[16]</sup>. The CNN component extracts spatial characteristics, whereas the LSTM model captures temporal dependencies between slices. Together, these models enhance detection sensitivity, making them more suitable for clinical application. Ngo *et al.* proposed a slice-level detection approach that leverages deep descriptors from

neighboring CT slices<sup>[17]</sup>. The sensitivity results demonstrated by the model in detecting hemorrhages against normal brain tissue provide adequate evidence that challenges related to variations in CT images at the slice level can be effectively addressed.

Helwan *et al.* applied deep learning techniques to classify CT images as healthy or hemorrhagic, using autoencoders for this task<sup>[18]</sup>. Their model, consisting of three hidden layers, achieved the lowest MSE and the highest classification accuracy, despite training on a small dataset. Mahajan *et al.* enhanced brain hemorrhage detection by integrating an artificial neural network with a watershed algorithm using CT scan images<sup>[19]</sup>. They utilized a grey-level co-occurrence matrix to extract features, which were then input into the neural network to classify hemorrhage types. Their findings indicated that image processing techniques, such as advanced segmentation and noise elimination, are beneficial for precise hemorrhage identification.

Gong *et al.* proposed a method to classify brain CT images as either normal or hemorrhagic by dividing the images into two regions<sup>[20]</sup>. However, the resulting dataset was highly imbalanced, with most images depicting normal cases. To address this, the researchers applied techniques such as wavelet disintegration, noise removal, and ellipse correctness. Their model achieved a recall score of 0.88 and a precision score of 0.83.

## IMAGE PROCESSING TECHNIQUES

### Statistical-based algorithms

Figure 1 illustrates the different methods used for image analysis. Open source computer vision (OpenCV) is an image processing library widely used in the computer vision research community. It offers an extensive range of functions to mold and transform input images into an output format of our choice. In color images, pixel values are represented across three channels. The pixel value in each channel can vary between 0-255. In contrast, for black-and-white images, the pixel values are either 0 (black) or 1 (white).

Color images are stored on computers as RGB (Red, Green, Blue) pixel values. The computer interprets this as an array of numbers, which represents the image. These color images are stored in a three-dimensional data array, where the first two dimensions correspond to the image's height and width, and the third dimension represents the percentage of red, green, and blue for each pixel.

### Learning-based algorithms

With the increasing computational power and rapid data generation, deep learning algorithms are now applied across nearly all fields<sup>[21]</sup>. One such algorithm is the CNN, as shown in Figure 2.

The three main layers of a CNN are:

- (i) Convolutional Layer: The input image is fed into this layer, where a convolution operation is performed to extract meaningful information or features from the image. Additional convolution operations can be applied to capture even more distinct features.
- (ii) Pooling Layer: Located immediately after the convolution layer, this layer reduces the dimensionality of the image by applying operations such as maximum, minimum, or other statistical functions. The resulting data are then passed through activation functions like rectifier linear unit (ReLU) to introduce nonlinearity.
- (iii) Fully Connected layer: The final layer connects all the neurons in a dense manner. The output from the pooling layer is flattened and passed on for tasks such as classification, image segmentation, or object detection.

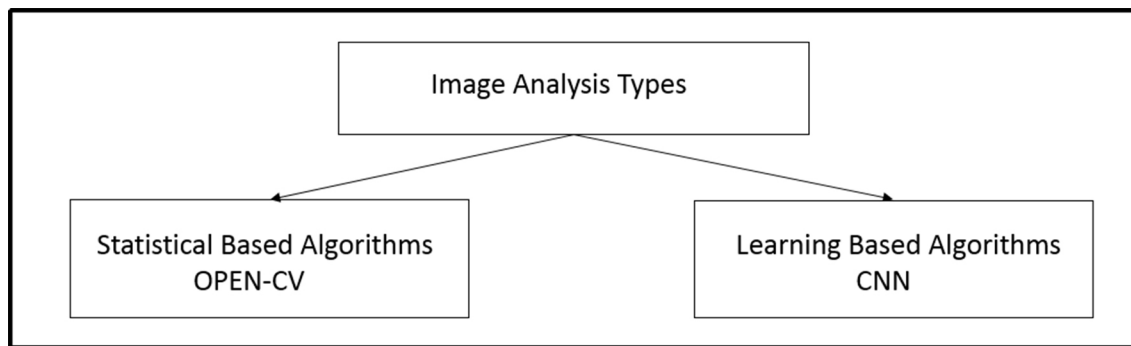


Figure 1. Image processing techniques.

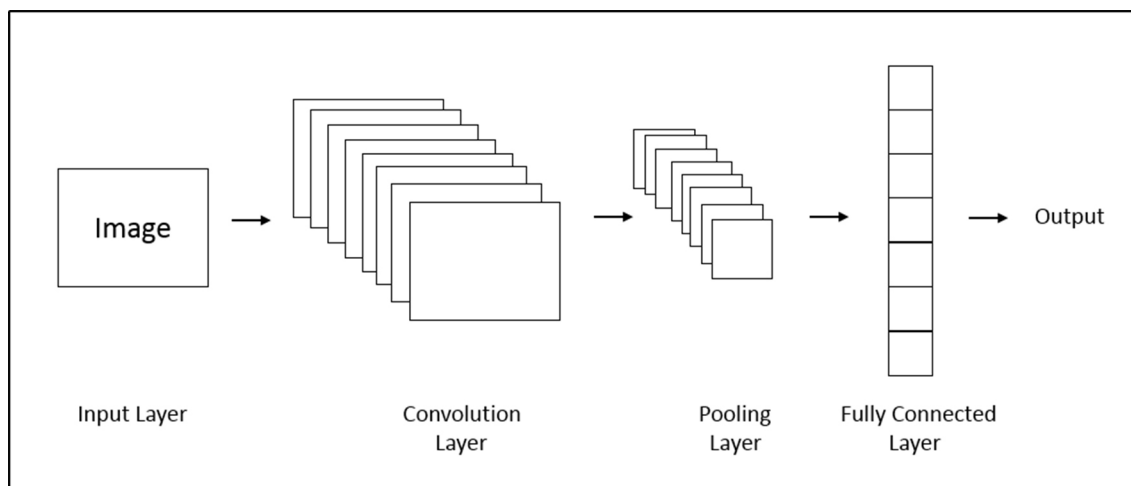


Figure 2. CNN. CNN: Convolutional neural network.

### Hyperparameters of CNNs

Learning-based algorithms are critical to modern computing, enabling machines to recognize patterns, make predictions, and even understand natural language. However, the performance of these algorithms can be highly sensitive to their hyperparameters - configuration settings that are not learned during training. Examples of hyperparameters include learning rates, regularization strengths, and the number of hidden layers in a neural network.

Setting these hyperparameters correctly is crucial for achieving optimal results. The chosen values should allow the algorithm to generalize well to new data, rather than simply memorizing the training set. Hyperparameter tuning has become an active area of research in machine learning, with many methods and tools developed to automate this process.

#### *Learning rate*

The optimization of weights in a neural network depends on the speed of gradient descent, which is controlled by the learning rate. This parameter can either remain fixed throughout training or gradually increase or decrease. Optimizers such as Adaptive Delta, Adaptive Momentum, RMSprop, and Stochastic Gradient Descent commonly adjust the learning rate during training.

### *Number of epochs*

This parameter determines how many times the neural network will pass through the entire training dataset. The number of epochs is important because it decides how often the network learns from the data. Normally, more epochs would mean better performance, but beyond a certain point, excessive epochs can cause overfitting, making the model too volatile. The challenge is to identify the optimal number of epochs that achieve the desired performance without overfitting.

### *Batch size*

The batch size determines the number of training samples used in each iteration. Weight updates are calculated based on the loss value of the batch. The size of the batch directly affects both the speed and stability of the training process. Smaller batch sizes lead to more precise weight updates but slow down the training process. Larger batch sizes are more computationally efficient but result in less accurate weight updates.

### *Activation function*

Activation functions introduce nonlinearity into the neural network, enabling the construction of models capable of handling complex input-output relationships. For example, the ReLu function is commonly used in CNNs. Other functions, such as the hyperbolic tangent (tanh) and sigmoid functions, are typically used in other types of neural networks. The choice of activation function depends on the specific problem and the nature of the data.

### *Hidden layers and units*

The total number of neurons and hidden layers in a network can significantly impact model performance. A small number of hidden layers and neurons may lead to underfitting, while too many can cause overfitting. Regularization techniques, such as L1 or L2 regularization, can be employed to balance the trade-off between model complexity and performance. The optimal number of hidden layers and units is typically determined through experimentation and meticulous hyperparameter tuning.

### *Weight initialization*

Weights are typically initialized with small random values that are uniformly distributed.

### *Dropout for regularization*

Dropout is a regularization technique designed to randomly deactivate a certain percentage of neurons during training. For example, if the dropout rate is set to 0.2, 20% of the neurons in that layer will be randomly deactivated at each training epoch. Therefore, the remaining active neurons learn more robust features, ensuring the system does not overfit. In principle, dropout can be applied to any layer in the network, but it is most commonly used in fully connected layers, where it effectively prevents overfitting in deep neural networks with multiple layers. This discussion assumes that dropout is applied during training, particularly on a validation set, and it is quite less likely to prevent overfitting during final deployment or inference, as all neurons remain active in inference mode, and predictions are made using the entire trained network.

### *Kernel size*

The kernel size refers to the dimensions of the filter used to extract features from an image. Different kernel sizes capture different types of features.

### *Pooling*

The pooling layer performs downsampling by partitioning the input image into a set of non-overlapping rectangular regions called pooling windows. A pooling function, such as max-pooling or min-pooling, then reduces each window to a single value. This process helps retain the most significant information while suppressing irrelevant details, improving computational efficiency and reducing the risk of overfitting. Max-pooling, the most commonly used technique, selects the maximum value within each pooling window, making it particularly effective in identifying features that remain invariant to small translations or rotations in the input image. In contrast, min-pooling selects the minimum value in each pooling window and is less commonly used, though it can be useful for detecting dark regions or edges. Other pooling functions, such as average pooling, are also available, but max-pooling remains the most popular pooling technique in CNNs due to its practical effectiveness.

Hyperparameter tuning is an essential step in optimizing model performance by selecting the best combination of hyperparameters for a given dataset. However, manually tuning hyperparameters can be tedious and time-consuming. Two popular techniques that expedite the process are grid search and randomized search.

Grid search systematically evaluates all possible combinations of hyperparameters specified in a predefined search space. While this exhaustive approach guarantees finding the optimal hyperparameter values for a given model architecture and dataset, it is computationally expensive.

Randomized search, on the other hand, randomly samples hyperparameters from a predefined search space. This method is particularly useful when the search space is large, making exhaustive evaluation impractical. Although randomized search does not guarantee absolute optimal values, it can efficiently identify good hyperparameter settings, significantly reducing computational cost.

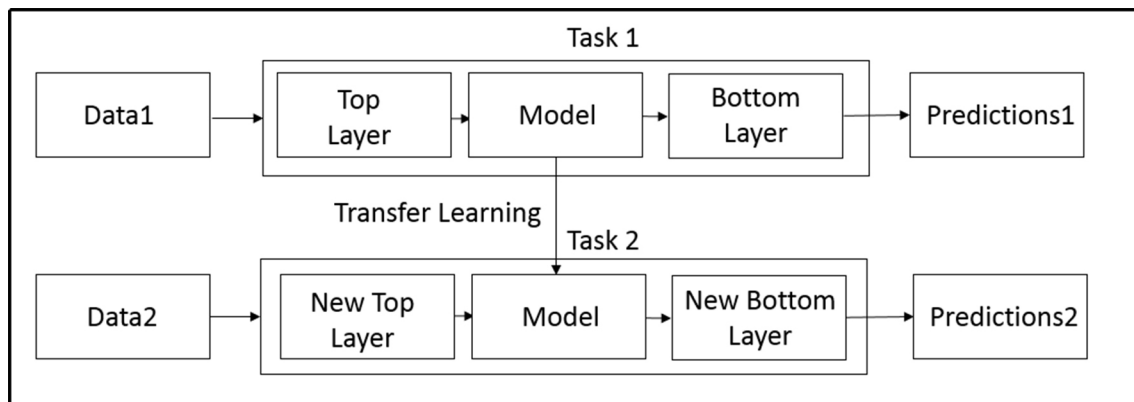
## **TRANSFER LEARNING**

As shown in [Figure 3](#), transfer learning is a technique where a model, already trained on one problem domain, is applied to solve a problem in a different domain.

There are several approaches to transfer learning:

**Domain adaptation:** Domain adaptation is used when there is a marginal difference between the target domain of the pre-trained model and the new problem being addressed. For example, a model pre-trained on a movie reviews dataset to classify reviews as positive and negative would differ from a model designed for product reviews. In such cases, domain adaptation methods are applied. These methods extract different features from different layers of the neural network, which can then be used to generate domain-invariant features, enabling quick transfer across multiple domains.

**One-shot learning:** Deep learning algorithms typically require large amounts of data and computational power to learn optimal weights. However, this is not how human learning works. A child can learn a new task from just a few examples. One-shot learning is a type of transfer learning where the model learns the decision function using only a few training examples. This approach is particularly useful in real-world scenarios where it is extremely difficult to obtain labeled datasets for all categories, especially in classification tasks.



**Figure 3.** Transfer learning approach.

**Zero-shot learning:** Zero-shot learning offers a great alternative to traditional transfer learning, as it does not require a labeled dataset for training. It is useful in scenarios such as machine translation, where a target sentence may not be available in the target language. In these cases, the model can still perform tasks despite lacking direct labeled data for every possible outcome.

The ImageNet large scale visual recognition challenge (ILSVRC)<sup>[22]</sup> is an annual, highly competitive event where researchers aim to develop the most accurate image classification models. To aid participants, the competition provides access to a vast dataset, including 1.2 million training images, 50,000 validation images, and 100,000 test images, covering a total of 1,000 different categories.

The following section will discuss various pre-trained models.

### AlexNet

Developed in 2012, AlexNet<sup>[23]</sup> marked a significant breakthrough in the advancement of CNN networks. It won the ILSVRC competition by reducing the error rate by half. One of its key innovations was leveraging GPUs for accelerated computation, particularly in implementing the ReLU activation function. Additionally, AlexNet introduced a stride of 4 pixels in the pooling layer, further contributing to error reduction.

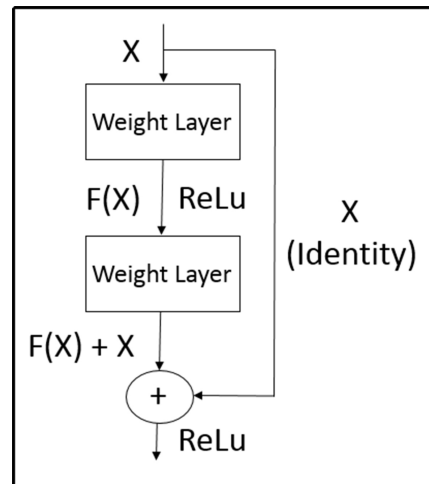
### Visual geometry group of Oxford University (VGGNet)

VGG<sup>[24]</sup> represents an advancement over AlexNet, primarily due to its distinctive architecture and use of smaller convolutional kernels. Instead of a single oversized filter, VGG employs multiple smaller filters, which enhances feature extraction. This approach is based on the concept of receptive fields: in a  $5 \times 5$  kernel, each pixel gathers information from 25 pixels in the previous layer. A larger receptive field enables the network to capture more complex patterns, whereas smaller receptive fields may struggle to detect overarching structures. As a result, VGGNet adopts a deeper network architecture to improve feature representation.

### ResNet

The core innovation of ResNet is the introduction of identity connections<sup>[25]</sup>, which allow the network to bypass one or more layers, as shown in Figure 4.





**Figure 4.** Skip connection.

This skip connection differs from the standard output. In the conventional process, weights are multiplied by  $X$ , and the bias term is added before passing the result through the activation function to get the output,  $M(x)$ :

$$M(x) = f(wx + b) \quad (1)$$

However, with the skip connection, the output changes to:

$$M(x) = f(x) + x \quad (2)$$

To match dimensional requirements, zero-padding is applied to the skip connection, and  $1 \times 1$  convolutional layers are added to the input layer. This results in the updated output:

$$M(x) = f(x) + w_1(x) \quad (3)$$

The only modification here is the introduction of an additional parameter,  $w_1$ .

In deep CNNs, ResNet's skip connections help eliminate the vanishing gradient problem by allowing gradients to propagate through shortcut paths. This not only facilitates stable training but also enhances model performance by improving regularization. Inspired by VGG-19, ResNet features a 34-layer architecture with skip connections, also referred to as residual blocks.

### InceptionNet

One of the key components of this CNN is the Inception Layer<sup>[26]</sup>, illustrated in Figure 5. This layer integrates multiple convolutional operations -  $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$  convolutions - alongside a parallel max-pooling layer. The outputs from these layers are concatenated into a single vector, which is then forwarded to the next layer in the network.

The  $1 \times 1$  convolutional layer serves as a dimensionality reduction mechanism, enabling the network to efficiently process large volumes of data. By reducing the input dimensionality, it helps the network extract



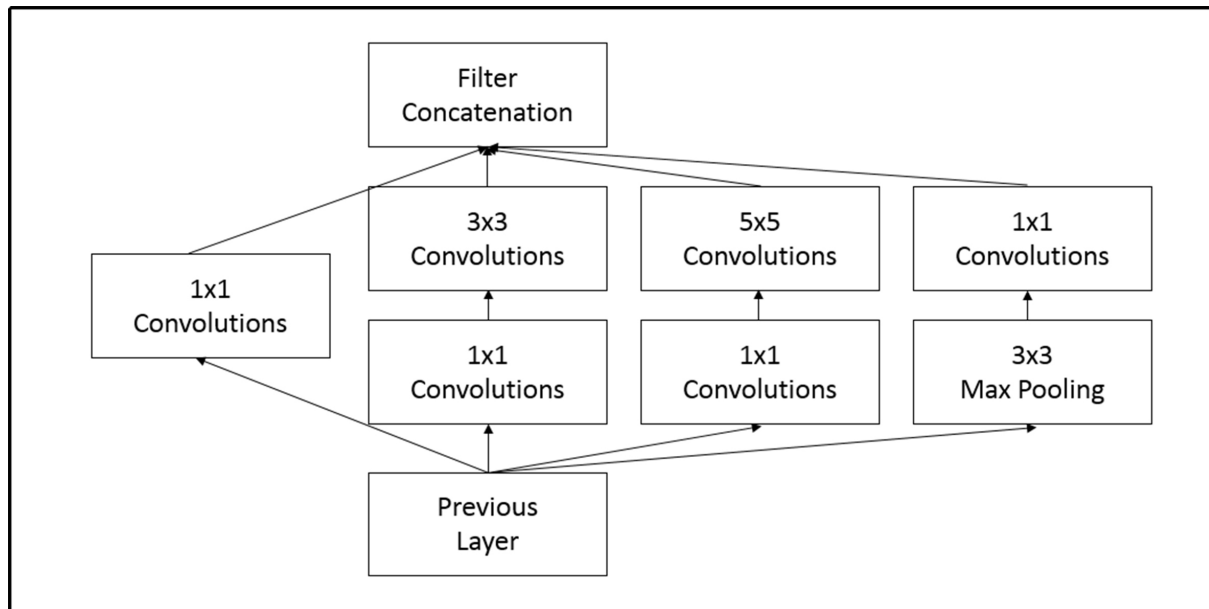


Figure 5. Inception layer.

more salient features, thereby enhancing its predictive accuracy.

### EfficientNet

EfficientNet<sup>[27]</sup> is a CNN architecture that uses a novel scaling method based on a compound coefficient. This approach optimally balances model depth, width, and input resolution, enabling high performance with relatively few parameters while maintaining computational efficiency.

The EfficientNet architecture has demonstrated state-of-the-art performance on several benchmark datasets. For instance, on the CIFAR-100 dataset, it attains an accuracy of 0.91% while using significantly fewer parameters than most contemporary models. Additionally, EfficientNet has been successfully applied to various image classification tasks, such as object detection and localization, yielding highly competitive results.

### MobileNet

MobileNet<sup>[28]</sup> is designed for embedded and mobile vision computation applications. It utilizes a novel technique called depth-wise separable convolution, which significantly reduces model size, minimizes the number of parameters, and lowers computational costs.

In the MobileNet architecture, the depth-wise convolution filter executes a separate convolution to each input channel independently, significantly reducing the computational complexity of convolutional layers. This efficient design enables MobileNet to make predictions with lower resource consumption compared to traditional CNNs.

By leveraging depth-wise discrete convolutions, MobileNet achieves state-of-the-art performance in various image classification tasks while demanding fewer parameters and computations than conventional CNNs. Thus, it is particularly well-suited for mobile and embedded devices, where efficiency is paramount.

### Effectiveness of transfer learning methods

Transfer learning encompasses various techniques, including domain adaptation, one-shot learning, and zero-shot learning, which are particularly useful in scenarios with limited labeled data. These methods tend to be more effective when labeled data are scarce and target domains share a greater similarity:

Domain Adaptation optimizes the feature mapping of a pre-trained network to better suit a closely related sub-domain.

One-shot learning leverages feature similarity to classify new instances using only a single example.

Zero-shot learning enables a model to categorize entirely unseen classes by transferring knowledge from related concepts.

### Comparison of different pre-trained models

Different pre-trained models yield varying results and are suitable for different applications. VGGNet is favored for its simple architecture and strong performance with little data, making it well-suited for medical imaging. However, due to its large number of parameters, training can be slow. AlexNet, one of the pioneering deep learning models, offers fast training but is less accurate than more recent models. It is mainly used for basic image classification and rapid testing.

ResNet introduced skip connections to address the vanishing gradient problem, making it highly effective for deep learning tasks such as object detection and large-scale image classification. Despite its advantages, ResNet models are computationally expensive and require large datasets to achieve optimal performance. InceptionNet enhances classification and object detection through multi-scale feature extraction facilitated by inception layers. However, its complex architecture results in longer training times.

For applications requiring lightweight and real-time characteristics, MobileNet is designed for deployment in embedded and mobile applications, making it a strong candidate for edge AI. While it is computationally efficient, its accuracy tends to be lower on large datasets. EfficientNet was developed to optimize the balance between accuracy and the number of parameters through improved scaling. It can achieve high accuracy with fewer computational resources, though fine-tuning can sometimes be challenging.

To determine the efficiency of these models, the following metrics are used.

$$Recall = TP / (TP + FN)$$

$$Precision = TP / (TP + FP)$$

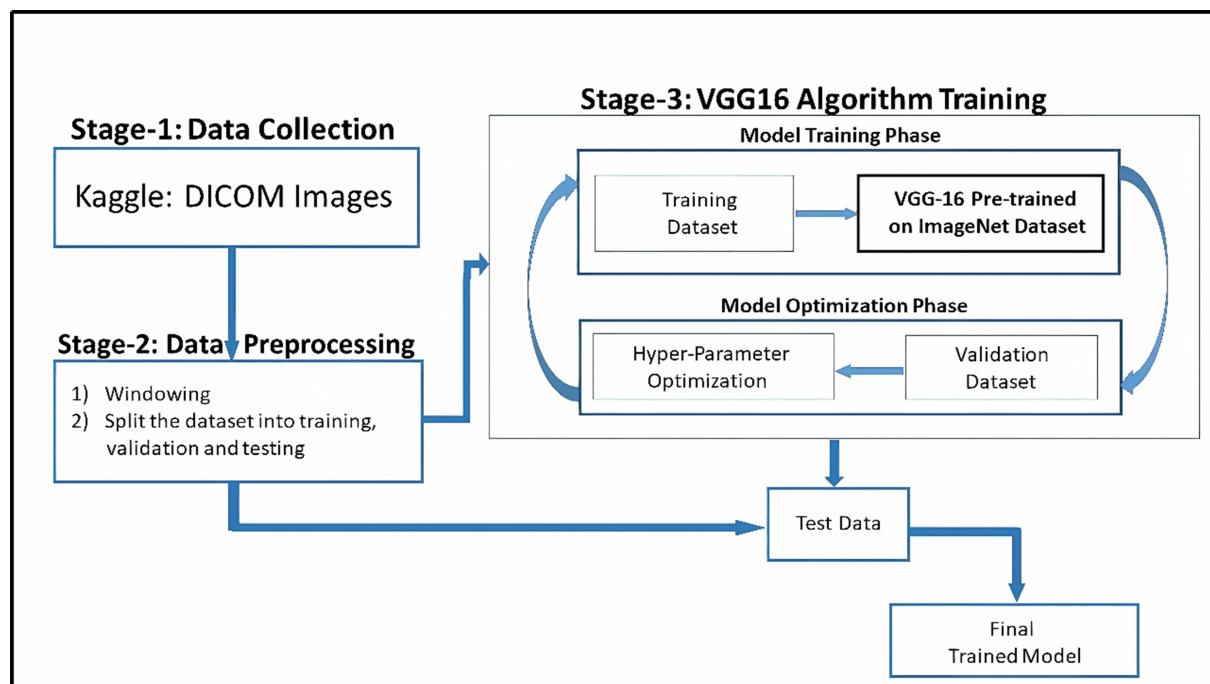
$$F1 = 2 * (Precision + Recall) / (Precision + Recall)$$

$$Accuracy = (TP + TN) / (TP + FP + FN + TN)$$

where TP (true positive), FN (false negative), TN (true negative), and FP (false positive) represent classification outcomes.

## IMPLEMENTATION

In this section, we describe the implementation of VGG-16 for hemorrhage detection. Our experiment utilizes TensorFlow, Keras, and Scikit-learn as the primary libraries.



**Figure 6.** Proposed framework.

TensorFlow<sup>[29]</sup> is a widely used open-source library developed by Google for deep learning and machine learning tasks. It provides a comprehensive suite of tools and features required for building and training deep neural networks. Keras<sup>[30]</sup>, a high-level API built on top of TensorFlow, has gained significant popularity among researchers and practitioners due to its user-friendly design. It provides both sequential and functional APIs, allowing users to build and train deep neural networks easily.

To conduct our experiment, we propose the high-level framework illustrated in Figure 6, which comprises four main stages: Data Collection, Data Preprocessing, Algorithm Training (VGG-16), and Testing with the Deployed Model. The first stage involves collecting a dataset of images relevant to hemorrhage detection. The second stage involves cleaning, formatting, and transforming the collected dataset to prepare it for training; this process may include techniques such as image augmentation and normalization.

In the algorithm training phase, the VGG-16 architecture is used to train a deep neural network on the preprocessed dataset. Hyperparameters are fine-tuned to optimize the model's performance as possible. The last stage is testing, where the trained model is evaluated on a separate test dataset to assess its performance on unseen data.

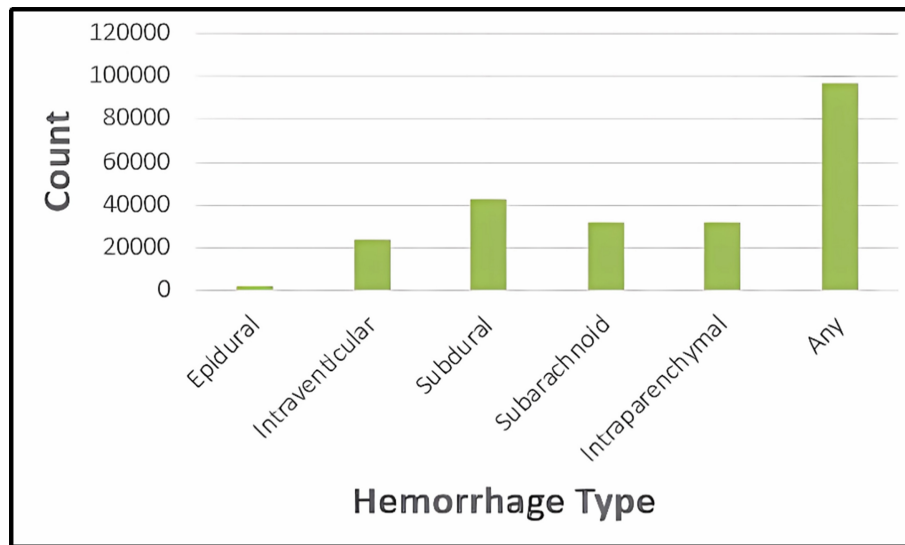
### Stage 1: data collection

The dataset used in this study is from the Radiological Society of North America (RSNA), available in digital imaging and communications in medicine (DICOM) format for download on Kaggle<sup>[31]</sup>. DICOM is one of the most popular formats for storing medical imaging data. Each DICOM file contains both metadata and image data, making it highly valuable for medical imaging analysis.

The RSNA dataset includes images labeled according to six types of hemorrhages: epidural (EDH), intraparenchymal (IPH), intraventricular (IVH), subarachnoid (SAH), and subdural (SDH). This detailed

**Table 1. Dataset composition**

Type	Count
Normal	3,814,760
Hemorrhage	230,812

**Figure 7.** Hemorrhage type distribution.

labeling provides a comprehensive view of the various hemorrhage types within the dataset.

**Figure 7** presents the distribution of images across different hemorrhage types, offering insight into the dataset's composition. Understanding this distribution is essential to ensure that the model is trained on a diverse and representative set of images. **Table 1** summarizes the dataset composition.

**Figure 8** shows the count of normal and hemorrhage images for each type. **Figures 9-13** provide sample images representing each hemorrhage type.

### Stage 2: data preprocessing

To prepare the dataset for model training, one-hot encoding is performed to transform categorical labels into a numerical format. Below are sample images of each hemorrhage type.

### Stage 3: VGG-16 algorithm training

For model training, we use the VGG-16 CNN from the Keras library. **Figure 14** illustrates the network, and **Table 2** details its parameters.

### Transfer learning for intracranial hemorrhage detection

Transfer learning enables a pre-trained VGGNet model to be adapted for intracranial hemorrhage detection, significantly reducing the amount of training required while improving performance. Initially, the model was trained on ImageNet to learn general visual representations. The lower convolutional layers, which extract fundamental features, are retained, while the upper layers are fine-tuned using the RSNA Intracranial Hemorrhage dataset to recognize hemorrhage patterns. In this work, the final layer of VGGNet - originally designed for classifying 1,000 ImageNet classes - is replaced with a new fully connected layer

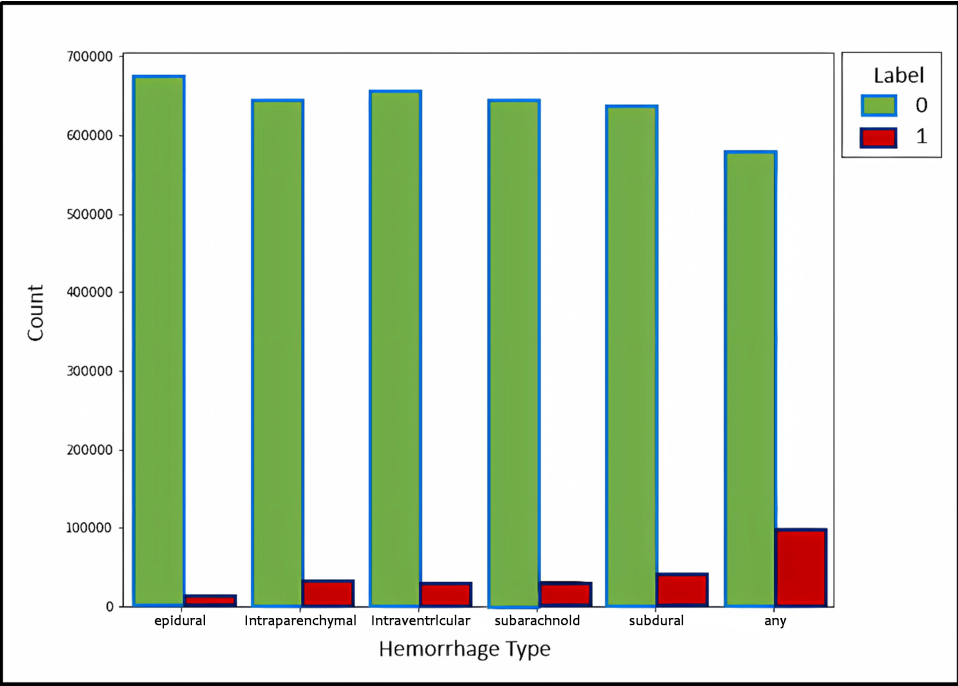


Figure 8. Examples from the dataset.

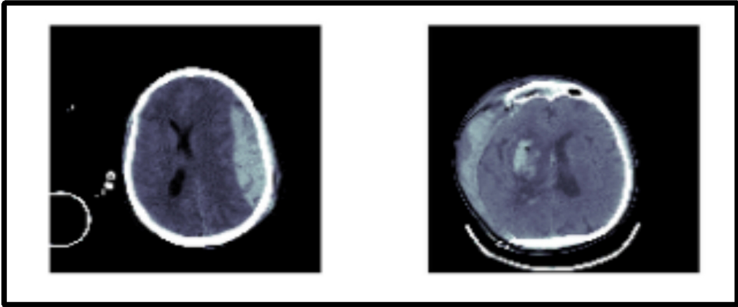


Figure 9. Epidural<sup>[31]</sup>.

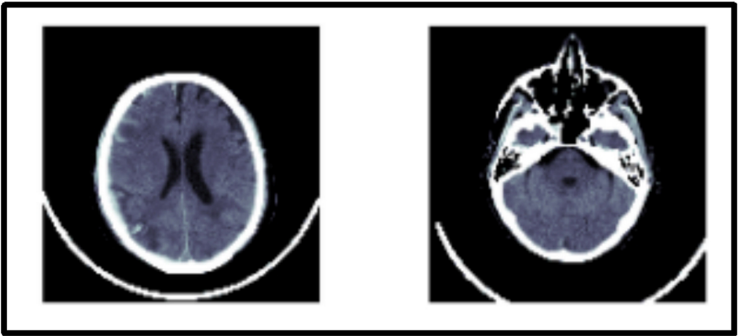


Figure 10. Intraparenchymal<sup>[31]</sup>.

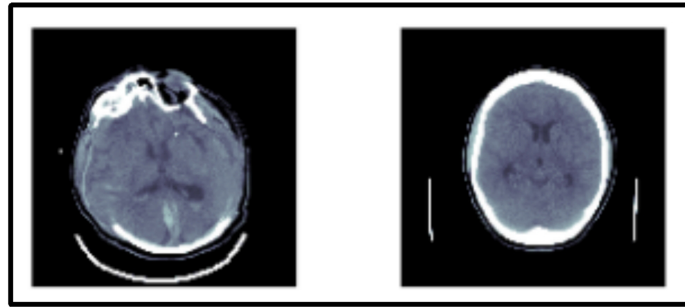
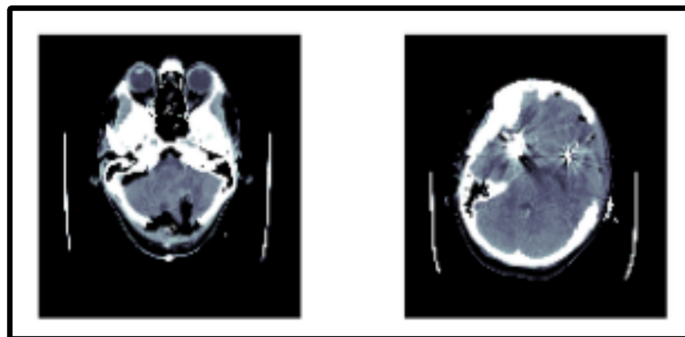
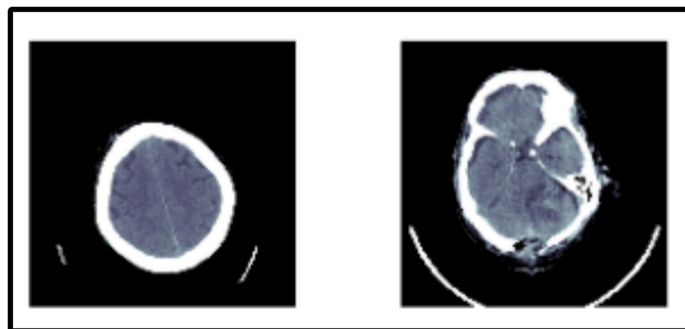
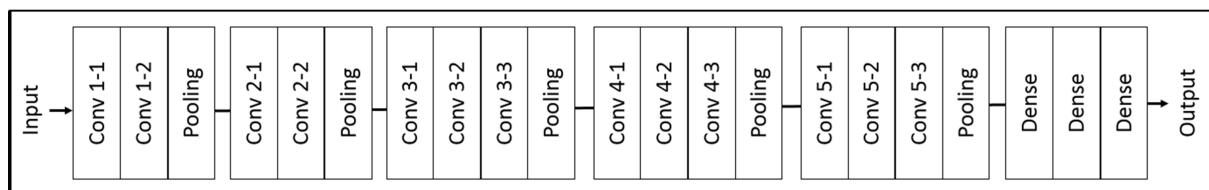
Figure 11. Intraventricular<sup>[31]</sup>.Figure 12. Subarachnoid<sup>[31]</sup>.Figure 13. Epidural subdural<sup>[31]</sup>.

Figure 14. VGG-16 architecture.

followed by a softmax function to classify hemorrhages into six categories. The entire network is fine-tuned using a small learning rate to preserve pre-trained knowledge while adapting to medical imaging data. Additionally, data augmentation methods such as rotation and contrast enhancement are applied to

**Table 2. Network parameters**

Layer	Type	Number of parameters
Conv1_1 & Conv1_2	Convolutional (3 × 3)	64 K
Conv2_1 & Conv2_2	Convolutional (3 × 3)	128 K
Conv3_1 to Conv3_3	Convolutional (3 × 3)	256 K
Conv4_1 to Conv4_3	Convolutional (3 × 3)	512 K
Conv5_1 to Conv5_3	Convolutional (3 × 3)	512 K
FC1	Fully connected	4,096
FC2	Fully connected	4,096

**Table 3. Evaluation metrics**

Accuracy	Precision	Recall	F1-Score
0.865	0.859	0.862	0.867

enhance model robustness. The Adam optimizer is employed for optimization.

This approach achieves high detection accuracy with a limited dataset, making the model efficient and suitable for real-world clinical applications.

[Table 3](#) presents the evaluation metrics. The ReLu activation function is applied to all convolution layers for faster and optimal training. Furthermore, to reduce overfitting, a dropout regularization strategy is implemented. The model achieved an accuracy of 0.865. [Figure 15](#) illustrates examples of misclassifications. [Table 4](#) provides details on the corresponding ground truth and predicted labels.

## CONCLUSION

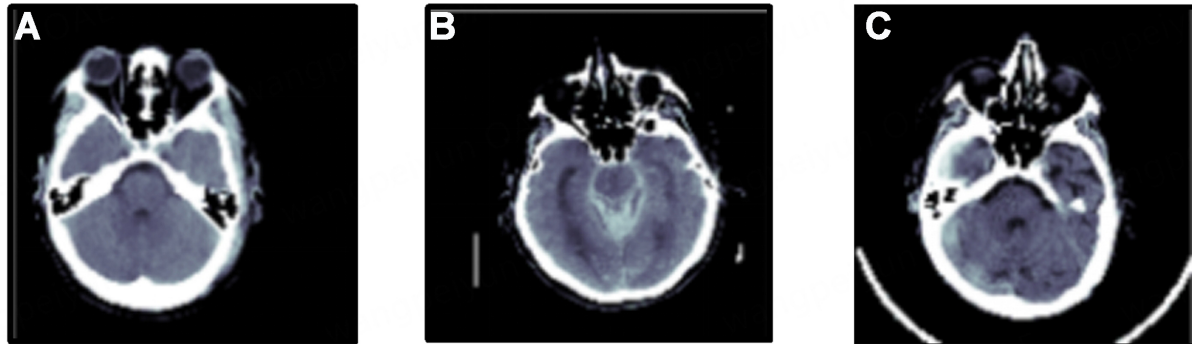
Deep learning techniques require a massive amount of data and substantial computational power to identify common patterns in medical imaging. While medical data availability is not a significant concern nowadays, a major challenge remains in accurately labeling these images. In this study, we reviewed convolution neural networks and explored various transfer learning-based algorithms. We also implemented a high-level approach for hemorrhage detection using a pre-trained VGG-16 network. Despite its effectiveness, our approach has certain drawbacks. The model was trained on RSNA data, which, although extensive, may not fully represent diverse clinical settings due to variations in scanner specifications and imaging protocols. Additionally, preprocessing techniques such as windowing and contrast adjustments can impact the model's generalizability across different healthcare environments. Another limitation is the computational cost of fine-tuning deep learning models, which may hinder real-time clinical deployment. Future work should focus on optimizing computational efficiency, ensuring compatibility with low-resource healthcare settings, and validating model performance across multiple hospital datasets to enhance generalizability. While our model demonstrated strong performance, it showed weaknesses in handling low-contrast CT images, leading to occasional misclassifications. To address this, we plan to incorporate advanced contrast enhancement techniques in future iterations of the model. Additionally, attention mechanisms, such as vision transformers or related methods, will be explored to improve feature extraction and hemorrhage localization.



**Table 4. Comparison of ground truth vs. predicted labels for misclassified cases**

CT scan example	Ground truth	Predicted label
15 (a)	Epidural	Intraventricular
15 (b)	Subarachnoid	Epidural
15 (c)	Subdural	Intraventricular

CT: Computed tomography.

**Figure 15.** Examples of misclassifications<sup>[31]</sup>.

## DECLARATIONS

### Authors' contributions

The author contributed solely to the manuscript.

### Availability of data and materials

Not applicable.

### Financial support and sponsorship

None.

### Conflicts of interest

The author declared that there are no conflicts of interest.

### Ethical approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Copyright

© The Author(s) 2025.

## REFERENCES

1. Fischbein NJ, Wijman CAC. Nontraumatic intracranial hemorrhage. *Neuroimaging Clin N Am.* 2010;20:469-92. DOI [PubMed](#)
2. Hemphill JC 3rd, Greenberg SM, Anderson CS, et al; American Heart Association Stroke Council, Council on Cardiovascular and Stroke Nursing, Council on Clinical Cardiology. Guidelines for the management of spontaneous intracerebral hemorrhage: a guideline for healthcare professionals from the American Heart Association/American Stroke Association. *Stroke.* 2015;46:2032-60. DOI [PubMed](#)

3. Nishijima DK, Offerman SR, Ballard DW, et al; Clinical Research in Emergency Services and Treatment (CREST) Network. Risk of traumatic intracranial hemorrhage in patients with head injury and preinjury warfarin or clopidogrel use. *Acad Emerg Med*. 2013;20:140-5. DOI PubMed PMC
4. Purrucker JC, Haas K, Rizos T, et al. Early clinical and radiological course, management, and outcome of intracerebral hemorrhage related to new oral anticoagulants. *JAMA Neurol*. 2016;73:169-77. DOI PubMed
5. Sacco S, Marini C, Toni D, Olivieri L, Carolei A. Incidence and 10-year survival of intracerebral hemorrhage in a population-based registry. *Stroke*. 2009;40:394-9. DOI PubMed
6. Zahuranec DB, Gonzales NR, Brown DL, et al. Presentation of intracerebral haemorrhage in a community. *J Neurol Neurosurg Psychiatry*. 2005;77:340-4. DOI PubMed PMC
7. Chan T. Computer aided detection of small acute intracranial hemorrhage on computer tomography of brain. *Comput Med Imaging Graph*. 2007;31:285-98. DOI PubMed
8. Soni J, Gangwani P, Sirigineedi S, et al. Deep learning approach for detection of fraudulent credit card transactions. In: Bhardwaj T, Upadhyay H, Sharma TK, Fernandes SL, editors. *Artificial intelligence in cyber security: theories and applications*. Cham: Springer International Publishing; 2023. pp. 125-38. DOI
9. Soni J, Sirigineedi S, Vutukuru KS, Sirigineedi SSC, Prabakar N, Upadhyay H. Learning-based model for phishing attack detection. In: Bhardwaj T, Upadhyay H, Sharma TK, Fernandes SL, editors. *Artificial intelligence in cyber security: theories and applications*. Cham: Springer International Publishing; 2023. pp. 113-24. DOI
10. Joshi S, Upadhyay H, Lagos L, Akkipeddi NS, Guerra V. Machine learning approach for malware detection using random forest classifier on process list data structure. *Proceedings of the 2nd International Conference on Information System and Data Mining*. 2018. pp. 98-102. DOI
11. Soni J, Prabakar N, Upadhyay H. Deep learning approach to detect malicious attacks at system level: poster. *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 2019. pp. 314-5. DOI
12. Hussain S, Anwar SM, Majid M. Segmentation of glioma tumors in brain using deep convolutional neural network. *Neurocomputing*. 2018;282:248-61. DOI
13. Arslan M, Bush IJ, Abiyev RH. Head movement mouse control using convolutional neural network for people with disabilities. 13th International Conference on Theory and Application of Fuzzy Systems and Soft Computing - ICAFS-2018. Cham: Springer International Publishing. 2019. pp. 239-48. DOI
14. Zhang G, Chen K, Xu S, et al. Lesion synthesis to improve intracranial hemorrhage detection and classification for CT images. *Comput Med Imaging Graph*. 2021;90:101929. DOI
15. Sengupta J, Alzbutas R, Falkowski-Gilski P, Falkowska-Gilska B. Intracranial hemorrhage detection in 3D computed tomography images using a bi-directional long short-term memory network-based modified genetic algorithm. *Front Neurosci*. 2023;17:1200630. DOI PubMed PMC
16. Nguyen NT, Tran DQ, Nguyen NT, Nguyen HQ. A CNN-LSTM architecture for detection of intracranial hemorrhage on CT scans. *arXiv* 2020; arXiv:2005.10992. Available from: <https://doi.org/10.48550/arXiv.2005.10992>. [accessed 9 Apr 2025]
17. Ngo DT, Nguyen TT, Nguyen HT, Nguyen DB, Nguyen HQ, Pham HH. Slice-level detection of intracranial hemorrhage on CT using deep descriptors of adjacent slices. *arXiv* 2022; arXiv:2208.03403. Available from: <https://doi.org/10.48550/arXiv.2208.03403>. [accessed 9 Apr 2025]
18. Helwan A, El-Fakhri G, Sasani H, Uzun Ozsahin D. Deep networks in identifying CT brain hemorrhage. *J Intell Fuzzy Syst*. 2018;35:2215-28. Available from: <https://dl.acm.org/doi/abs/10.3233/JIFS-172261>. [Accessed 9 Apr 2025]
19. Mahajan R, Mahajan PM. Survey on diagnosis of brain hemorrhage by using artificial neural network. *Int J Sci Res Eng Technol*. 2016;5:378-81. Available from: <https://www.scribd.com/document/322159299/Survey-On-Diagnosis-Of-Brain-Hemorrhage-By-Using-Artificial-Neural-Network>. [accessed 9 Apr 2025]
20. Gong T, Liu R, Tan CL, et al. Classification of CT brain images of head trauma. *Proceedings of the IAPR International Workshop on Pattern Recognition in Bioinformatics*, Melbourne, Australia. Springer; 2007. pp. 401-8. DOI
21. Soni J, Prabakar N, Upadhyay H. Comparative analysis of LSTM sequence-sequence and auto encoder for real-time anomaly detection using system call sequences. *Int J Innov Res Comput Commun Eng*. 2019;7:4225-30. Available from: [https://www.academia.edu/41696249/Comparative\\_Analysis\\_of\\_LSTM\\_Sequence\\_Sequence\\_and\\_Auto\\_Encoder\\_for\\_real\\_time\\_anomaly\\_detection\\_using\\_system\\_call\\_sequences](https://www.academia.edu/41696249/Comparative_Analysis_of_LSTM_Sequence_Sequence_and_Auto_Encoder_for_real_time_anomaly_detection_using_system_call_sequences). [accessed 9 Apr 2025]
22. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *arXiv* 2014; arXiv:1409.0575. Available from: <https://doi.org/10.48550/arXiv.1409.0575>. [accessed 9 Apr 2025]
23. Alom MZ, Taha TM, Yakopcic C, et al. The history began from alexnet: a comprehensive survey on deep learning approaches. *arXiv* 2018; arXiv:1803.01164. Available from: <https://doi.org/10.48550/arXiv.1803.01164>. [accessed 9 Apr 2025]
24. Wang L, Guo S, Huang W, Qiao Y. Places205-VGGNet models for scene recognition. *arXiv* 2015; arXiv:1508.01667. Available from: <https://arxiv.org/abs/1508.01667>. [accessed 9 Apr 2025]
25. Targ S, Almeida D, Lyman K. Resnet in resnet: generalizing residual architectures. *arXiv* 2016; arXiv:1603.08029. Available from: <https://doi.org/10.48550/arXiv.1603.08029>. [accessed 9 Apr 2025]
26. Das D, Santosh KC, Pal U. Truncated inception net: COVID-19 outbreak screening using chest X-rays. *Phys Eng Sci Med*. 2020;43:915-25. DOI
27. Marques G, Agarwal D, de la Torre Diez I. Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural

- network. *Appl Soft Comput*. 2020;96:106691. DOI PubMed PMC
28. Chen HY, Su CY. An enhanced hybrid MobileNet. 2018 9th International Conference on Awareness Science and Technology (iCAST); Fukuoka, Japan. IEEE, 2018. pp. 308-12. DOI
29. Dillon JV, Langmore I, Tran D, et al. Tensorflow distributions. *arXiv* 2017; arXiv:1711.10604. Available from: <https://doi.org/10.48550/arXiv.1711.10604>. [accessed 9 Apr 2025]
30. Ketkar N. Introduction to keras. In: Deep learning with Python. Berkeley: Apress. 2017. pp. 97-111. DOI
31. Flanders AE, Prevedello LM, Shih G, et al; RSNA-ASNR 2019 Brain Hemorrhage CT Annotators. Construction of a machine learning dataset through collaboration: the RSNA 2019 brain CT hemorrhage challenge. *Radiol Artif Intell*. 2020;2:e190211. DOI PubMed PMC