Journal of High School Science

Automated classification and segmentation of Intracranial Hemorrhages using 2D Convolutional Neural Networks and U-Net architecture on Computed Tomography scans

Goteti S

Submitted: September 17, 2024, Revised: version 1, November 27, 2024
Accepted: December 2, 2024

Abstract

Intracranial hemorrhage (ICH) is a type of stroke and a life-threatening condition that requires rapid and accurate diagnosis. This research utilized advanced Convolutional Neural Networks (CNNs) and a U-Net architecture to automate the classification and segmentation of ICH in Computed Tomography (CT) scans to improve the accuracy and efficiency of the diagnosis of ICH. Three primary models were constructed, 1) a 2D binary CNN model for classifying individual slices of CT scans as being hemorrhagic or non-hemorrhagic, 2) a 2D multi-label CNN model for diagnosing the type of hemorrhage for only hemorrhagic scans, and 3) a 2D U-Net model for segmenting hemorrhagic regions within the individual CT slices already classified as hemorrhagic from the 2D binary classification model. The results showed that the binary classification model, the multi-label classification model and the U-Net segmentation model achieved test accuracies of 97.7%, 81.3% and 98.7% respectively. These findings highlight the potential of CNNs and image segmentation to assist radiologists in faster and more accurate ICH detection and diagnosis.

Keywords

Intracranial Hemorrhage, ICH, Classification, Segmentation, U-Net model, Convolutional Neural Networks, Computed Tomography, CT scan, Data augmentation, Loss function

Shruti Goteti, International School of Zug and Luzern, Rothusstrasse 4b 6331 Hünenberg, Switzerland.
shruti.goteti@gmail.com

Introduction

Intracranial hemorrhage is a type of stroke that refers to bleeding within the skull. The bleeding can occur in different locations, including the brain tissue itself or in spaces surrounding the brain. When bleeding occurs within the skull, it can rapidly increase intracranial pressure and reduce blood flow to brain cells and tissues (1). Timely diagnosis is especially crucial because nearly half of the mortality rate associated with ICH occurs within the first 24 hours (2). Hence, an early diagnosis is essential for making critical treatment decisions. These include treating the hemorrhage with surgical intervention or medical management, the rapidity of treatment onset significantly influencing patient outcomes (3).

Intracranial hemorrhages are further classified into 4 subtypes, Epidural hemorrhage, Subdural hemorrhage, Subarachnoid hemorrhage, and Intracerebral Hemorrhage. The first three types of hemorrhages refer to the bleeding within the skull but outside the brain tissue, whereas Intracerebral hemorrhage includes bleeding outside the brain tissue as well (1). Additionally there are 2 types of Intracerebral hemorrhages, Intraventricular and Intraparenchymal (Figure 1).

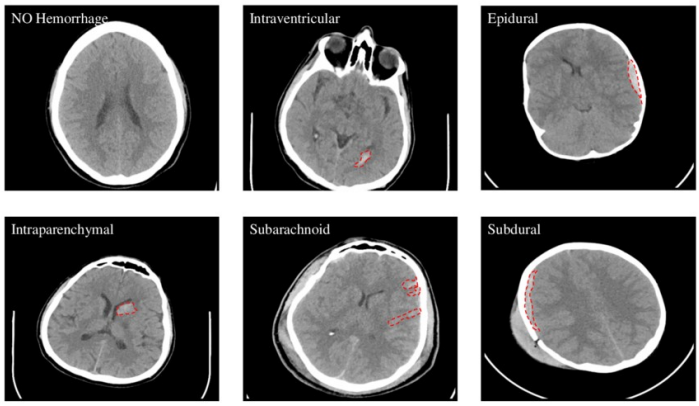

Figure 1. This figure shows different types of ICH, with the hemorrhagic segments for each subtype outlined and compared to a non-hemorrhagic CT scan. As shown, hemorrhages are quite small and can often be either misdiagnosed or diagnosed improperly during manual observations (8)

Epidural hemorrhages have a mortality rate of 5-15%. Subdural hemorrhages present with high mortality rates of 50-90% (4). Subarachnoid hemorrhages have a mortality rate of up to 65% (5). Intraventricular hemorrhages have a mortality rate spanning from 10% to 50% depending on the degree of IVH (2nd and 3rd degree) (6) and lastly, Intraparenchymal hemorrhages have a long-term mortality rate that can reach as high as 81% (7). Factoring all of these high rates into consideration, it is clear that there is an urgent need for rapid and accurate diagnosis. Conventionally, this process has relied on manual review of CT scans by radiologists, which is time-consuming and subjective. However, new breakthroughs in machine learning, specifically deep learning, offer

promising results for automating medical image analysis.

Primary objectives of this study

The primary objectives of this study were to 1) design 2D CNN models for classification of individual CT slices as being hemorrhagic or non-hemorrhagic while also classifying the subtype of hemorrhagic slices, and 2) construct a 2D CNN model that could segment hemorrhagic regions in CT slices. The findings of this study can aid in clinical decision-making by providing radiologists with automated tools for quicker and more reliable diagnosis, especially after the type and location of the hemorrhage is known. Figure 2 below shows a model flowchart of this study.

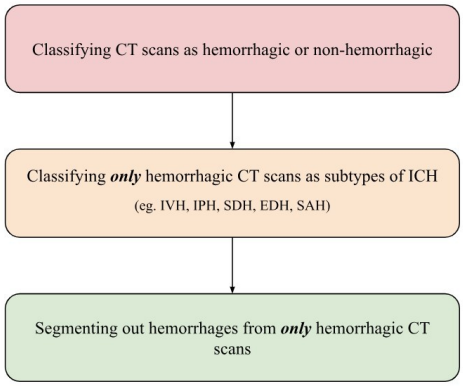

Figure 2. This figure shows the model flowchart for this study, first performing classification of the CT scans, and then segmenting out hemorrhagic areas as shown in Figure 1.

Background

Convolutional Neural Networks and U-Net models have both revolutionized medical image analysis and classification by automating the learning of complex features from raw image data. CNNs and U-Net models have shown remarkable success in various medical imaging tasks, such as tumor detection, organ segmentation, and disease classification. Their ability to learn from large datasets has made both of these models the preferred choice for developing diagnostic models in healthcare. This section will describe these models in detail and present examples of papers implementing these models in the medical field.

The U-Net architecture for image segmentation

The U-net is very recent, introduced by Ronneberger et al. in 2015 at the University of Freiburg, Germany (9). This architecture was designed for biomedical image segmentation and has gained significant popularity due to its robust nature. The model is specifically designed to run on image data. The name of the model is derived from its U-shaped architecture, which consists of a downsampling path followed by an upsampling path. The Encoder (downsampling path) extracts specific important features of the image and reduces its resolution, while the Decoder (upsampling path) restores the original image and creates a

segmentation. These two paths are essential for the U-net to learn the features of the image and successfully segment out a particular feature; in this case; a hemorrhage (10).

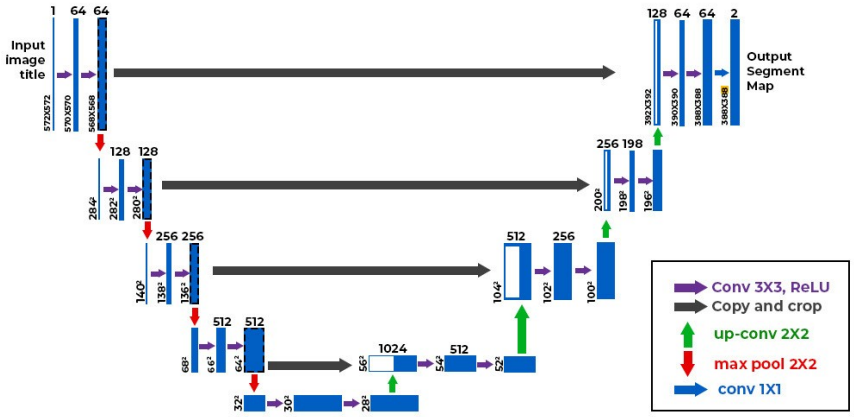

Figure 3. Generalized U-Net architecture design (11)

Since 2015, many research papers have used the U-net architecture for image segmentation, especially for hemorrhages and strokes. For example, Hua et al. proposed the MSRL-Net model, an enhanced form of the U-Net architecture with MaxPool and Softpool for improved small lesion segmentation (12). Chang et al. also enhanced segmentation, integrating attention mechanisms to create an All-Attention U-Net (13), while Ahmed et al. introduced a CSR-based Spatial Attention U-Net specifically for subdural and epidural hemorrhages (14). Piao et al. implemented a TransHarDNet, combining HarDNet blocks and transformers to enhance accuracy and speed in intracranial hemorrhage segmentation (15). These adaptations and hybrid architectures increased the U-Net model's versatility and impact in advancing medical imaging.

Paper 1: Modified U-Net architecture for segmenting acute ischemic strokes

One specific example is a report on Deep - Learning-enabled detection of acute ischemic stroke using brain Computed Tomography images on 3D CT images (16). This paper aimed to enhance the segmentation of brain regions affected by ischemic stroke using CT images by modifying the classic 3D architecture to improve stroke lesion segmentation. However, the model still consisted of an encoder, which captured important features, and a decoder, which localized important details such as stroke regions. This study made several unique contributions to the medical image segmentation field, despite the limitation of a small dataset. The modified U-Net architecture with dilated convolutions and instance normalization significantly improved the performance of the model.

Paper 2: Enhanced U-Net architecture for segmenting lung diseases

This paper (17) focused on applying the U-Net model to lung CT slices for segmentation. Additionally, the study explored variations in

the U-Net architecture to address challenges in lung disease detection, such as processing irregularly shaped or faint regions of interest in CT scans. It introduced attention mechanisms to refine feature maps and focus on important regions of the image and used specific segmentation metrics to effectively evaluate the model. This research highlighted the adaptability of U-Net for various medical imaging applications and demonstrated its effectiveness in lung segmentation tasks.

Similarities between models

Similarities between these two papers and the model in this study include the use of an encoder for feature extraction and a decoder for spatial reconstruction. All models, despite the primary focus, were designed for segmentation tasks in medical imaging, identifying the regions of interest in CT scans. All three papers also used an Adam optimizer for efficient parameter updates and convergence, and incorporated dice loss (or combined dice and binary-cross entropy) to address imbalanced datasets and ensure accurate overlap between prediction and ground truth. The image segmentation dataset for this study is different from the classification dataset, and grapples with the challenges of a small dataset. The reason for this will be stated in the *data collection* section. This study shared these challenges with the first paper, which also was limited to a total of 103 scans. This challenge however, was solved through data augmentation techniques such as rotation, flipping, and zooming to increase dataset size. This is important to note because this paper demonstrated that performing image segmentation on a small dataset still has the potential to make important contributions to medical diagnosis. All of these approaches and metrics improved the accuracy and efficiency of the model.

Differences between models

However, it is important to note the differences between the model in this study and papers 1 and 2. This model specifically focused on segmenting out ICH in 2D brain sleeves, whereas paper 1 segmented out ischemic stroke lesions in 3D CT images, and paper 2 segmented lung regions in CT images (dimensions not specified) to focus on irregularly shaped features caused by diseases. The model in this study used a basic U-Net architecture with minor adjustments (e.g., dropout, combined loss), whereas paper 1 added dilated convolutions and instance normalization to improve segmentation for small or faint lesions. Additionally, paper 2 incorporated attention mechanisms to refine segmentation and focus on the most relevant areas. Lastly, the metrics used differed slightly. This study used dice coefficient and binary cross-entropy for segmentation performance, while paper 1 specifically focused on dice and stroke-specific metrics for lesion segmentation, and paper 2 used dice, intersection over union, and lung-specific performance metrics to evaluate segmentation accuracy.

The Convolutional Neural Network for classification

CNNs were conceptualized in the 1980s, and they have since evolved to become the advanced models they are today (18). A Convolutional Neural Network is a type of machine learning model, most commonly used for image classification and processing tasks. It consists of multiple layers including convolutional layers, pooling layers, and fully connected layers (as shown in Figure 3). The

convolutional layers extract spatial features, such as edges and textures, from the input images. The pooling layers downsample the feature maps, reducing the computational complexity and retaining only the most critical information. The fully connected layers then combine these features to make the predictions.

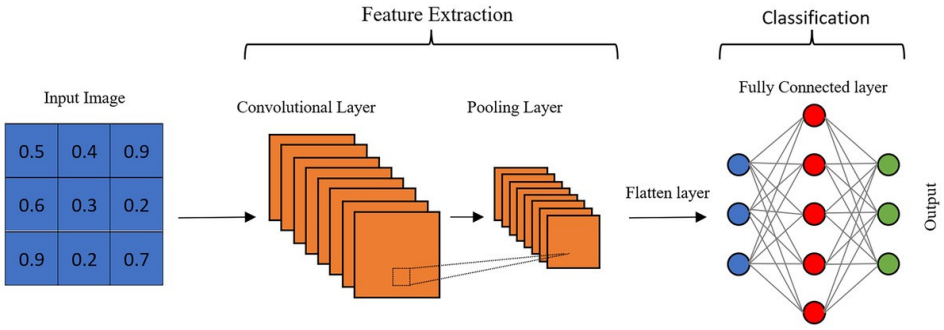

Figure 4. A basic Convolutional Neural Network model demonstrating all of the steps from the input image, all the way to the output image (19)

Due to CNNs effectiveness, many research papers have explored them for classifying and detecting hemorrhages. For example, Umapathy et al. proposed an ensemble CNN model combining SE-ResNeXT and lSTM networks to classify subtypes of intracranial hemorrhages with high precision and recall (20). Von Krummer et al. introduced a classification framework integrating imaging and clinical data for precise intracranial hemorrhage subtyping, emphasizing the clinical relevance of CNN applications in post-stroke analysis (21). Lastly, Lee et al. introduced a CNN-based method leveraging advanced data augmentation techniques to classify intracranial hemorrhages, achieving a high sensitivity (22). These advancements underscore the potential for CNNs to improve diagnostic accuracy and clinical outcomes in brain imaging.

Paper 1: Stroke classification using CNNs

Marbun et al. constructed a CNN model for the automatic classification of stroke types (hemorrhagic and ischemic) using CT scans. The unique contributions of this research paper were in its use of image preprocessing, tailored CNN architecture, and its focus on stroke type classification in CT scans. By achieving a high accuracy of 90%, this report highlighted the potential for automated stroke diagnosis in the medical field. (23)

Paper 2: Deep learning for acute ischemic stroke detection

Babutain et al. (24) attempted to detect acute ischemic strokes using a multi-stage approach by combining segmentation and classification. They employed a ResNet50 for feature extraction and segmentation and a custom 5-Scale CNN for classification.

Similarities between models

All models aimed to classify using CT scans, with varying actual medical conditions. Normalization was consistent across all three models, as augmentation techniques were used to increase dataset size and strengthen the

generalization of the models. Additionally, all models utilized the Adam optimizer for efficient weight updates and calculated evaluation metrics—precision, recall, F1 score, and accuracy—ensuring robust performance assessment. Lastly, both Paper 2 and this study combined image segmentation and classification, creating a more advanced diagnostic system.

Differences between models

The dataset used in this study consisted of > 4000 CT scans, a significantly larger dataset that ensured generalization. Paper 1 utilized only 45 images for training and testing, which limited model robustness and required careful handling of overfitting. Paper 2 incorporated a medium-sized dataset (approximately 300 acute strokes and 300 normal images for classification) but still contained a significantly lesser number of images, than the number used for this study. Additionally, preprocessing techniques and CNN model architectures varied. This study performed simple normalization and augmentation and used a multi-layer CNN along with a custom multi-label architecture for classifying subtypes of ICH. Paper 1 used advanced techniques like CLAHE (Contrast Limited Adaptive Histogram Equalization) to enhance image contrast and employed a basic CNN with a single hidden layer and limited scalability. Paper 2 conducted sophisticated preprocessing for contrast enhancement and used a complex multi-pipeline combining segmentation (ResNet50) and a custom classification model (5S-CNN). Lastly, the metrics that were used to score the models' accuracies varied: this study incorporated a wide array of metrics such as ROC-AUC curves, confusion matrices, and subtype-specific evaluations to gain insight spanning from true positives to false negatives, whereas Paper 1 reported only accuracy, and Paper 2 provided metrics such as cross-validation performance.

Differentiation with prior literature

This study was unique because, while many studies focused solely on classification or image segmentation independently, this study combined the two. By first classifying a scan as hemorrhagic or non-hemorrhagic and then performing image segmentation on the hemorrhagic scans, the models provided a valuable solution for ICH detection and localization. This approach made the model highly efficient and scalable. Additionally, this model also classified the types of hemorrhages, an attribute that ranks high in importance for radiologists' and neurosurgeons' interpretations of CT scans.

Methods

Dataset and collection

Two datasets were used for this study. The first dataset was sourced from data made publicly available by Hssayeni et al. (25). However, the CT scans had originally been taken from patients at the Al Hilla Teaching Hospital in Iraq. The Al Hilla dataset had been compiled to detect and segment ICH from CT scans of patients. The Kaggle dataset provided head CT images in JPG format, consisting of 2500 brain window images and 2500 bone window images, covering 82 patients. For this study, however, only the brain window scans were used. The dataset also included intracranial image masks for 318 of the images: each patient had a folder containing approximately 30 slices of 5mm thickness of their CT scans, and a few of the scans included a file ending in **_HGE_Seg.jpg**, which was the hemorrhage

segmented from the corresponding full CT scan. For example, patient 49 had a file called **27.jpg** and a corresponding file **27_HGE_Seg.jpg**, which showed the hemorrhage segmented from **27.jpg**. Alongside the hemorrhage masks, the dataset provided csv files containing hemorrhage diagnosis data and patient information, along with the type of hemorrhage for each patient classified as hemorrhagic. The number of CT scans for each subtype included 5 patients classified as IVHs, 16 as IPHs, 7 as SAHs, 21 as EDHs, and 4 as SDHs. For this study, only the type of hemorrhage was extracted from the csv file **patient_demographics.csv** and nothing else, as the primary focus of the study was on classification and segmentation, not on the patients demographics.

The second dataset was the CQ500 dataset, sourced from Qure AI (26). The dataset scans were collected from several radiology centers in New Delhi, India. The annotations for each ICH subtype were manually carved by three senior radiologists with 10 years of experience in CT interpretation. The CT scanners included GE BrightSpeed, GE Discovery CT750 HD, GE LightSpeed, GE Optima CT660, Philips MX 16-slice, and Philips Access-32 CT (27). The dataset consisted of 491 CT scans in dicom format, including 205 classified as ICHs, 40 fractures, 65 middle shifts, 127 mass effects, and 54 normal controls. A csv file containing hemorrhagic subtypes showed that the 205 ICH scans included 28 IVHs, 134 IPHs, 60 SAHs, 13 EDHs, and 53 SDHs. For this paper, only the 205 ICH scans were used. The dataset also contained CT scans of different thicknesses, including 0.625 mm, 2.55 mm, 3.00 mm, 5.00 mm, and 55 mm. However, for this study, only slices with a 5 mm thickness were used to maintain compatibility with the other dataset and to ensure control and predictability for the model.

2D Classification Model

Both datasets contained hemorrhagic CT scans that were often classified as more than one hemorrhage type, reflecting real-world diagnosis. Due to this, a labeling system was created that assigned a binary value to each subtype.

CQ500 Dataset

For this model, an integration between the two datasets was conducted. First, the program downloaded the ZIP files comprising the CQ500 dataset. The ZIP files were inspected to ensure their validity and to contain subfolders with only "5 mm" in their names, ensuring that only folders with 5mm thick slices were processed. Then, each dicom file in the extracted folders was read using the **pydicom** library, which provided access to the pixel array containing the grayscale CT scans. The pixel array values were then normalized to the range [0, 255]. This normalization ensured consistency across images. Subsequently, the images were resized to 256 x 256 pixels, which was crucial for ensuring uniform input dimensions for the CNN model. Finally, the resized images were saved as jpg files using **cv2.imwrite**.

After the images were converted to .jpg files, label assignment classified each CT scan as being either hemorrhagic or not. The csv file indicated whether a hemorrhage had been classified as one or more of the subtypes, with columns R1:ICH, R1:IPH, R1:IVH, R1:SDH, R1:EDH, and R1:SAH containing a 1 for "yes" or a 0 for "no" in the row associated with each

folder name. Each folder in the dataset corresponded to a scan ID (e.g., CQ500CT211). The scan ID was then matched with an entry under the "name" column in the csv file. When a match was found, the labels were extracted from the corresponding row. First, the program checked the column "R1:ICH" to determine whether the CT scan was classified as being hemorrhagic. If affermative, the rest of the labels for the subtypes were extracted from the csv file. This was important because later, two CNN models were used: one to classify CT scans as hemorrhagic or non-hemorrhagic, and another to classify the subtype of hemorrhagic CT scans.

The extraction of subtypes was not limited to one subtype per scan. Each hemorrhagic patient fit into more than one subtype, reflecting real-world diagnosis. To address this challenge, a labeling system of an array containing of six integers for each subtype was created. For example, a patient from the CQ500CT211 folder associated with the labels [1, 1, 0, 0, 0, 0], indicated that they had ICH (classified as being hemorrhagic) and IPH subtypes of hemorrhages. Processed images were stored in a list, with pixel values normalized to the range [0,1]. Neural networks perform better when the input values are small and standardized. Next, the images were reshaped to include a channel dimension, changing the shape from (256, 256) to (256, 256, 1). This step ensured compatibility with CNNs, which required input dimensions of height, width, and channels. After preprocessing, the CQ500 dataset contained 2172 labeled brain CT scans for the binary classification model, with non-hemorrhagic and hemorrhagic cases, the latter, further distributed across all subtypes.

PhysioNet dataset

The PhysioNet dataset consisted of multiple patient folders spanning from 049 to 129, each of which contained a **brain** subfolder. This subfolder included preprocessed jpg images of CT scans. The folder name corresponded to the patient number. Then, labels for the different hemorrhagic subtypes were extracted in a similar way to the CQ500 dataset, i.e. by matching the folder name (patient number) with the "Patient Number" column in the csv file **patient_demographics.csv**. Once again, the images within the folder were first classified as being hemorrhagic or non-hemorrhagic based on the presence of a 1 (yes) or 0 (no) in the "Intracranial Hemorrhage" column. If hemorrhagic, then binary labels were assigned based on the presence of subtypes. The labeling system was the same as that of the CQ500 dataset, which consisted of an array of six binary integers, each representing a subtype. Then, each image in the brain subfolder was resized to 256 x 256 pixels, and the pixel values were normalized to [0,1] in the same way as the CQ500 dataset, which improved the stability of the neural network training. Grayscale images were reshaped to (256, 256, 1). A total of 2501 images were processed into a single dataset along with their corresponding labels.

At this point, both datasets contained 1790 IPH slices, 699 IVH slices, 743 SDH slices, 322 EDH slices, 1358 SAH slices, along with 4673 hemorrhagic and non-hemorrhagic slices in total. Once again, all of these numbers when added were greater than the number of hemorrhagic CT scans, because one scan could be classified as more than one subtype. The multi-classification dataset was not balanced before being input into the classification

algorithm. Data augmentation was performed on the subtype dataset, in an attempt to create a more balanced dataset artificially (discussed later).

Combining datasets

After preprocessing, images and labels from the CQ500 and PhysioNet datasets were concatenated to form a unified dataset. For binary classification of images as hemorrhagic or non-hemorrhagic, only the first integer of each six-integer array was used, as a 1 indicated an intracranial hemorrhage, and a 0 indicated otherwise. The multi-label classification model predicting specific hemorrhage subtypes, however, was only applied to images already classified as hemorrhagic after verifying that the labels associated with the images did not contain [0, 0, 0, 0, 0, 0]. For this classification, contrasting with the binary classification model, the first index integer was disregarded, as an intracranial hemorrhage did not represent a subtype but only whether the image was hemorrhagic. After creating two separate labels, both datasets were split into training (70%), validation (15%), and test (15%) sets using the **train_test_split** function. The validation set allowed for hyperparameter tuning, while the test set ensured that the model's performance was evaluated on completely new data. A stratify split argument was not used in training. Instead, data augmentation was applied to balance the training dataset artificially (see later).

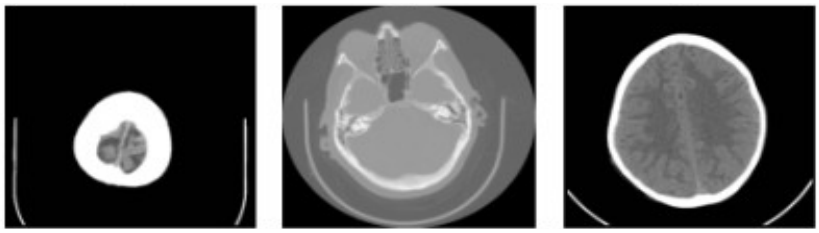

Figure 5. Example CT slices before data augmentation

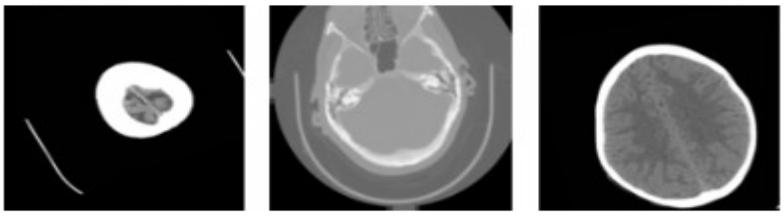

Figure 6. Same CT slices from figure 5 after data augmentation, showing rotating and zoomed in images

Then, data augmentation was applied to the training data to artificially increase its diversity and prevent overfitting. The following transformations were implemented: random rotations up to ±30^{o}, random zooms in or out by up to 20%, random translations in both horizontal and vertical directions by up to 20% of the image size, and random flipping of images along the horizontal axis. The model ensured that the dataset was balanced by

selectively augmenting samples to increase representation from the minority class (ICH=1) or randomly downsampling excess samples from the majority class (ICH=0). The same process was applied to the multi-label classification dataset simultaneously. Data augmentation balanced the different subtype classes by augmenting less frequent subtypes (e.g., epidural hemorrhage). Finally, the classes were combined into a balanced training set. This process was performed to improve the robustness of the model by teaching it to process variations in the input data. In this case, total data augmentation (including the binary and multi-label datasets) was performed on 3472 images of the training set, and after data augmentation, there were 6944 images, twice the original amount. Figure 5 shows example CT slices before data augmentation, and Figure 6 shows the same CT slices after data augmentation.

Table 1. Brief overview of data being processed for binary and multi-label classification

Aspect	Binary Classification	Multi-Label Classification
Labels	ICH = 1 (hemorrhagic), ICH = 0 (non-hemorrhagic)	IPH, IVH, SDH, EDH, SAH with 1 indicating the subtype, 0 indicating no subtype
Scope	All images in the dataset	Hemorrhagic images only
Task	Single-label (binary) classification	Multi-label classification
Data Augmentation	Applied to balance hemorrhagic/non-hemorrhagic classes	Applied to balance subtype distributions

2D Image Segmentation Model

The program used for the image segmentation model utilized only the PhysioNet dataset because the CQ500 dataset did not include segmentation masks for this mode. The program processed the data in the same way, loading each grayscale brain image from the brain folder and resizing it to a standard of 256 x 256 pixels from the original resolution of 650 x 650 pixels. However, instead of determining whether an image was hemorrhagic or not based on the csv file, a function called *classify_patient* determined if the patient had a hemorrhage based on the presence of **_HGE_Seg.jpg** files in the folder. If such files were found, the patient was labeled as 1 (hemorrhagic); otherwise, the patient was assigned a label of 0 (non-hemorrhagic).

Once all the images were loaded and resized, the pixel values were normalized by dividing them by 255.0, scaling them to a range of [0,1]. Next, a channel dimension was added to transform their shape from (256, 256) to (256, 256, 1). At this point, there were 318 input images (hemorrhagic CT scans) and 318 output images (the segmented-out hemorrhages). Subsequently, data augmentation was applied in a similar way to the 2D classification model to increase the variety of training samples by applying random transformations, including rotations, flips, and shifts to the input images and masks. The transformations that were applied to the dataset consisted of randomly rotating the image by up to 10^{o}, shifting the image horizontally or vertically by 10%, applying a shearing transformation or zooming in/out by 10%, and randomly flipping the

image horizontally. A custom data generator function then yielded batches of augmented images and their corresponding masks during training. Data augmentation was applied on 223 images, generating 223 entirely new images. Hence, the training model was composed of 446 images. This function ensured that the transformations applied to the images were mirrored in the segmentation masks, maintaining the correct correspondence between input and output. Then, the model was split into the training set (70%), a validation set (15%), and a testing set (15%).

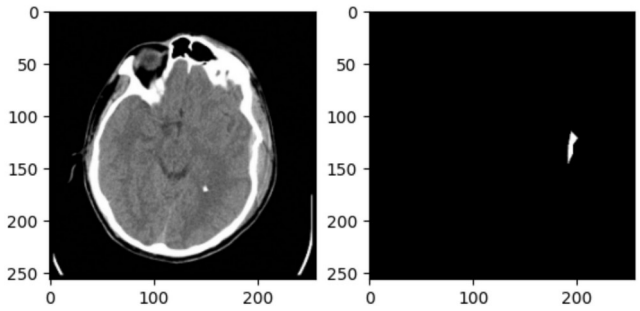

Figure 7. Example CT scan with corresponding hemorrhage segmentation from the dataset (19)

2D classification models (Binary and Multi-Label)

As previously mentioned in the Data Collection section, this model used CNNs to perform binary classification and multi-label classification to classify hemorrhagic cases and subtypes. CNNs consist of three layers: convolutional layers and filters, pooling layers, and fully connected layers.

Convolutional layer

Both models used three convolutional layers, each with the same increasing filter sizes (32, 64, 128). By using the Conv2D function to build the neural networks, the kernel size was set as (3, 3). Kernels, also called filters, are small matrices that slide across the input image, extracting important features of images such as edges, textures, and shapes. The higher the number of filters, the more complex the features the model could extract. In early layers, the filters learned simple patterns (like edges), while in later layers, they captured more abstract and detailed patterns (like shapes). However, significantly increasing the number of layers could also have led to overfitting, which was why only three layers were used in the convolutional layer.

After each convolutional layer, the ReLU (Rectified Linear Unit) activation was applied, keeping only positive values and replacing the negative values with zero. This was important because the function ensured that the network could model non-linear relationships, which captured complex patterns and relationships that simple linear functions could not. The ReLU function is expressed as ReLU(x) = max (0, x), where input x is any real number, and the function only output numbers from 0 to x (28).

Pooling layers

Both models included a max-pooling layer

following the convolutional layer. The 2D pooling layer function (**MaxPooling2D**) was used to reduce the spatial dimensions of the feature maps. In this case, the max-pooling operation took a 2x2 patch of the image for both models and selected the maximum value from that patch. For example, a feature map of (64, 64) was downsampled to (32, 32). Pooling reduced the resolution of the feature maps while retaining the most prominent features, which helped prevent overfitting.

Flattening and fully connected layers

Subsequent to the pooling layers, the **Flatten()** operation converted the multi-dimensional feature maps produced by the convolutional layers into a one-dimensional vector. This was necessary because the final classification layers (fully connected layers) expected a 1D output. For example, if the final feature map had dimensions of (32, 32, 256), it was flattened into a vector of size 32 * 32 * 256 = 262,144.

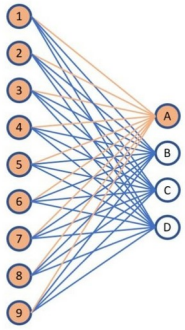

Figure 8. Fully connected layer backend diagram (29)

The fully connected layers (or dense layers) were standard neural network layers where every neuron was connected to the neurons in the previous layer, as shown in Figure 8.

In this case, the binary and multi-label classification models followed different structures. The first dense layer in the binary model included 256 neurons with ReLu activation. Once again, more neurons enabled the network to learn more complex patterns but increased the risk of overfitting. Hence, 256 neurons were used as a balance between the two. The final dense layer (output layer) contained a single neuron with a sigmoid activation function to perform binary classification. The sigmoid activation function output a probability between 0 and 1, representing the likelihood that the image belonged to the positive class (hemorrhage). If the output was > 0.5, the model classified the image as belonging to the hemorrhage class.

The first dense layer in the multi-label model also contained 256 neurons with ReLu activation for the same reason. The output layer included 5 neurons, however, with sigmoid activation, one for each subtype. The sigmoid activation function was used in the same way as it had been for the binary classification model, outputting a number between 0 and 1 for each neuron independently. For example, a prediction for an input image with the following array [0.8, 0.1, 0.6, 0.0, 0.4] would return [1, 0, 1, 0, 0], indicating that it belonged to IPH and SDH.

Model compilation and loss function

Both models were then compiled and used an optimizer called the Adam optimizer, a loss function termed the binary cross-entropy, and accuracy as a performance metric.

The Adam optimizer was used to update the weights of the model during training. Adam is an adaptive learning rate optimizer that combines the benefits of Stochastic Gradient Descent (SDG). Adam is effective because it adjusts the learning rate for each parameter based on the estimates of the first and second moments of the gradients, and therefore speeds up the training process and improves the convergence of neural networks. The loss function, Binary Cross-Entropy (BCE), is a key metric in binary classification tasks because it directly measures how well the model's predicted probabilities align with the true class labels (30).

$$BCE = \frac{-1}{N} \sum_{i=1}^{N} p_i \quad (1)$$

Where $\mathbf{y_i}$ is the true label (1 for hemorrhage, 0 for non-hemorrhage) and $\mathbf{p_i}$ represents the predicted probability output from the sigmoid function, the loss penalizes incorrect predictions, and the goal of training is to minimize this loss.

Training and evaluating the model

Both 2D classification models were trained on 10 epochs with a batch size of 32. In each epoch, the model processed small batches of 32 images, computed the gradients based on losses, and updated the weights using the Adam optimizer. This was important because batch processing improved computational efficiency and enabled frequent weight updates. After each epoch, the model was evaluated on the validation set. This provided an estimate of how well the model generalized to unseen data. If the validation accuracy remained consistent with the training accuracy, it indicated that the model was not overfitting. After training and validation, the model was evaluated on the test set. The test accuracy and loss were calculated to measure the model's performance on unseen data. The model predicted probabilities for the test images using the sigmoid activation in the final layer.

Precision, Recall, and F1 Score

Precision, recall, and the F1 score are critical metrics for evaluating the performance of machine learning models, especially in tasks where the cost of false positives and false negatives is high. These metrics not only provide insight into the accuracy of the model but also ensure that the model is clinically reliable, minimizing both false diagnoses and misclassification. This is essential for patient safety.

Precision is the ratio of true positives to all positive predictions (both true and false positives). True positives represent the number of correctly predicted hemorrhagic or subtype cases, whereas false positives indicate the number of non-hemorrhagic or incorrect subtype cases predicted as positive. Precision answers the question of how many patients predicted to have hemorrhages or specific

subtypes, actually had a hemorrhage or the subtype. Hence, the higher the precision score, the more accurate the diagnosis from the model.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (2)$$

For the binary model, a single precision score was calculated to reflect the accuracy of the model, whereas the precision for the multi-label model was calculated independently for each subtype.

Recall (or sensitivity) is the ratio of true positives to all actual positives (true positives + false negatives), where false negatives are the actual positive cases missed by the model. Recall answers the question: of all the patients who had had a hemorrhage, how many are correctly identified? And if all those had been hemorrhagic, how many subtypes are correctly identified? The closer the number is to 1, the better, as that had indicates that a low number of predicted false positives.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3)$$

Once again, one recall score was calculated for the binary model, and five recall scores were calculated for each subtype for the multi-label model.

The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance when dealing with imbalanced datasets. A low score in either metric significantly reduces the F1 score, hence the higher the F1 score, the better the performance of the model.

$$F1\,Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

A single F1 score summarized the overall performance in detecting hemorrhages for the binary model, whereas F1 scores were computed for each subtype, identifying which hemorrhage types the model performed well on and which ones required improvement.

Confusion matrices

Confusion matrices provide a detailed breakdown of model predictions, helping to identify specific mis-classifications. A confusion matrix contains the following elements; True Positives (TP) containing correct positive predictions; True Negatives (TN) containing correct negative predictions; False Positives (FP) containing incorrect positive predictions; and False Negatives (FN) containing missed positive predictions.

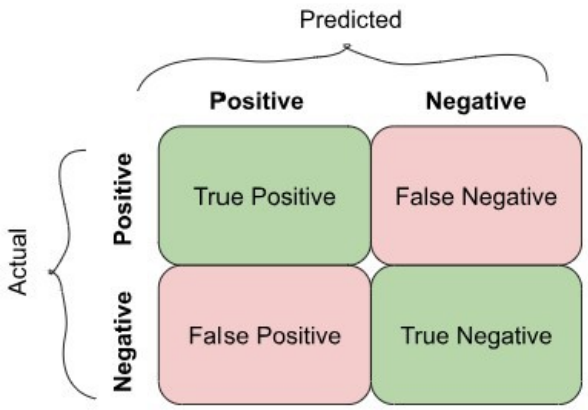

Figure 9. Example confusion matrix layout

Confusion matrices were extremely important for visualizing where the model had made errors and for balancing precision and recall by analyzing the trade-offs between FP and FN. For the binary model, a single confusion matrix was used to evaluate the classification of hemorrhagic versus non-hemorrhagic scans. For the multi-label model, separate confusion matrices were computed for each subtype, identifying specific subtype prediction errors.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

ROC and AUC evaluate the model's ability to distinguish between classes across all thresholds. These metrics provide another important attribute in addition to confusion matrices, precision, recall, and F1 scores. The ROC curve plots the True Positive Rate (TPR), otherwise known as recall, on the y-axis and the False Positive Rate (FPR) on the x-axis.

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

AUC, which is the area under the ROC curve, quantifies the model's performance; an AUC of 1.0 represents perfect classification, while 0.5 represents random guessing.

2D CNN image segmentation using U-Net

Unlike a classification model (which outputs a class label), a U-Net model predicts a mask for each input image, where each pixel in the mask contributes to the determination of whether that image is obtained from a hemorrhagic patient. The data used in this model contained 318 input images (full hemorrhagic CT scan slices) and 318 output images of the hemorrhage segmented out. Additionally, the dataset was split into three sets: training (70%), validation (15%) and testing (15%).

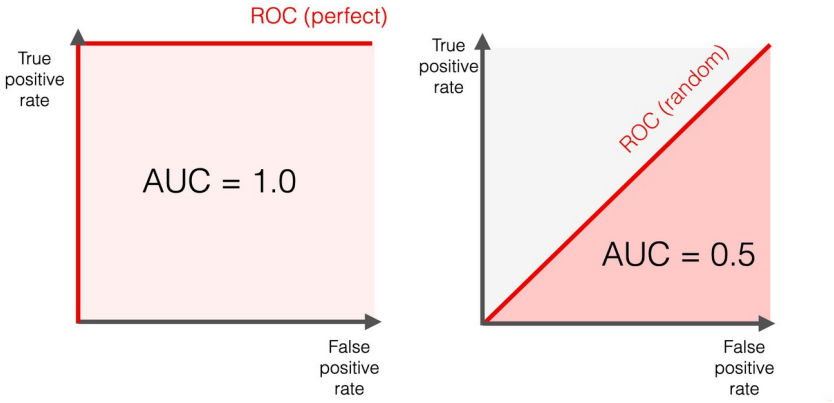

Figure 10. Example of a perfect classification and a random classification as shown by ROC-AUC curve (31). For the binary classification model, a single ROC curve was generated, showing the trade-off between sensitivity (TPR) and specificity (1-FPR) for hemorrhage detection. For the multi-label model, separate ROC curves were plotted for each subtype to assess performance independently. Based on all six ROC curves, the AUC values were output.

Downsampling (Encoder) path and bottleneck

The downsampling path was the first part of the U-Net architecture and extracted the spatial features from the input image while reducing the image resolution through max-pooling. The downsampling function applied a double convolutional layer consisting of n filters (n in the double convolutional layer for the U-Net model is a parameter of the function whose value depends on the depth of the U-Net architecture: the first downsampling block uses 64 filters while the subsequent blocks use 128, 256, and 512), a kernel size of 3x3, and a ReLU activation. This double convolutional layer extracted features from the input. Then, a MaxPooling operation with a pool size of 2x2 was applied, which reduced the spatial dimensions of the feature maps by half while retaining the important information. This operation represented the downsampling process. After max-pooling, dropout was applied with a rate of 0.3, meaning that 30% of the neurons in this layer were randomly set to zero during training. This prevented the model from overfitting to the training dataset. The bottleneck was the deepest part of the U-Net, where the model extracted the most abstract features. The convolutional block contained 124 filters, representing high-level features about the brain and the hemorrhage.

Upsampling (Decoder) path and output layer

The upsampling path comprised the second half of the U-Net architecture. First, the function Conv2DTranspose was used to increase the spatial dimensions of the feature map by a factor of 2. The skip connection was created by concatenating the feature maps from the downsampling path to the output from the upsampling path. This process helped retain fine-grained spatial information that might otherwise have been lost during downsampling. Dropout was set to 0.3, the same as in the downsampling path, to reduce overfitting. After upsampling, another double convolutional layer was applied to refine the feature maps. The output layer used a 1x1 convolution to produce a single-channel output for segmentation.

Additionally, a sigmoid activation function was used to map the values to probabilities between 0 and 1. Each pixel in the mask represented the probability that the pixel belonged to a hemorrhagic region.

Loss functions: Combined Dice loss and binary cross-entropy

In this segmentation task, the model used a combined loss function: Binary Cross-Entropy (BCE) and Dice Loss. This combination balanced pixel-wise accuracy from BCE and overlap from Dice Loss. Binary Cross-Entropy was used in the same way as in the classification models. What set the image segmentation model apart from the classification models was the utilization of Dice Loss. Dice Loss was used to measure the overlap between the true and predicted segmentation masks. It was useful when there was an imbalance between the images, as in medical images where hemorrhage (target) regions might have been small compared to the overall image.

$$DiceLoss(y, \underline{p}) = 1 - \frac{2 \cdot y \underline{p} + 1}{y + \underline{p} + 1} \quad (6)$$

Where y represented the truth labels (1 = hemorrhage, 0 = no hemorrhage), and $\underline{p}$ represented the predicted probability, which indicated how likely it was that each pixel belonged to a hemorrhage derived image. The numerator as a whole $2 \cdot y \underline{p} + 1$ accounted for the overlap between the prediction and truth, while the denominator $y + \underline{p} + 1$ added the total size of the predicted and true segments (32).

Training process

The training process allowed the U-Net model to learn how to segment hemorrhagic regions from CT images by minimizing the combined loss (BCE and Dice Loss). The model was trained for 20 epochs using the training data generator. Each epoch consisted of multiple steps, where batches of augmented images and corresponding segmentation masks were fed into the model. The validation generator evaluated the model's performance on a separate validation set after each epoch.

Evaluation and Dice coefficient

After training, the model was evaluated on a separate test set to assess its accuracy and loss. The Dice coefficient was used to provide a measure of how well the model segmented the hemorrhagic regions, focusing on the overlap between the predicted and true masks. A Dice coefficient near 1 indicated that the predicted segmentation closely matched the true mask.

Results and Discussion

2D Binary classification model

Table 2. Model performance overview of the 2D binary classification model (%)

Metric	Value
Final Training Accuracy	99.13
Final Validation Accuracy	97.04
Test Accuracy	97.72
Precision	97.45
Recall (Sensitivity)	98.59
F1 score	98.02
AUC Score	99.71

Training vs.Validation accuracy and loss

The training accuracy steadily increased, plateauing at 99.13% by the final epoch 10. This suggested that the model had effectively learned the patterns in the training dataset. The validation accuracy closely followed the training trend, reaching 97.04%, demonstrating that the model generalized well without overfitting. The training loss decreased consistently throughout the epochs, plateauing at 0.0251, reflecting the effective optimization of the model's parameters. Similarly, the validation loss decreased to 0.1039, indicating that the model learned well on unseen validation data without significant divergence from the training loss. The similarity between the final validation and test accuracy showed that the model had generalized well to unseen data, indicating that it had learned patterns effectively. The model did not show signs of overfitting, as often seen when training accuracy is very high but validation accuracy is significantly lower; which was been the case. Moreover, the steady decline in both training and validation loss reflected robust learning.

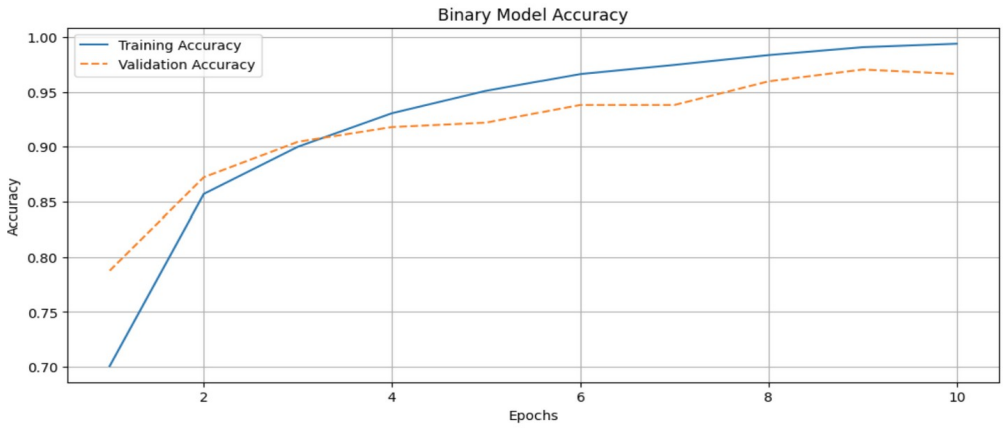

Figure 11. Graph of training and validation accuracy for the 2D binary classification model, and training and validation loss against epochs for 2D binary classification model

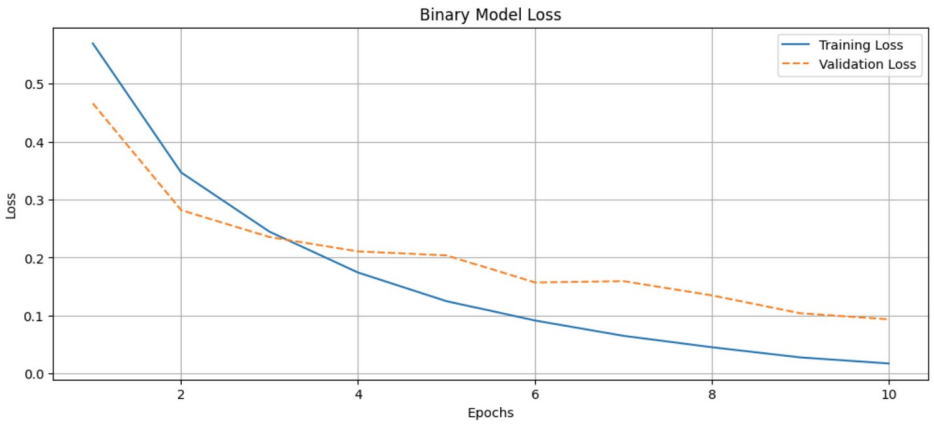

Figure 12. Graph of training and validation loss for the 2D binary classification model

Test set performance

The binary model achieved a test accuracy of 97.72%, which confirmed the model's ability to predict unseen data accurately. It also loosely matched the validation accuracy of 97.04%, showing that the model generalized well. This consistency between validation and test performance was a strong indicator that the model had not overfitted. The Receiver Operating Characteristic (ROC) for the binary model also demonstrated high classification performance. This was evidenced by the curve hugging closely at the top-left corner, indicating high sensitivity and specificity (high true positive, low false positive). The Area Under the Curve (AUC) of 99.71% highlighted the model's near-perfect ability to classify hemorrhagic and non hemorrhagic cases across all classification thresholds.

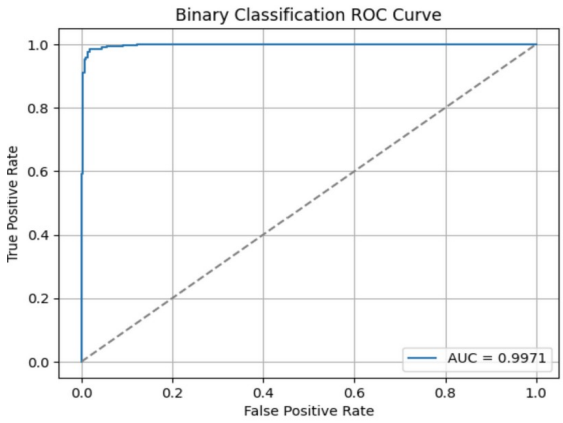

Figure 13. Graph of ROC curve for the binary classification model

Additionally, its high precision, recall, and F1 score indicated a strong balance between identifying hemorrhagic cases (true positives) and avoiding false positives. The precision score was 97.45%, highlighting the reliability of positive hemorrhage predictions. This meant

that out of all images classified as hemorrhagic, 97.45% were correctly identified. This was extremely important to ensure that patients were not missed as being hemorrhagic and incorrectly diagnosed. The recall rate of 98.59% showed that the model successfully detected 98.59% of all cases; almost all cases being correctly identified. Lastly, the F1 score of 98.02% reflected a balanced performance between precision and recall, making the model highly effective for hemorrhage detection.

Confusion matrix

A confusion matrix was also output to provide a better understanding of the model's performance. Figure 14 shows that out of 317 non-hemorrhagic images, 306 were classified correctly, and 6 were misclassified as hemorrhagic. Out of 427 hemorrhagic images, 421 had been classified correctly, and 6 were misclassified as non-hemorrhagic. The low number of false negatives (6) are crucial in medical diagnosis, since this implies that nearly all hemorrhagic cases were detected. See figure 15 for correctly and incorrectly diagnosed CT scans.

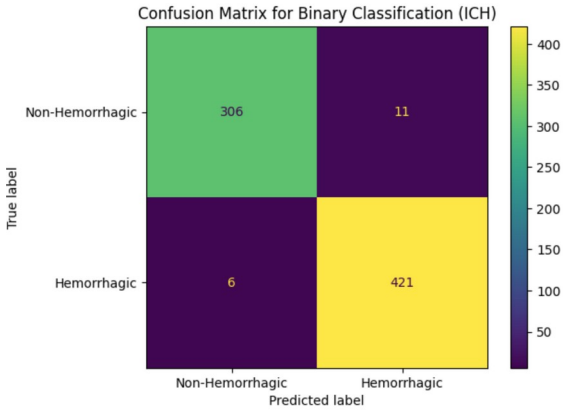

Figure 14. Confusion matrix for the 2D binary classification model

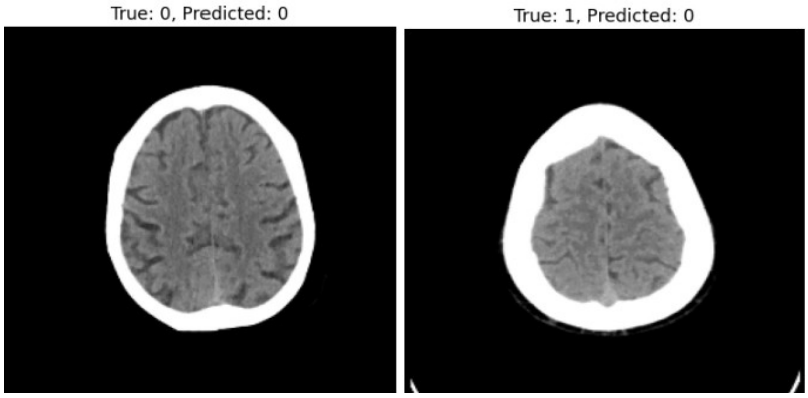

Figure 15. The image on the left is non-hemorrhagic and was correctly classified, whereas the image on the right was incorrectly classified as non-hemorrhagic.

2D Multi-Label classification model

Table 3. Model performance overview of the 2D multi-label classification model (%)

Metric	IPH	IVH	SDH	EDH	SAH
Final Training Accuracy	84.50				
Final Validation Accuracy	81.12				
Test Accuracy	81.26				
Precision	96.54	99.05	98.96	100.00	92.89
Recall (Sensitivity)	95.80	97.20	94.06	71.43	100.00
F1 Score	96.17	98.11	96.45	83.33	96.31
AUC Score	98.86	99.66	99.89	99.36	99.23

Training vs. Validation accuracy and loss

The training accuracy started at 61.72% at epoch 1 and, for the most part, increased steadily with each epoch. By epoch 10, the training accuracy reached 84.50%, confirming that the model had effectively learned patterns within the training set. The validation accuracy started at 76.46%, already relatively high compared to the training accuracy in the first epoch. This suggested that the model was able to generalize well from the start. By epoch 4, the validation accuracy had reached 84.15%, surpassing the training accuracy, indicating that the model generalized well to the validation data early in training. The validation accuracy then fluctuated after epoch 4, peaking at 84.38% in epoch 9 and slightly decreasing to 81.12% by epoch 10. These fluctuations in validation accuracy suggested that the model experienced small variations in performance on unseen data during optimization, potentially due to the complexity of multi-label classification and the varying frequency of certain subtypes. These fluctuations were minor, however, and did not reflect instability in the learning process.

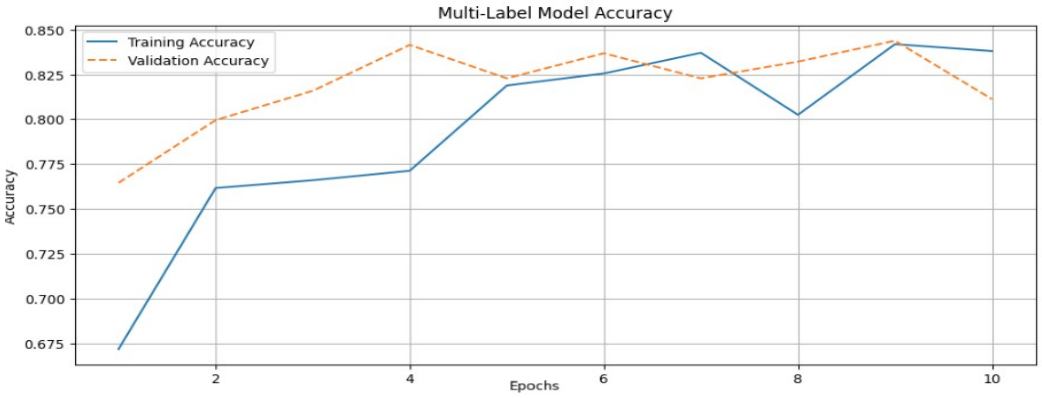

Figure 16. Graph of training and validation accuracy for the 2D multi-label classification model

The training loss decreased consistently from 0.5906 in epoch 1 to 0.0107 by epoch 10, reflecting that the model's parameters were effectively optimized. This also confirmed that

the model fit the training data well without significant overfitting. The validation loss started at 0.4610, lower than the training loss in the first epoch. The validation loss also decreased steadily, reaching 0.1021 by epoch 10, closely aligning with the training loss. The convergence between training and validation loss suggested that the model learned robustly without overfitting, generalizing well to unseen validation data.

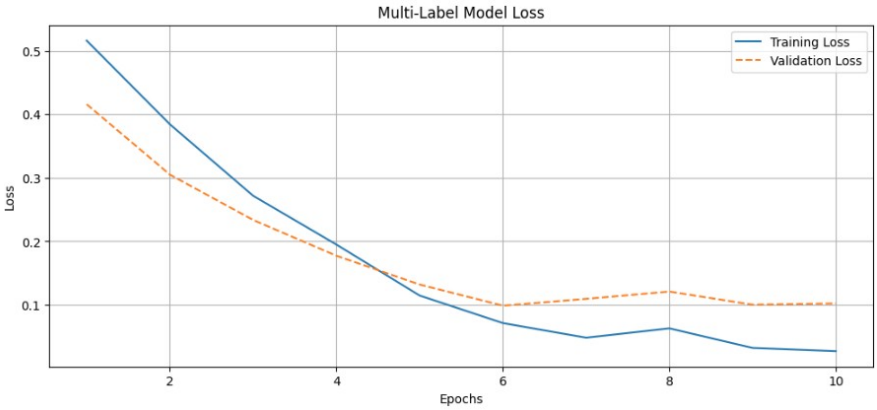

Figure 17. Graph of training and validation loss for the 2D multi-label classification model

Test set performance

The model demonstrated high accuracy and robust metrics across most subtypes, with an overall test accuracy of 81.12%. This showed that the multi-label model generally performed well across all subtypes. However, there were differences in subtype metrics that could be attributed to the imbalance in the number of cases in the test set for each subtype (e.g., fewer EDH cases compared to IPH or SAH cases). However, the number of subtype cases reflected real-world cases.

For Intraparenchymal Hemorrhages, a high precision of 96.54% indicated that the model was reliable in predicting IPH cases, with few false positives. A high recall of 95.80% showed that most IPH cases were successfully detected in the test set, but a small number of false negatives were not detected. A high F1 score of 96.17% suggested a strong balance between precision and recall, reflecting robust classification. An AUC of 98.87% suggested that the model had an acceptable ability to distinguish between IPH and non-IPH cases. The large number of IPH cases in the dataset (267 in the test set) likely contributed to the model's strong performance, as the model had sufficient data to learn IPH patterns effectively.

For Intraventricular Hemorrhages, a precision of 99.05% indicated very few false positives, meaning nearly all predicted IVH cases were correctly identified. A recall of 97.20% showed that most true IVH cases were identified, with a small number of false negatives undetected. An F1 score of 98.11% reflected strong performance in balancing precision and recall, and an AUC of 99.66% demonstrated that the model performed near-perfect discrimination between positive and negative IVH cases. With 103 IVH cases in the test set, the model's high recall suggested that it generalized well to unseen data for IVH, despite having unseen

data for IVH, and despite having fewer examples compared to IPH.

For Subdural Hemorrhages, a precision of 98.86% ensured minimal false positives for SDH cases. A recall of 94.06% showed that most true SDH cases had been detected, although there were some missed cases (false negatives). An F1 score of 96.45% reflected balanced precision and recall, and a near-perfect AUC score of 99.89% reflected the model's ability to classify SDH cases across all thresholds. The moderate number of SDH cases in the test set (115) likely contributed to the slight drop in recall. While sufficient examples existed, fewer examples compared to IPH could explain the slight performance drop.

For Epidural Hemorrhages, a perfect precision was scored, indicating no false positives: every predicted EDH case was correct. A notably lower recall of 71.43%, however, indicated that a significant number of EDH cases were missed (false negatives). An F1 score of 83.33% reflected the disparity between perfect precision and moderate recall, and an AUC score of 99.36% showed that the model's acceptable discrimination ability for EDH cases. The low number of EDH cases in the test set (35) likely contributed to the lower recall. The model struggled to generalize well for EDH due to insufficient training data (only 322 cases in the entire dataset).

For all subtypes, the ROC curves demonstrated high classification performance. All curves, similar to the binary classification model ROC curve, hugged the top-left corner. This demonstrated a combination of high sensitivity and specificity. Additionally, AUC scores were > 98% for all subtypes, reflecting near-perfect classification abilities across thresholds.

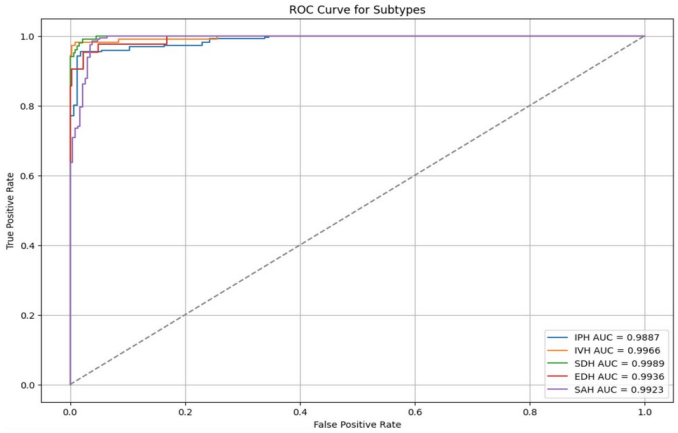

Figure 18. Graph of ROC curve for the multi-label classification models, with separate ROC curves representing each subtype.

Confusion matrices

Once again, confusion matrices for each subtype were output to gain a better understanding of the model's performance. The confusion matrix in Figure 19 shows that out of 267 IPH cases, 251 were correctly classified as IPH, 11 were missed as IPH cases, and 9 were incorrectly classified as IPH cases. The large number of IPH cases probably allowed the model to achieve high precision and recall, as

previously mentioned. See Figure 20 for an example of an incorrectly and correctly classified CT scan as Intraparenchymal.

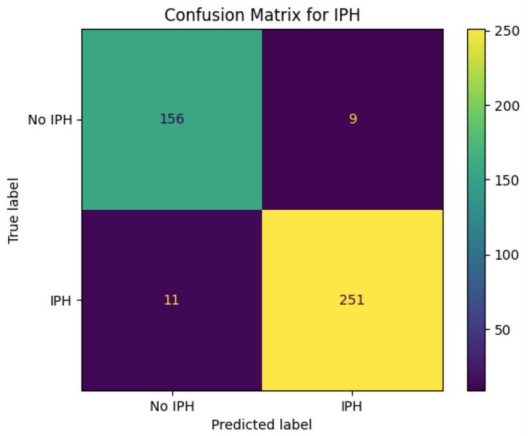

Figure 19. Confusion matrix for Intraparenchymal Hemorrhage

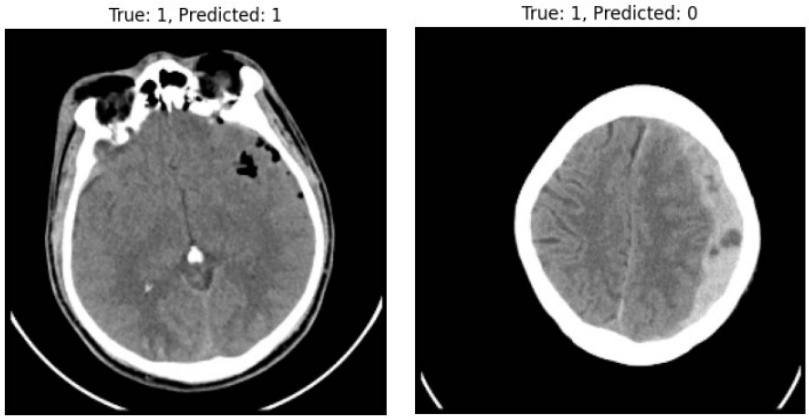

Figure 20. The hemorrhage on the left was correctly classified as IPH, while the image on the right was incorrectly classified as non-IPH.

Figure 21 shows that out of 103 IVH cases, 104 were correctly classified as IVH, 3 were incorrectly classified as non-IVH, and 1 was incorrectly classified as IVH. The relatively smaller class size compared to IPH still enabled high performance, probably due to strong patterns learned during training.

Figure 22 shows that out of 115 SDH cases, 95 were correctly classified as SDH, 6 were incorrectly classified as non-SDH, and 1 case was incorrectly classified as SDH. The moderate number of examples allowed good performance, but the slightly fewer examples than IPH likely caused a slightly lower recall.

Figure 23 shows that out of 35 EDH cases, 30 were correctly classified as EDH, 12 were missed EDH cases, and there were no incorrect classifications as EDH. The low number of

EDH cases may have significantly impacted recall, as the model struggled to generalize for this subtype due to limited training examples, as previously mentioned.

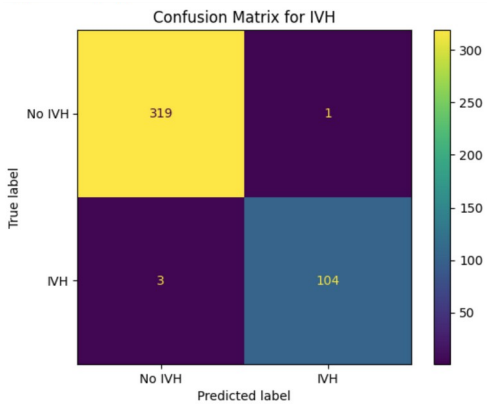

Figure 21. Confusion matrix for Intraventricular Hemorrhages

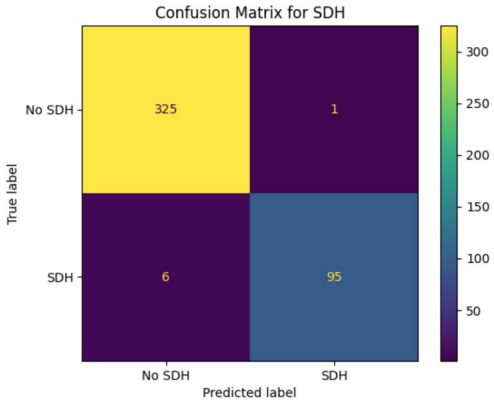

Figure 22. Confusion matrix for Subdural Hemorrhages

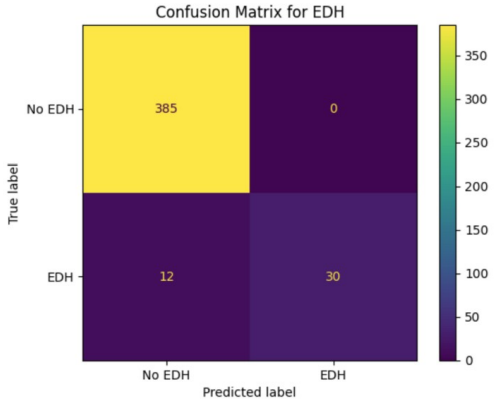

Figure 23. Confusion matrix for Epidural Hemorrhages

Figure 24 shows that out of 202 SAH cases, 196 were correctly classified as SAH, there were no missed SAH cases, and there was a slight over-classification of SAH cases with 15 false positives. The large number of SAH cases likely contributed to a perfect recall but led to a

slightly higher false positive rate, thereby impacting precision.

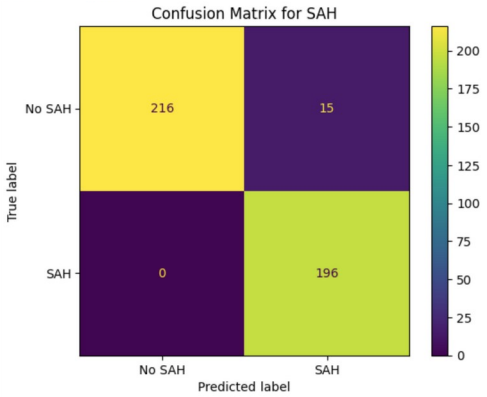

Figure 24. Confusion matrix for Subarachnoid Hemorrhages

2D Classification image segmentation - U-Net model

Table 4. Model performance overview of the 2D image segmentation (%)

Final Training Accuracy	98.66
Final Validation Accuracy	97.34
Test Accuracy	98.72
Test Dice Coefficient	43.78

Training & Validation accuracy and loss

The training accuracy steadily increased and plateaued at ~ 98.66%, suggesting that the model had learned effectively. The validation accuracy followed a similar trend, reaching 97.34% by the final epoch. This closeness between training and validation accuracy showed that the model generalized well to the validation set without overfitting. The training loss decreased to 0.8676 by the final epoch, suggesting effective learning and fitting of the data. The validation loss reduced steadily to 0.8194, close to that of the training loss.

Test set performance

The test accuracy was 98.72%, indicating that the model performed well on unseen data. With a Dice coefficient of 43.78%, the model might have been over or under-predicting in certain areas of the segmentation task. The Dice coefficient is a more nuanced metric for evaluating the overlap between predicted and ground-truth masks, and a low score suggested that the model was not yet been fully optimized for segmentation, despite the high accuracy. See figure 26 for the image segmented output performed by the testing model.

Figure 25. Smoothed out graph of training and validation accuracy, and training and validation loss for the 2D image segmentation model

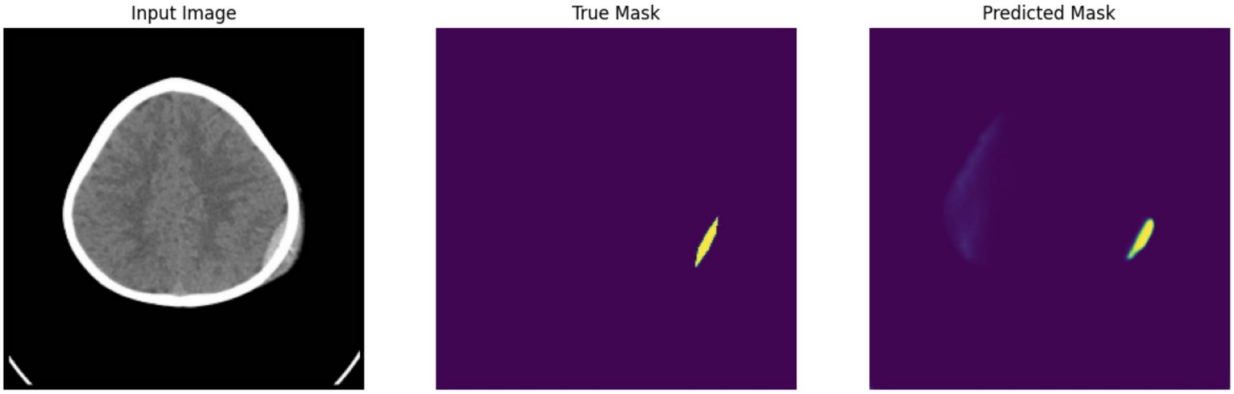

Figure 26. Picture of a CT scan with true hemorrhage segmented out shown, along with the predicted mask output by the test set

Errors and potential optimization

Dice Optimization: Since segmentation models required more nuanced predictions, the combined loss of binary cross-entropy and Dice loss was appropriate, but the model might have benefited from assigning more weight to the Dice loss to prioritize segmentation quality. Additionally, increasing the variety of augmentations to expose the model to more diverse patterns might have further improved its ability to generalize.

Test Performance: Segmentation tasks often suffered from image imbalance, where most of the image was background and only a small portion represented the hemorrhage. Increasing the model's focus on smaller or less frequently occurring regions could have helped increase the Dice Coefficient score. Techniques like image balancing or focal loss might have been helpful in this regard.

Conclusions

The objective of this research was to construct deep learning models capable of detecting intracranial hemorrhages using CT scans,

automating the classification and segmentation for faster, more accurate diagnosis. Given the high mortality rates associated with intracranial hemorrhages, efficient, automated detection systems are more critical than ever for improving patient outcomes and reducing diagnostic delays. Two primary models were constructed, a 2D CNN-based classification model for identifying hemorrhage vs non-hemorrhage cases and also classifying the subtype of hemorrhagic scans, and a U-Net model for segmenting the hemorrhage region in CT scans. Data augmentation, resizing, normalization, and custom loss functions were utilized to enhance the model's performance and robustness.

Two types of CNN models for classification were tested, a binary classification to first classify CT scans as hemorrhagic or non-hemorrhagic, followed by a multi-label classification model to further classify the subtype of hemorrhagic CT scans. The 2D CNN binary classification model and the 2D CNN multi-label classification model achieved test accuracies of 97.7% and 81.3% respectively. The U-Net model achieved a test accuracy of 98.7%, although the Dice coefficient of 43.8% indicated some challenges in precisely segmenting the hemorrhagic regions. The acceptable accuracy of these 2D classification models could be attributed to the rich feature set provided by brain CT scans, which included clear contrasts for hemorrhage detection. The Dice coefficient for the segmentation model suggested that further optimization, such as enhanced data augmentation and loss function tuning, would be necessary to improve the overlap between the predicted and the true masks.

To improve models, future work should collect a larger and more balanced dataset, apply advanced augmentation techniques, and experiment with other architectures such as attention mechanisms or modified U-Net models. In summary, this research demonstrated that deep learning models, particularly those trained on brain CT scans, hold significant promise for improving the speed and accuracy of intracranial hemorrhage detection, potentially revolutionizing the diagnostic process in clinical settings.

Acknowledgements

The contributions of Al Hilla Teaching Hospital, Iraq, and Murtadha Hssayeni, and the radiologist centers in New Delhi, India who provided the datasets used in this research, are acknowledged. Appreciation is extended to the radiologists who annotated the datasets, as well as the open-source community for making valuable resources accessible for research.

References

1. Cleveland Clinic. (2023). Brain bleed, hemorrhage (intracranial hemorrhage). *Cleveland Clinic.* https://my.clevelandclinic.org/health/diseases/14480-brain-bleed-hemorrhage-intracranial-hemorrhage

2. Arbabshirani, M. R., Fornwalt, B. K., Mongelluzzo, G. J., Suever, J. D., Geise, B. D., Patel, A. A., et al. (2018). Advanced machine learning in action: Identification of intracranial hemorrhage

on computed tomography scans of the head with clinical workflow integration. *NPJ Digital Medicine, 1*(9). https://doi.org/10.1038/s41746-017-0015-z

3. Roshan, M. P., Al-Shaikhli, S. A., Linfante, I., Antony, T. T., Clarke, J. E., Noman, R., et al. (2024). Revolutionizing intracranial hemorrhage diagnosis: A retrospective analytical study of Viz.ai ICH for enhanced diagnostic accuracy. *Cureus, 16*(8), e66449. https://doi.org/10.7759/cureus.66449

4. UCLA Health. (2024). Acute subdural hematomas. *UCLA Health.* https://www.uclahealth.org/medical-services/neurosurgery/conditions-treated/acute-subdural-hematomas

5. Cleveland Clinic. (2022). Subarachnoid hemorrhage (SAH). *Cleveland Clinic.* https://my.clevelandclinic.org/health/diseases/17871-subarachnoid-hemorrhage-sah

6. Piccolo, B., Marchignoli, M., Pisani, F. (2022). Intraventricular hemorrhage in preterm newborn: Predictors of mortality. *Acta Bio Medica: Atenei Parmensis, 93*(2), e2022041. https://doi.org/10.23750/abm.v93i2.11187

7. Lahti, A.-M., Nätynki, M., Huhtakangas, J., Bode, M., Juvela, S., Ohtonen, P., Tetri, S. (2021). Long-term survival after primary intracerebral hemorrhage: A population-based case–control study spanning a quarter of a century. *European Journal of Neurology, 28*(11), 3663–3669. https://doi.org/10.1111/ene.14988

8. Hssayeni, M., Al-Janabi, M., Al-Ani, A., Al-Khafaji, H. F., Yahya, Z. A., Ghoraani, B. (2019). Intracranial hemorrhage segmentation using deep convolutional model. *arXiv.* https://doi.org/10.48550/arXiv.1910.08643

9. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

10. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., Ronneberger, O. (2016). 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, W. Wells (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (pp. 424–432). Springer. https://doi.org/10.1007/978-3-319-46723-8_49

11. Taparia, A. (2023). U-Net architecture explained. *GeeksforGeeks.* https://www.geeksforgeeks.org/u-net-architecture-explained/

12. Wang, H., Wang, X. (2023). MSRL-Net: An automatic segmentation of intracranial hemorrhage for CT images based on the U-Net framework. *Applied Sciences, 13*(21), 11781. https://doi.org/10.3390/app132111781

13. Chang, C. S., Chang, T. S., Yan, J. L., Ko, L. (2023). All Attention U-Net for semantic segmentation of intracranial hemorrhages in head CT images. *arXiv.* https://doi.org/10.48550/arXiv.2312.10483

14. Ahmed, N. S., Prakasam, P. (2024). Spatial attention-based CSR-Unet framework for subdural and epidural hemorrhage segmentation and classification using CT images. *BMC Medical Imaging, 24*(285). https://doi.org/10.1186/s12880-024-01455-6

15. Piao, Z., Gu, Y. H., Jin, H., Yoo, S. J. (2023). Intracerebral hemorrhage CT scan image segmentation with HarDNet based transformer. *Scientific Reports, 13*(7208). https://doi.org/10.1038/s41598-023-33775-y

16. Omarov, B., Tursynova, A., Postolache, O., Gamry, K., Batyrbekov, A., Aldeshov, S., et al. (2022). Modified UNet model for brain stroke lesion segmentation on computed tomography images. *Computers, Materials & Continua, 71*(3), 4701–4717. https://doi.org/10.32604/cmc.2022.020998

17. Saimassay, G., Begenov, M., Sadyk, U., Baimukashev, R., Maratov, A., Omarov, B. (2024). Enhanced U-Net architecture for lung segmentation on computed tomography and X-ray images. *International Journal of Advanced Computer Science and Applications, 15*(5), 921–930. https://doi.org/10.14569/IJACSA.2024.0150593

18. GeeksforGeeks. (2024). Convolutional neural network (CNN) in machine learning. *GeeksforGeeks.* https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/

19. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2015). Recent advances in convolutional neural networks. *Pattern Recognition.* https://doi.org/10.1016/j.patcog.2017.10.013

20. Umapathy, S., Murugappan, M., Bharathi, D., Thakur, M. (2023). Automated computer-aided detection and classification of intracranial hemorrhage using ensemble deep learning techniques. *Diagnostics, 13*(18), 2987. https://doi.org/10.3390/diagnostics13182987

21. Von Kummer, R., Broderick, J. P., Campbell, B. C. V., Demchuk, A., Goyal, M., Hill, M. D., et al. (2015). The Heidelberg bleeding classification: Classification of bleeding events after

ischemic stroke and reperfusion therapy. *Stroke, 46*(10), 2981–2986. https://doi.org/10.1161/STROKEAHA.115.010049

22. Lee, J. Y., Kim, J. S., Kim, T. Y., Kim, Y. S. (2020). Detection and classification of intracranial haemorrhage on CT images using a novel deep-learning algorithm. *Scientific Reports, 10*(1), 20546. https://doi.org/10.1038/s41598-020-77441-z

23. Marbun, J. T., Seniman, Andayani, U. (2018). Classification of stroke disease using convolutional neural network. *Journal of Physics: Conference Series, 978*(1), 012092. https://doi.org/10.1088/1742-6596/978/1/012092

24. Babutain, K., Hussain, M., Aboalsamh, H., Al-Hameed, M. (2021). Deep learning-enabled detection of acute ischemic stroke using brain computed tomography images. *International Journal of Advanced Computer Science and Applications, 12*(12), 386–396. https://doi.org/10.14569/IJACSA.2021.0121252

25. Hssayeni, M. (2020). Computed tomography images for intracranial hemorrhage detection and segmentation. *PhysioNet.* https://doi.org/10.13026/4nae-zg36

26. Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., et al. (2018). Deep learning algorithms for detection of critical findings in head CT scans: A retrospective study. *The Lancet, 392*(10162), 2388–2396. https://doi.org/10.1016/S0140-6736(18)31645-3

27. Wang, X., Shen, T., Yang, S., Lan, J., Xu, Y., Wang, M., et al. (2021). A deep learning algorithm for automatic detection and classification of acute intracranial hemorrhages in head CT scans. *NeuroImage: Clinical, 32*, 102785. https://doi.org/10.1016/j.nicl.2021.102785

28. Deepchecks AI. (2024). Rectified linear unit (ReLU). *Deepchecks Glossary.* https://www.deepchecks.com/glossary/relu/

29. Unzueta, D. (2022). Fully connected layer vs. convolutional layer: Explained. *Built In.* https://builtin.com/artificial-intelligence/fully-connected-layer-vs-convolutional-layer

30. Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep learning.* MIT Press. https://mitpress.mit.edu/9780262035613

31. Evidently AI Team. (2024). How to explain the ROC curve and ROC AUC score? *Evidently AI.* https://evidentlyai.com/guide/roc-auc

32. Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M. (2017). Generalized Dice overlap as a deep learning loss function for highly unbalanced segmentations. In M. Cardoso et al. (Eds.), *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS 2017. Lecture Notes in Computer Science* (Vol. 10553, pp. 240–248). Springer, Cham. https://doi.org/10.1007/978-3-319-67558-9_28