IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 May 2023, accepted 7 July 2023, date of publication 1 August 2023, date of current version 11 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3300771

RESEARCH ARTICLE

Intracranial Hemorrhage Classification From CT Scan Using Deep Learning and Bayesian Optimization

SHIFAT E. ARMAN[1], SAYED SAMINUR RAHMAN[2], NILOY IRTISAM[1], SHAMIM AHMED DEOWAN[1], MD. ARIFUL ISLAM[1], SAADMAN SAKIB[1], AND MEHEDI HASAN[1]

[1]Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka 1000, Bangladesh
[2]Aromo Health, Chattogram, Bangladesh

Corresponding author: Shamim Ahmed Deowan (shamimdeowan.rme@du.ac.bd)

This work was supported by the Centennial Research Grant (CRG), University of Dhaka, Dhaka, Bangladesh.

ABSTRACT Intracranial hemorrhage is a medical condition characterized by bleeding within the skull or brain tissue. It has mainly five subtypes: epidural, subdural, subarachnoid, intraparenchymal, and intraventricular. To ensure a successful outcome for a patient, timely and accurate identification of intracranial hemorrhage is crucial. However, a shortage of radiologists, particularly in rural areas, can lead to a delay in diagnosis. In this work, we proposed an automatic way of identifying intracranial hemorrhage from a Computed Tomography (CT) scan. To classify intracranial hemorrhage accurately, we have optimized the Densely Connected Convolutional Network (DenseNet) using Bayesian Optimization (BO). We utilized Bayesian optimization (BO) to determine the optimal learning rate, optimizer, and the number of nodes in the dense layer for the DenseNet architecture. Our proposed model can analyze CT scans to detect the presence of hemorrhage and identify its subtype. The optimized DenseNet model showcased remarkable performance. By ensuring accurate and reliable diagnoses, our method will assist doctors in making better-informed decisions and providing better care for their patients.

INDEX TERMS Bayesian optimization, CT scan, deep learning, intracranial hemorrhage, medical image analysis, radiology, stroke, DenseNet, automatic diagnosis.

I. INTRODUCTION

When bleeding within the skull causes blood to gather around or within the brain, it leads to intracranial hemorrhage, a condition that poses a serious threat to life. Several factors can contribute to this condition, including trauma, high blood pressure, or rupture of an aneurysm. Aside from severe headaches, nausea, vomiting, seizures, and loss of consciousness, intracranial hemorrhage symptoms can vary based on the site and severity of the bleeding. If not diagnosed and treated promptly, intracranial hemorrhage can cause permanent brain damage or death. Therefore, accurate and timely diagnosis of intracranial hemorrhage is essential to ensure the patient's survival and effective treatment.

The associate editor coordinating the review of this manuscript and approving it for publication was Nuno M. Garcia.

Diagnosing intracranial hemorrhage involves the careful analysis of medical history, physical examination findings, and the utilization of advanced imaging modalities like CT scans. Although the manual interpretation of CT scans by radiologists can provide invaluable insights, this process can often prove time-consuming, leading to delays in diagnosis and treatment. Particularly in rural areas, the scarcity of radiologists may further exacerbate these delays, putting patients at risk of severe brain damage or even death. It can also be challenging for radiologists to detect small bleeds in the brain, and some minor hemorrhages may go undetected. As a result, relying solely on manual interpretation for diagnosis can lead to errors and reduce the availability of timely diagnosis.

The adoption of computer-aided diagnosis (CAD) offers a promising alternative to manual diagnosis of intracranial

hemorrhage, especially in scenarios where there is a shortage of radiologists or a need for rapid and precise diagnoses. CAD systems utilize machine learning algorithms to analyze medical images and provide an automatic diagnosis of diseases. Particularly in emergency situations, where prompt treatment is crucial, CAD can significantly reduce diagnosis time and enhance accuracy.

This paper presents a CAD system that uses Deep Learning (DL) and Bayesian Optimization (BO) to detect intracranial hemorrhage. Several works have been done on the detection of intracranial hemorrhages from head CT scan [1], [2], [3]. Unlike other works, we have done ours from an optimization-focused perspective. Our system utilizes a Convolutional Neural Network (CNN) to analyze CT scans and accurately identify intracranial hemorrhage. The CNN undergoes training on a comprehensive dataset of CT scans, with its hyperparameters optimized through the BO algorithm. This method has proven to be highly effective in diagnosing intracranial hemorrhage. Medical professionals can use this system to diagnose and treat intracranial hemorrhages, especially in emergency situations where timely diagnosis and treatment are crucial. The graphical abstract of our work is illustrated in FIGURE 1.

We utilized the DenseNet [4] architecture proposed by Huang et al. as our convolutional neural network model. Compared to other CNN architectures, DenseNet offers significant advantages. This network's dense connectivity pattern makes it possible to propagate information and gradients efficiently throughout the network, thereby preventing vanishing gradients and improving performance. Furthermore, DenseNet's dense blocks facilitate efficient information propagation and fusion across layers, which facilitates the learning of complex and hierarchical data representations. These advantages made our task simpler, which allowed us to create a reliable and accurate model.

We used the Bayesian Optimization (BO) technique [5] to fine-tune the DenseNet architecture's hyperparameters. As the DenseNet architecture was complex and large, we preferred BO over other exhaustive search techniques. There are many hyperparameter combinations, but evaluating them all can be impractical and computationally expensive. BO uses a surrogate function and an acquisition function to focus on the most promising regions of the search space. This saves time and resources. The surrogate function is constructed on the basis of previous evaluations, enabling it to predict the performance of neural networks under different hyperparameter configurations. In order to determine which hyperparameter combinations to try next, the acquisition function is used. It maintains the trade-off between exploration and exploitation. These two functions enable BO to perform hyperparameter optimization efficiently and quickly.

The contributions of this paper are as follows:

1) In the proposed CAD system, deep learning and Bayesian optimization techniques are used to optimize the convolutional neural network, improving the effectiveness of hemorrhage diagnosis.
2) This method saves significant time and resources by optimizing the network on a small dataset before training it on a large dataset.
3) In an emergency situation where time is of the essence, the proposed system offers a potentially life-saving alternative to manual interpretation of CT scans.
4) In rural areas with a lack of skilled radiologists, the proposed system offers a valuable diagnostic and treatment tool for intracranial hemorrhage.

The subsequent sections of the paper are presented in the following order: Section II provides a review of the relevant literature. The methodology employed is described in Section III. The experimental outcomes are outlined in Section IV, followed by the conclusion of the paper in Section V.

II. LITERATURE REVIEW

Intracranial hemorrhages are commonly detected using medical imaging techniques, such as CT scans and MRIs. Chang et al. [6] developed an ROI-based CNN architecture for accurate identification of intracranial hemorrhage from non-contrast computed tomography (NCCT) scans. Their proposed method was able to detect and quantify intraparenchymal, subarachnoid, epidural, and subdural hemorrhages. Chilamkurthy et al. [7] developed a deep learning model that can detect critical abnormalities in the head that include hemorrhage, fractures, midline shift, and mass effect from CT scans. Their proposed method was also able to detect subtypes of hemorrhages. Lee et al. [8] developed an explainable deep convolutional neural network for the identification of brain hemorrhage from CT scans using attention maps and prediction basis.

Grewal et al. [9] took a cascaded CNN-RNN approach to detect brain hemorrhages. They developed a model named (RADnet) that uses CNN for extracting features from individual slices and LSTM for capturing the dependencies between the slices. They used a 40-layer DenseNet architecture for the classification task. Using RADnet, they were able to achieve performance comparable to senior radiologists. A notable limitation of this study is the utilization of a small dataset for training the RADnet model. Hence, the data might not be able to capture all sorts of difficult cases. Nguyen et al. [10] also performed a similar method to detect intracranial hemorrhage. They used ResNet-50 and SE-ResNeXt-50 to extract features and LSTM to capture the dependencies between slices. A 3D convolutional neural network was implemented by Jnawali et al. [11] to detect brain hemorrhage from CT scans.

For the classification and segmentation of intracranial hemorrhage, Kuo et al. [12] developed a Patch-based Fully Connected Neural Network (PatchFCN). Their segmentation model can perform semantic segmentation, which is better for this task compared to the traditional mask R-CNN approach.

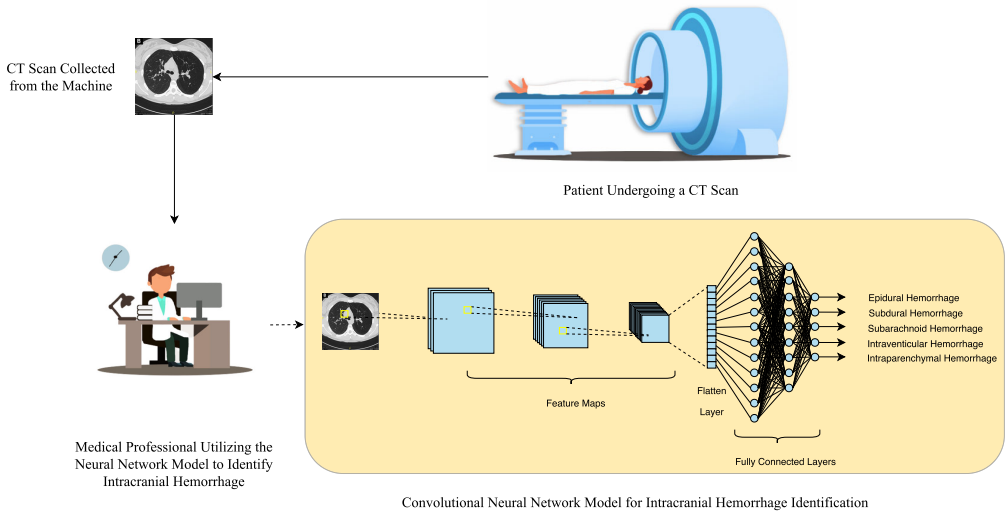


FIGURE 1. Automated diagnosis of intracranial hemorrhage using deep learning.

The model was able to localize some abnormalities which were even missed by some expert radiologists.

Deep learning has emerged as one of the most powerful tools for the analysis of medical images in the last few years. This technique has been successfully applied to various medical applications, including pulmonary nodule detection, liver segmentation, retroperitoneal sarcoma segmentation, lung cancer risk prediction, and pancreatic ductal adenocarcinoma detection from CT scan. Li et al. [13] employed deep learning for pulmonary nodule detection, while Araujo et al. [14] used it for liver segmentation. Salvaggio et al. [15] utilized this technique for retroperitoneal sarcoma segmentation. Furthermore, Mikhael et al. [16] proposed a deep learning-based approach for lung cancer risk prediction, while Alves et al. [17] developed a deep learning system for pancreatic ductal adenocarcinoma detection.

Deep learning techniques were also used successfully to detect COVID-19 in CT scans. Islam et al. [18] developed a novel CNN model to extract 100 prominent features from CT scan images and then used an ensemble model for COVID-19 classification. Kogilavani et al. [19] compared various CNN architectures for COVID-19 detection. Ravi et al. [20] employed a large-scale learning approach using a stacked ensemble meta-classifier and feature fusion that outperformed existing models. These studies' outcomes reveal the capabilities of deep learning for medical image analysis.

It is important to note that many of the studies discussed in this section do not take optimization into account. However, optimizing a neural network can lead to significant performance improvements. Therefore, it's essential to consider optimization techniques in the development of neural networks. In this research, we used a smaller dataset to optimize the DenseNet, followed by a larger dataset to retrain the most promising configuration of the DenseNet for an extended period of time. We have used the BO to optimize the DenseNet in this work.

III. METHODOLOGY

The methodology of our work is presented in FIGURE 2. There are mainly 4 steps: (a) Dataset Description, (b) Image Pre-Processing, (c) Model Development & Optimization (d) Model Evaluation. Each of the steps is discussed in the following sections.

A. DATASET DESCRIPTION

In this study, we used the RSNA Intracranial Hemorrhage Detection dataset [21]. The dataset contains CT images of five subtypes of intracranial hemorrhages - (a) intraparenchymal, (b) intraventricular, (c) subarachnoid, (d) subdural, and (e) epidural. The aim is to detect the presence of hemorrhage and identify its subtype. A total of 1,074,271 unique images are included in this dataset to detect brain hemorrhage. Images were collected from clinical images and the archives of Thomas Jefferson University, Stanford University, and the Universidade Federal de São Paulo. Creating such a large dataset aims to improve detection accuracy and reduce errors. Using commercial online annotation platforms, 60 neuroradiologists from the American Society of Neuroradiology have annotated five types of hemorrhage - subarachnoid, intraventricular, subdural, epidural, and intraparenchymal. Different subtypes of intracranial hemorrhages are shown in FIGURE 3. The RSNA dataset contains images of both labelled and unlabelled CT scans. We utilized the labelled set in our experiments. The labelled dataset was randomly split into train, validation, and test sets with a ratio of 80:10:10.

Intraparenchymal hemorrhage refers to the phenomenon where a blood vessel ruptures inside the brain, leading to bleeding within the brain tissue itself. It can be caused by a variety of reasons that include hypertension, aneurysms, arteriovenous malformations, or head trauma. Intraventricular

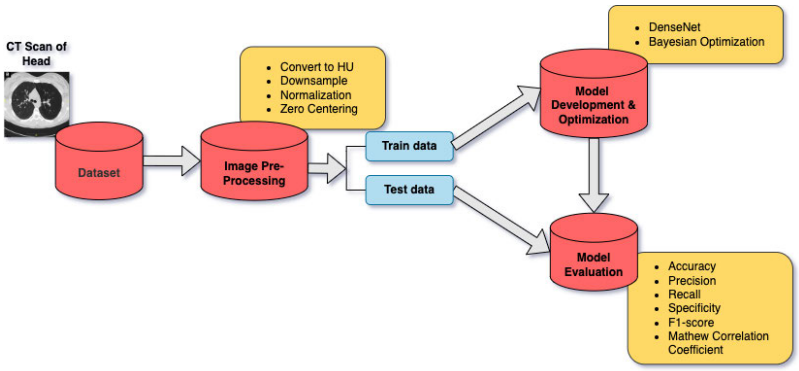


FIGURE 2. Methodology of our work.

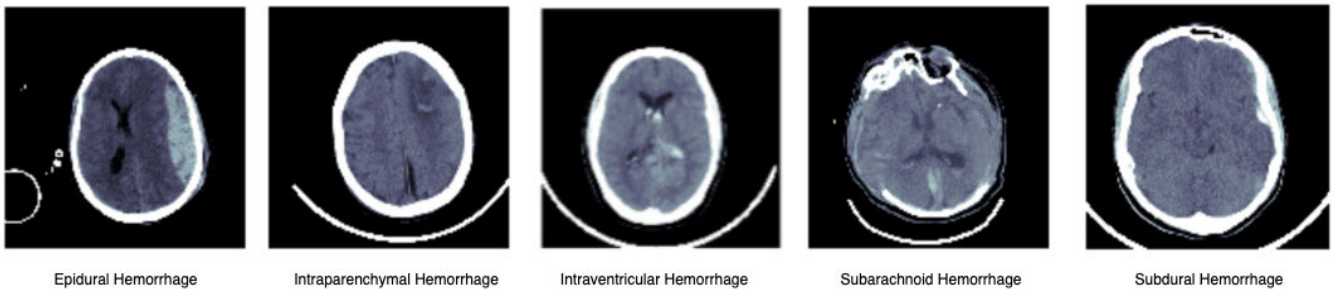


FIGURE 3. Different subtypes of intracranial hemorrhages [21].

hemorrhage occurs when blood vessels within the ventricular system of the brain rupture, causing bleeding into the fluid-filled spaces of the brain. Premature infants are most at risk for this type of hemorrhage, but it can also occur in adults after a head injury. Subdural hemorrhage occurs when blood vessels between the brain and the dura mater rupture, causing bleeding in the space between the brain and the skull. In addition to head trauma, subdural hemorrhages can also be caused by both blood clotting disorders and brain tumors. Subarachnoid hemorrhage occurs when blood vessels between the arachnoid and pia mater rupture, causing bleeding in the space between the brain and the meninges. The most common cause of subarachnoid hemorrhages is ruptured aneurysms, but other medical conditions can also cause them. Epidural hemorrhage occurs when blood vessels between the dura mater and the skull rupture, causing bleeding in the space between the skull and the meninges. Head trauma is the most common cause of epidural hemorrhages.

The class distribution of the five types of hemorrhages is shown in FIGURE 4. The figure illustrates that the dataset is highly imbalanced. The number of samples of epidural hemorrhage is very low compared to other classes. Subdural hemorrhage has the highest number of samples. The number of samples of intraparenchymal and subarachnoid hemorrhage is also very high.

B. IMAGE PRE-PROCESSING

For each slice of the CT scan, we performed several preprocessing operations that included windowing, resizing, normalizing, and zero centering.

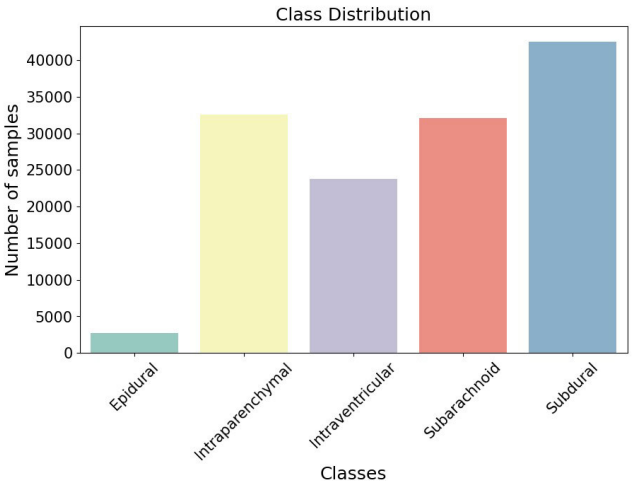


FIGURE 4. Class distribution of images in the training set.

Hounsfield units (HU) are used in CT scans to measure the densities of objects. The amount of X-rays absorbed by an object determines its density. The range of HU values typically ranges from -1000 to $+1000$. Lower values indicate lower density i.e., less X-rays are absorbed, whereas higher values indicate higher density i.e., more X-rays are absorbed. Different tissues and structures in the body have different densities, which can be measured using HU. For example, bones have high density, and depending on the type of bone, the HU value varies from $+400$ to $+1000$. The denser the bone tissue, the brighter it will appear on the CT image. Air has a low density and has HU value in the range

of −1000 to −700. CT images show darker areas where HU values are low. Hounsfield units are useful for identifying and characterizing different structures in the body on CT scans and highlighting abnormalities or subtle differences in tissue density. Different tissues and structures in the body have different densities, which can be measured using HU. The HU number distinguishes between air, water, lung, kidney, blood, and muscles. To display specific ranges of HU values, windowing can be used.

Windowing is a technique used in medical imaging in which the brightness and contrast of an image is adjusted to enhance specific features of interest. This is done by mapping a range of pixel values into a new range. The pixel values mapped to a new range can then be visualized on a screen to visualize the specific structure of interest. A window level sets the midpoint of the display range, while a window width sets the range of values. By utilizing these two parameters, doctors can highlight or omit certain structural features in CT scans, assisting them in identifying abnormalities and making diagnoses.

Several windowing techniques are commonly used in brain CT scans, including the bone window, gray-white differentiation window, brain matter window, and subdural (blood) window. Regions whose density is closer to that of bone can be found using the bone window. Skull lesions can be located using this windowing. The brain window can be used to focus on the regions whose average density is similar to that of the brain's soft tissue. These regions in the brain comprise air and fluid-containing areas. To differentiate between gray and white matter of the brain, the gray-white differentiation window is used. The presence of a tumor or brain swelling can be identified through this method. A subdural window is a specific type of windowing technique that can be used to detect subdural hemorrhages.

If the intensity of the pixel (i, j) in the k_{th} channel of the CT scan is ζ_{ij}^{k} and the maximum and minimum range of the specific window are ζ_{max}^{k} and ζ_{min}^{k} respectively, the updated value of the pixel after windowing operation for the k_{th} channel is described by the following:

$$\zeta_{ij}^{k'} = \zeta_{ij}^{k} \cdot \begin{Bmatrix} 1 & \text{if} \quad \zeta_{\min}^{k} \leq \zeta_{ij}^{k} \leq \zeta_{\max}^{k} \\ 0 & \text{if} \quad \zeta_{ij}^{k} < \zeta_{\min}^{k} \text{ or } \zeta_{ij}^{k} > \zeta_{\max}^{k} \end{Bmatrix} \tag{1}$$

To get more information about different density region, we applied 3 windowing (brain, soft tissue, and subdural) operation to the original image and merged them into a 3 channel RGB image:

$$\mathcal{X}_{ij} = \left[\zeta_{ij}^{0'}, \zeta_{ij}^{1'}, \zeta_{ij}^{2'}\right] \tag{2}$$

We aim to use different windows as different channels in the input image. FIGURE 5 depicts CT scans with different windowing. It is observed from FIGURE 5 that different regions of the scan are visible at different windows, which is very useful. Bone windowing can be used to identify bone fractures, whereas brain windowing can be used to identify brain hemorrhages. We generated three single-channel images in the windowing process according to the HU of the brain, soft tissue, and subdural range. Then we stacked these three images into an RGB image. Thus, we could capture the necessary features of our desired range (brain, soft tissue, and subdural). The bone window of the CT scan is not relevant to us, so we have not included it.

After that, we resized or downsampled the RGB CT images. Downsampling is required since we have limited Video Random Access Memory (VRAM) in the Graphics Processing Unit (GPU). This VRAM stores all the images during training. If the images are large, VRAM cannot accommodate all the images in a batch, and the training stops. The RGB images were resized to (224, 224, 3) from (512, 512, 3).

After downsampling the images, normalization and zero-centering were performed. Normalization involves rescaling the intensity values of an image to a specific range. Several types of normalization are found in the literature that include min-max normalization, z-score normalization etc. In min-max normalization, the values are scaled to range between 0 and 1. Min-max normalization can be calculated as follows:

$$X_{mm\_norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3}$$

here, X, X_{min}, and X_{max} represent the original, minimum, and maximum values in the dataset, respectively. The normalized value of X within the range [0, 1] is denoted as X_{mm_norm}.

In z-score normalization, the values are scaled in such a way that they achieve a mean of 0 and standard deviation of 1. It can be calculated as follows:

$$x_{z\_norm} = \frac{x - \mu}{\sigma} \tag{4}$$

here, x represents the original data, while μ and σ represent the mean and standard deviation of the data, respectively. The normalized data having mean of 0 and standard deviation of 1 is denoted as x_{z_norm}.

Zero centering is a preprocessing technique that involves adjusting the mean of the data to zero. In order to achieve this, the mean value of the data is deducted from each data point.

$$x_{zero} = x - \bar{x} \tag{5}$$

here, x represents the original data, $\bar{x}$ represents the mean value of x, and x_{zero} represents the new data with the mean centred around zero.

C. MODEL DEVELOPMENT

1) CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is a specialized neural network architecture specifically tailored to handle data that exhibits a grid-like pattern, such as images [22]. CNNs apply a type of operation called convolution to extract significant features from an image, which helps in detecting edges and other relevant information. This operation performs well in extracting important information from images.

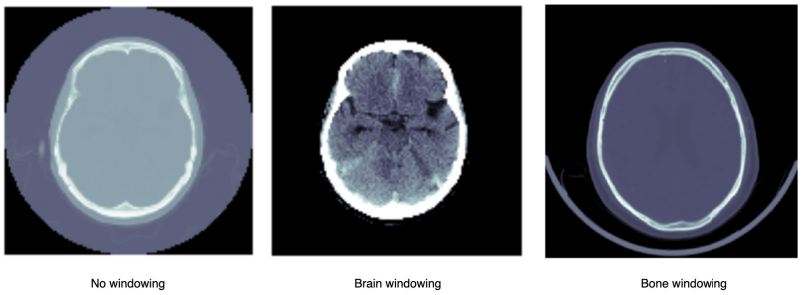


FIGURE 5. CT scan at different windowing.

The convolution operation involves taking a small matrix of numbers, known as a filter or kernel, and sliding it across the image, one pixel at a time, to create a modified output image. For each position, the values in the filter are multiplied by the corresponding pixels in the image, and the resulting products are summed to generate an output. An output image is created by repeating this process for every pixel of the image.

The convolution operation on an image is as follows:

$$X'(i,j) = \sum_m \sum_n X(i+m, j+n)K(m,n) \qquad (6)$$

here, the input image is represented by the matrix X and the filter is represented by the matrix K. The resulting output is represented by matrix the X'. The $*$ symbol represents the convolution operation. i and j are the pixel indices in the output image, and m and n are the kernel indices.

A visualization of the convolution operation is shown in FIGURE 6. Here, the input is a 5×5 grid, and the filter has a dimension of 3×3. The filter goes through the input and generates the output by performing the convolution operation. The output dimension is determined by both the input and the filter dimensions.

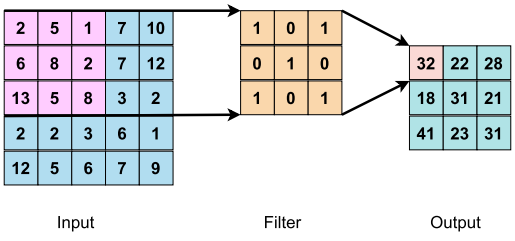


FIGURE 6. Visualization of convolution operation on an Image.

Convolutional layers are often succeeded by a pooling layer to decrease the spatial size and summarize the features of the outputs. The pooling layer serves a dual purpose: it not only decreases the dimensionality of the feature maps but also enhances the receptive field of the subsequent convolutional layers. Several kinds of pooling operations are found in the literature. The most common types of pooling are maxpooling and average pooling. Max pooling assigns the output as the maximum value within each pooling region, while average pooling assigns the output as the average value within each pooling region.

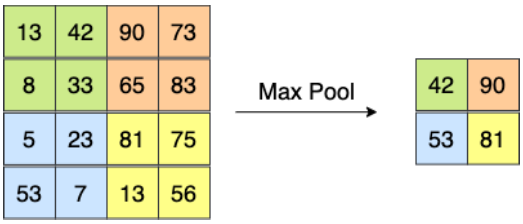


FIGURE 7. MaxPooling operation.

A visualization of the maxpooling operation is shown in FIGURE 7. It is observed that the maxpooling operation has effectively reduced the spatial dimensions of the feature map from 4×4 to 2×2 by taking the maximum value within each 2×2 region.

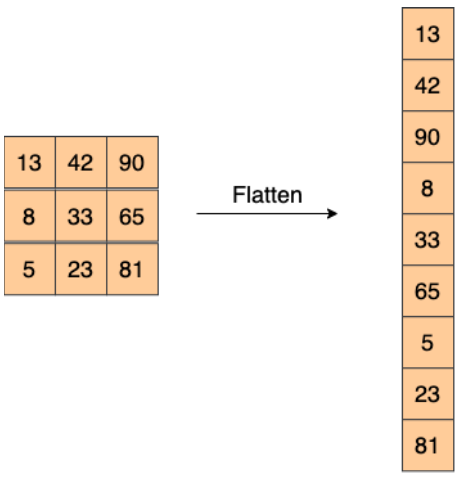


FIGURE 8. Flattening operation.

The convolution and pooling layer make up the feature extraction part of the CNN. Classification is done using a Fully Connected Network (FCN) that uses the features extracted by these layers as input. Prior to entering the FCN, the features are first flattened. A visualization of the flattening operation is shown in FIGURE 8.

Modern CNNs stack multiple convolution layers on top of each other to extract deep features. Subsequently, the final layer is flattened and fed into a FCN. A simple CNN for the identification of five subtypes of intracranial hemorrhage is shown in FIGURE 9. The network takes an image as input.

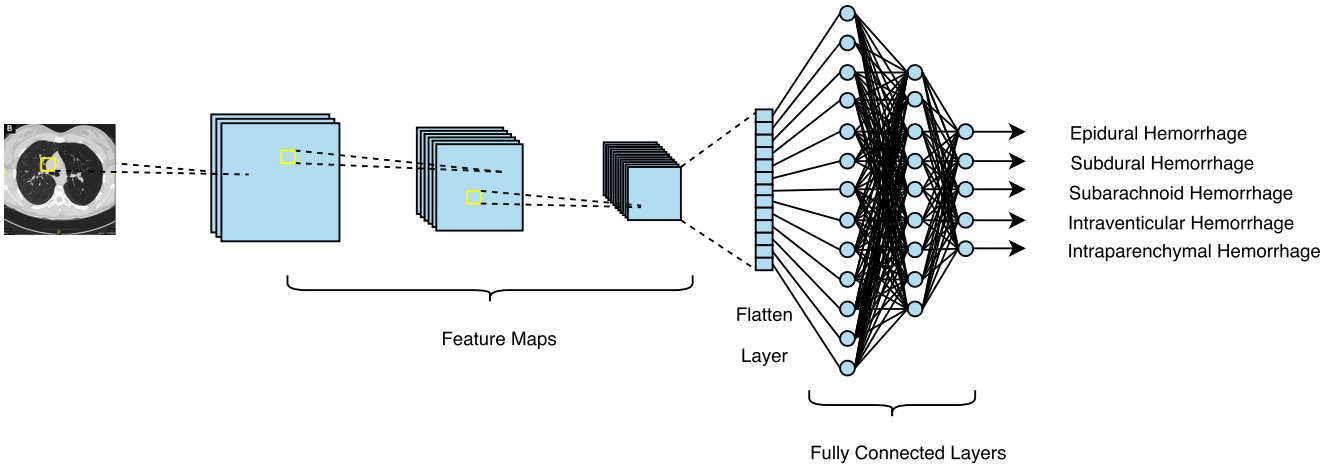


FIGURE 9. How convolutional neural network works.

Then it goes into three successive stacks of convolutional and pooling layers, resulting in downsampled feature maps at each stage. The feature maps obtained after the final convolutional layer are subsequently flattened and processed by a fully connected network consisting of three layers. These layers perform the task of classification. In the final layer, there are five nodes. Each node represents a subtype of intracranial hemorrhage.

2) DENSELY CONNECTED CONVOLUTIONAL NETWORK (DenseNet)

DenseNet [4], was introduced to address a common issue of CNN. There have been networks with various numbers of layers since the inception of AlexNet, some reaching 100 layers or more. Adding so many layers to a neural network created a new problem known as the vanishing gradient problem. Since there are many intermediate layers between the input and output layers, information from the input layer disappears before reaching the output layer. DenseNet addresses this problem by connecting multiple layers using skip connections instead of propagating them linearly. The network also reduces its complexity by ensuring feature reuse, and hence reduces the time taken to train.

For any traditional CNN, the output of i_{th} layer, x_i is obtained by applying nonlinear transformation H_i to the output of the previous layer x_{i-1}:

$$x_i = H_i(x_{i-1}) \tag{7}$$

ResNet [23] uses the concept of skip connection. In the case of ResNet, the output of the i_{th} layer, x_i is obtained by adding the output of a previous layer x_j to the result of the nonlinear transformation applied to the output of the previous layer x_{i-1}.

$$x_i = H_i(x_{i-1}) + x_j \tag{8}$$

DenseNet [4] introduces direct connections between each layer to each subsequent layer to improve the flow of information. The output of the i_{th} layer is:

$$x_i = H_i([x_0, x_1 \ldots, x_{i-1}]) \tag{9}$$

The architecture of the DenseNet model is shown in FIGURE 10. Unlike typical CNNs, where convolution and pooling layers are stacked one after another to form the entire network, DenseNet has four blocks with multiple layers in each block. In each block, there are several layers, and the features' dimension remains the same throughout all the layers. All the layers are stacked as a pair of 1×1 convolution layer followed by a 3×3 convolution layer. Dense blocks 1, 2, 3 and 4 contain 6, 12, 24 and 16 such pairs respectively. The 1×1 convolution layer acts as a bottleneck layer, which results in faster computation and performance gain.

The transition layer connects each dense block to the next block through a composite function of convolution and pooling. There are three transition layers connecting the four dense blocks. The transition layers consist of a 1×1 convolution filter and a 2×2 average pooling with a stride of 2. As a result of the transition layer, the number of channels is reduced, resulting in downsampling. Following the fourth dense block, a global average pooling operation is performed before transitioning to a fully connected layer.

The pre-activation value of a neuron in a neural network can be calculated as:

$$Z_i = \sum_{j=1}^{n} w_j x_j + b_i \tag{10}$$

here, x_j represents the input to the neuron i from the j-th input neuron, w_j represents the weight associated with the input x_j, b_i represents the bias term for neuron i.

In DenseNet, ReLU activation is applied to the output neurons from each layer except the final layer. The ReLU activation function takes an input value Z_i and returns the maximum between that value and 0. That means, if Z_i is

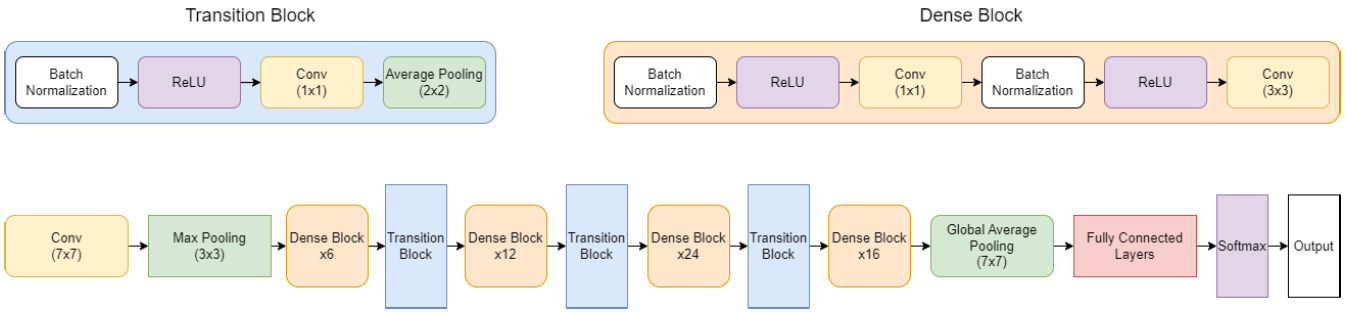


FIGURE 10. Architecture of the DenseNet121 Model.

positive it returns Z_i, if its negative it returns 0. It can be calculated as follows:

$$ReLU(Z_i) = \begin{cases} Z_i & \text{if } Z_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

In this work, we used sigmoid activation function in the final layer since it is a multi-label classification problem and more than one class can be activated for the same image input. For example, a person who has subdural hemmorhage might also be a victim of epidural hemorrhage. Given an input value Z_i, the sigmoid activation function $\sigma(Z_i)$ transforms the value to a range between 0 and 1. It can be calculated as follows:

$$\sigma(Z_i) = \frac{1}{1 + e^{-Z_i}} \tag{12}$$

We used the Binary Cross Entropy (BCE) loss function. If the predicted output is $\hat{Y}$ and original output is Y, the BCE loss will be,

$$L_{BCE}(Y, \hat{Y}) = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i)) \tag{13}$$

The loss is propagated backward to update the weights of all the layers to reduce the future loss. The model parameters are updated as,

$$w_i' = w_i - \alpha \frac{\partial L_{BCE}}{\partial w_i} \tag{14}$$

here, w_i is the current value of the weight, w_i' is the updated value of the weight w_i after applying the weight update rule, α is the learning rate which controls the magnitude of the weight update, and $\frac{\partial L_{BCE}}{\partial w_i}$ is the partial derivative of the binary cross-entropy loss function L_{BCE} with respect to the weight w_i. In order to perform classification, a classification threshold of 0.5 is used. If the output of a neuron in the final layer of the network is above 0.5, it is identified as positive.

The output layer of our neural network has 5 neurons, each corresponding to a specific type of hemorrhage. Since sigmoid activation function is used in the neurons of the final layer, the output values will range between 0 and 1. A higher output value of a neuron indicates more likelihood of presence of the corresponding hemorrhage in the image. A classification threshold of 0.5 is used to make a final classification decision. If the output value of a neuron is above 0.5, it is identified as positive for the corresponding hemorrhage. Otherwise, it is identified as negative.

$$\mathcal{Y}_i = \begin{cases} 1 & \text{if } Y_i \geq .5 \\ 0 & \text{if } Y_i < .5 \end{cases} \tag{15}$$

D. MODEL EVALUATION

The evaluation metric is a score that measures how well a machine learning model performs [24]. It plays a crucial role in determining the effectiveness of models and is used to compare different models and choose the best one. There is a wide range of evaluation metrics depending on the nature of the problem. The accuracy, precision, recall, and F1 score are a few of the most common metrics used for classification tasks [25]. In regression tasks, errors are typically evaluated as mean squared error, mean absolute error, and root mean squared error [25]. Different evaluation metrics focus on different aspects of model performance, so choosing the right metric for a problem is essential. In a balanced dataset, accuracy may be the best metric, but precision and recall provide more helpful information in an imbalanced dataset. In this study, multiple evaluation metrics, including accuracy, precision, recall, F1-score, and specificity, were used to evaluate the model to avoid bias due to any single evaluation metric's influence.

Accuracy: Accuracy represents the percentage of correctly classified instances (both positive and negative) out of the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision represents the percentage of correctly predicted positive instances out of all predicted positive instances.

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

Recall or True Positive Rate: Recall represents the percentage of correctly predicted positive instances out of all true

positive instances.

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

Specificity or True Negative Rate: Specificity represents the percentage of correctly predicted negative instances out of all true negative instances.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

F1 Score: F1-Score takes into account both the ability to correctly identify positive instances (precision) as well as the ability to capture all the actual positive instances (recall).

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

E. BAYESIAN OPTIMIZATION BASED FRAMEWORK FOR INTRACRANIAL HEMORRHAGE CLASSIFICATION

1) BAYESIAN OPTIMIZATION

Various optimization strategies can be used to optimize hyperparameters for deep learning [26]. The most common and easiest way is to use grid search or random search. In grid search, all possible combinations of hyperparameters are explored whereas, in random search, random configurations of hyperparameters are explored. Both of these methods work well when the search space is small. When the search space becomes large, these approaches become inefficient since a huge number of sample points have to be evaluated to get a good solution, which is very expensive.

A probabilistic model is employed in Bayesian Optimization (BO) [27] to guide the search within the search space. This model helps to identify promising areas of the search space, allowing the search to focus on the most promising configurations. By concentrating on the most promising configurations, BO offers the advantage of reducing the number of evaluations required to discover the best configuration, unlike the grid or random search methods.

Additionally, BO uses previous evaluations to guide the search. By incorporating information from previous evaluations, the search can explore the search space more effectively and find the best configuration more quickly. This is because previous evaluations provide information about which configurations are most likely to lead to good results, allowing the search to focus on those configurations. By using both a probabilistic model and information from previous evaluations, BO is a powerful technique for finding the best configuration of a model.

BO uses the concept of Bayes' theorem to perform the optimization [27], [28]. According to the Bayes' theorem [29]:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{16}$$

$P(A)$ represents the prior probability of event A before considering any evidence. The probability of observing evidence B is $P(B)$. $P(B|A)$, commonly referred to as the likelihood of event B, signifies the probability of evidence B occurring under the assumption that event A is true. $P(A|B)$, also known as the posterior probability of A, represents the probability of A being true if the evidence B is observed.

BO makes use of a prior belief to produce a posterior that closely relates to the objective function f. In order to get a good approximation of the real function, it takes samples from the hyperparameter space and then constructs a surrogate function. One such surrogate function is Gaussian Process (GP) [27]. Evaluation of GP posteriors is inexpensive and is useful for discovering areas in a search space where sampling is likely to improve results.

The distribution over functions that map input points to output values is modeled in GP regression. Given a set of input points $\mathbf{X} = x_1, x_2, \ldots, x_n$ and their corresponding function values $\mathbf{y} = y_1, y_2, \ldots, y_n$, we can use this data to make predictions about the output value at a new input point x_*. The GP posterior at x_* is a Gaussian distribution with mean μ_* and variance σ_*^2. In order to calculate the mean, function values at the training points are linearly combined. Covariances between the training and test points determine the weights used in the combination. The variance measures the uncertainty about the predicted function value at x_*. It is affected by the uncertainties in the estimates of the values of the function at the training points.

$$\mu_* = m(x_*) + \mathbf{k}_*^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{X})), \tag{17}$$

$$\sigma_*^2 = k(x_*, x_*) - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*, \tag{18}$$

where $k(x_*,x_*)$ is the self covariance of the point x_*, the vector $\mathbf{k}_*$ contains the covariances between the test point x_* and the training points y, matrix $\mathbf{K}$ denotes the covariances among the training points. and $\mathbf{m}(\mathbf{X})$ is the vector of mean values at the training points.

The acquisition function determines the next point to be sampled in the search space utilizing the GP surrogate function. It effectively balances exploration and exploitation during the optimization process. During exploration, the acquisition function prioritizes sampling points in the search space where the surrogate function exhibits higher variance. In contract, during exploitation, it samples from promising regions of the search space. Several acquisition functions are found in the literature that includes Upper Confidence Bound (UCB), Probability of Improvement (PI), Expected Improvement (EI) etc [27].

EI [27] is commonly used in BO to select the next point to sample. The equation for EI is:

$$EI(x) = \begin{cases} (y_{min} - \mu(x) - \zeta)\Phi(Z) + \sigma(x)\phi(Z) & \sigma(x) > 0 \\ 0 & \sigma(x) = 0 \end{cases} \tag{19}$$

here, y_{min} stands for the minimum function value observed thus far. $\mu(x)$ and $\sigma(x)$ correspond to the mean and standard deviation of the GP posterior at point x. Φ and ϕ are the standard normal cumulative distribution function and probability density function. The exploration parameter ζ is a constant, which purpose is to encourage the optimizer to sample points

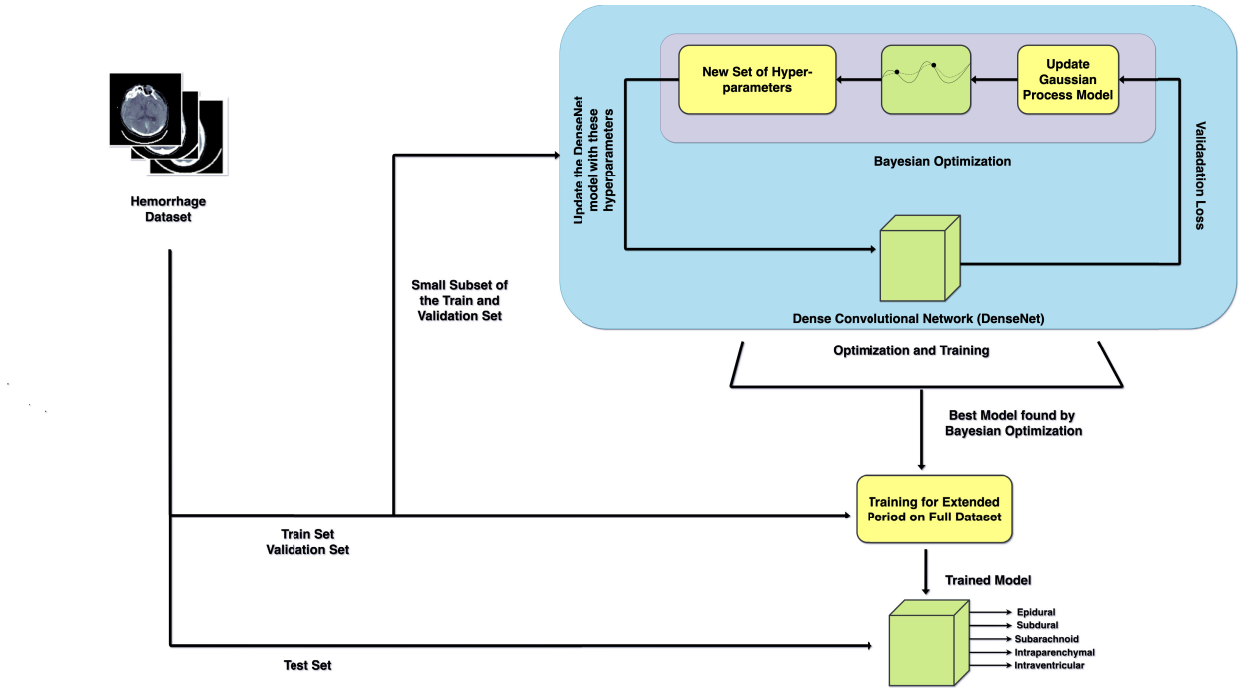


FIGURE 11. Bayesian optimization based framework for hemorrhage classification.

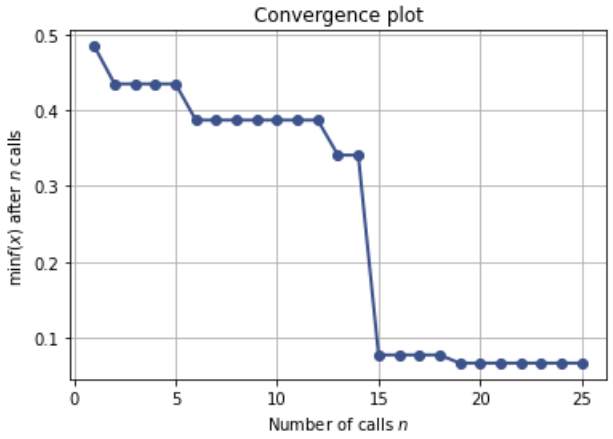


FIGURE 12. Convergence plot.

with higher variance rather than just high mean. A higher value of ζ means that the acquisition function will explore the search space more. Z is a standardized normal random variable.

$$Z = \begin{cases} \dfrac{y_{min} - \mu(x)}{\sigma(x)} & \sigma(x) > 0 \\ 0 & \sigma(x) = 0 \end{cases} \tag{20}$$

The EI measures the expected improvement over the current minimum value, taking into account the uncertainty in the GP model. Intuitively, it selects points with a high probability of improving over the current best point, while also considering the uncertainty in the model.

Validation loss is considered the objective function in our framework. The number of neurons, learning rate, and the optimizers are the hyperparameters that we tuned.

2) THE FRAMEWORK

We have showed our Bayesian Optimization based framework for hemorrhage classification in FIGURE 11.

The first step involves splitting the labeled hemorrhage dataset into three separate sets: training, validation, and test sets adopting an 80:10:10 split ratio. For the optimization and training step, the training and validation sets are employed, whereas the evaluation of the best model is conducted exclusively on the test set. We have optimized the DenseNet architecture in our work. Three hyperparameters were optimized using BO - learning rate, number of dense nodes, and optimizer. During the first iteration of optimization, a predefined hyperparameter configuration is used. Then the DenseNet architecture is trained on this predefined hyperparameter. After training the DenseNet with this hyperparameters, a validation loss is obtained. This validation loss is used to update the GP model which acts as the surrogate function. Then a new set of hyperparameter values is proposed by the EI acquisition function. The DenseNet architecture is trained again with these new set of hyperparameter values. The optimization and training process iterates until the maximum number of iterations is reached. Once this process is completed, the best model is determined by evaluating the validation loss. This model is trained on the full dataset for an extended period. Subsequently, the performance of this model is assessed using the test set.

TABLE 1. Results of each iteration of Bayesian optimization applied to the DenseNet architecture.

Iteration No.	Objective	Learning Rate	No. of dense nodes	Optimizer
1	0.484	0.001	2	Adam
2	0.434	7.875×10^{-5}	75	RMSprop
3	0.548	0.003	88	Adam
4	0.477	0.001	41	RMSprop
5	0.797	2.825×10^{-5}	103	SGD
6	0.387	0.0001	134	Adam
7	0.458	1.235×10^{-5}	132	RMSprop
8	0.472	0.0004	21	RMSprop
9	0.441	0.0021	88	Adam
10	0.542	0.0030	18	RMSprop
11	1.088	1.572×10^{-5}	10	SGD
12	0.639	10^{-5}	150	Adam
13	0.341	0.0006	127	SGD
14	0.377	0.0034	94	RMSprop
15	0.077	7.809×10^{-5}	10	RMSprop
16	0.149	0.0002	122	RMSprop
17	0.093	0.00015	126	RMSprop
18	0.312	0.0008	56	SGD
19	0.067	2.637×10^{-5}	131	RMSprop
20	0.362	0.0025	79	RMSprop
21	0.343	0.0014	88	RMSprop
22	0.345	0.0018	18	RMSprop
23	0.107	0.01	105	SGD
24	0.113	0.0099	10	SGD
25	0.133	0.01	4	SGD

IV. EXPERIMENTAL RESULTS

The DenseNet architecture was used to perform the classification. The optimization of DenseNet's hyperparameters was accomplished using BO. Initially, the CT scans were converted to Hounsfield Unit (HU) to distinguish between different organs. These are also resized and normalized for training the deep neural networks.

A. HYPERPARAMETER OPTIMIZATION OF DENSELY CONNECTED CONVOLUTIONAL NETWORK (DenseNet)

We utilized BO to identify the optimal hyperparameters for the DenseNet in order to accurately detect intracranial hemorrhage from CT scans. The hyperparameters subjected to optimization included the learning rate, number of dense nodes, and optimizer. To begin, we defined the search space for hyperparameter optimization. The learning rate search space ranged from 10^{-5} to 10^{-2}, while the number of dense nodes varied from 2 to 150. Throughout the optimization process, we explored three different optimizers: Adam, Stochastic Gradient Descent (SGD), and Root Mean Squared Propagation (RMSProp).

The convergence plot in Figure 12 illustrates the optimization progress of the DenseNet. On the x-axis, the graph displays the number of iterations or trials, while the minimum value of the validation loss is shown on the y-axis. From FIGURE 12, it is observed that the validation loss first decreased in the 2nd iteration of the optimization. From 2nd to 5th iteration, no improvement in validation loss was observed. Later the validation loss decreased in the 6th, 13th, 15th, and 19th iterations. Although there was a significant improvement in validation loss from the 14th to 15th iteration, the improvement in validation loss from the 18th to 19th iteration is relatively minimal. The model was able to achieve a validation loss of less than 0.1.

TABLE 1 presents a summary of our findings on each iteration of Bayesian optimization applied to the DenseNet architecture. From this table, we can see how our objective function has changed with each iteration and what the hyperparameters were. Each table row shows the exact hyperparameters used in a trial and the resultant validation loss. It is apparent from the table that, the model with the lowest validation loss was obtained during the 19th iteration of optimization. The validation loss at that iteration was 0.067 and the hyperparameters were 2.637×10^{-5}, RMSprop, and 131 which were the learning rate, optimizer, and the number of dense nodes respectively.

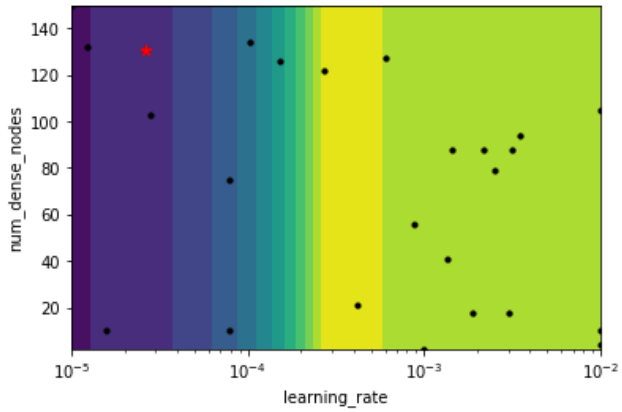
FIGURE 13 shows three Partial Depedence Plots (PDPs) with the three optimized hyperparameters - learning rate, optimizers, and the number of dense nodes. The PDPs are useful for understanding how the change of hyperparameters influences the objective function or, in this scenario, the validation loss. The yellow areas on the plots represent regions with low validation losses, and the blue areas on the plots represent the regions where the validation losses are high. As we experience low validation losses in the yellow regions, they are particularly interesting to us. A point in the plot represents a particular set of hyperparameters. These are represented as black dots. In the figure, the red star indicates the location of the optimal hyperparameter value.

We can see from the PDP that when the learning rate changed, the number of dense nodes did not significantly alter the validation loss 10^{-3} to 10^{-4} seem to be the optimum range for learning rate. SGD performed best when the learning rate was close to the 10^{-2} region, according to the PDP with optimizers and learning rate. With RMSprop and a learning rate under 10^{-3}, the best validation loss was achieved.

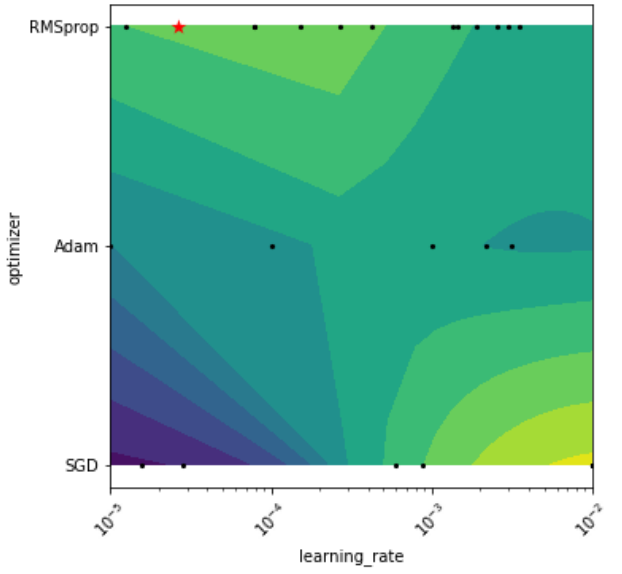
B. TRAIN & TEST RESULTS OF THE OPTIMIZED MODEL

During optimization, we used a smaller subset of the original dataset to find the most suitable model for diagnosing intracranial hemorrhage. Bayesian optimization was used to determine the best model on the smaller dataset, which was then trained on the original larger dataset. From TABLE 1, we see that the best model or the best set of hyperparameters was found during iteration 19 of the optimization process. The best model at this stage has a learning rate of 2.637×10^{-5}, and the number of dense nodes was 131. The RMSprop optimizer was used to train this model. We again trained the model found at this iteration on the original larger dataset to check its performance.

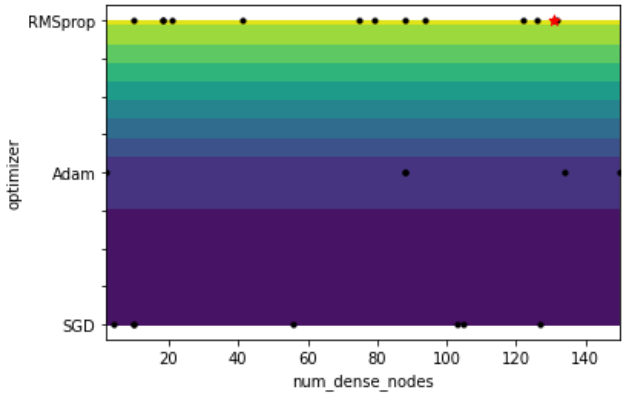
FIGURE 14 depicts the relationship between the number of epochs and the metrices used to infer the model. The model is trained for 50 epochs. Change in loss, accuracy and F1-score are shown in the figures. The blue curves represent



(a) PDP with the number of dense nodes and learning rate



(b) PDP with learning rate and optimizer



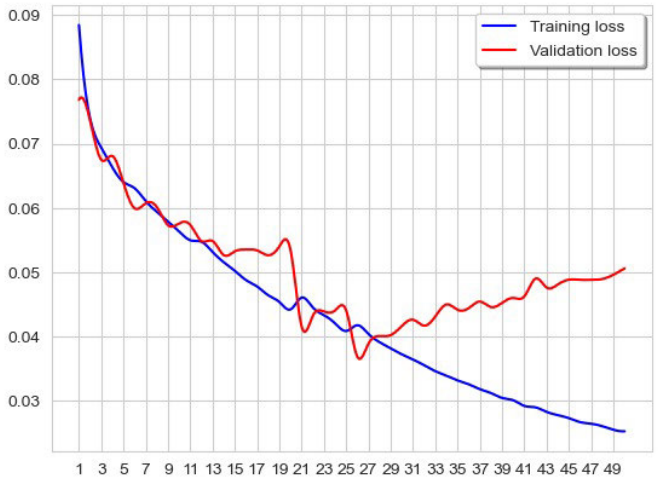
(c) PDP with number of dense nodes and optimizer

FIGURE 13. **Partial dependence plots.**

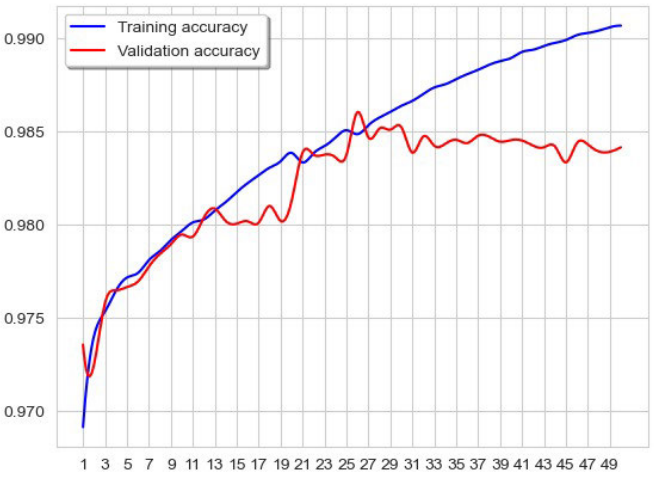
the training metrices (training loss, accuracy, and F1-score), whereas the red curves represent the validation metrices (validation loss, accuracy, and F1-score). From FIGURE 14a, it is observed that the training loss began to decrease from the very first epoch. The training loss increased slightly from epoch 19 to 20 and epoch 25 to 26. It decreased during the remainder of the epochs. The dynamics of validation loss were inconsistent. It often went up and down. Quite similar phenomena were observed for accuracy and F1-score curves in FIGURE 14b and FIGURE 14c. The training accuracy and F1-score began to rise from the first iteration. The validation accuracy and F1-score did not follow any straight pattern.

Upon completing the training, the model's classification performance on the test set was evaluated and is presented in TABLE 2. The table shows the model's overall performance and performance in each class. Accuracy, precision, recall, specificity, F1-score, and Mathew correlation coefficient are reported.

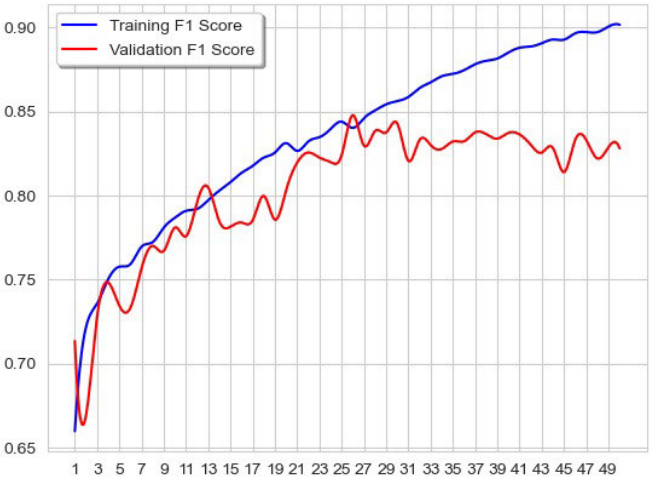
The evaluation of the model based on individual class accuracies in a multi-class problem is not recommended due to the calculation method used. In this approach, the accuracy of a specific class is determined by considering it as the positive class and all other classes as the negative class. Consequently, if a class has a smaller sample size, the model is more likely to achieve a higher individual class accuracy for the smaller class. Conversely, the model may achieve a lower accuracy for the majority class. This exact phenomenon was observed in the current study. For instance, Epidural hemorrhage, which had the lowest number of samples, achieved a very high accuracy of 99.72%. However, a closer examination of the precision and recall values reveals their relatively low percentages of 75.37% and 59.08%, respectively. Precision represents out of the total predicted positive cases, what percentage were actually positive. Here, Intraparenchymal hemorrhage exhibits a precision of 90.20%, indicating that 90.20% of the cases classified as intraparenchymal hemorrhage were indeed true positive instances, while the remaining 9.80% were misclassified as intraparenchymal hemorrhage, but actually belong to other categories. Similarly, subdural hemorrhage demonstrates a precision of 66.44%, indicating that only 66.44% of the cases classified as subdural hemorrhages were accurately identified, whereas the rest were assigned incorrectly. The model achieved an overall precision of 85.45%. Recall, also known as sensitivity, represents what percentage of the actual positive cases were actually captured by the model. It is observed from the table that, the recall for intraventricular hemorrhage is 82.20%, indicating that the model accurately predicted 82.20% of all the intraventricular hemorrhage cases present in the test set. Similarly, the recall for epidural hemorrhage is 59.08%, meaning that the model correctly identified 59.08% of all epidural hemorrhage cases. It is evident that the model did not perform well in predicting epidural hemorrhage cases, despite having a high overall accuracy. One plausible reason behind this discrepancy could be the lack of sufficient samples for the epidural hemorrhage class in the dataset. As epidural hemorrhages are relatively rare compared to other types of hemorrhages, there may not be enough training data for the model to learn and generalize effectively for this specific class. The model achieved high specificity or true negative rate for all classes. It represents the percentage of



(a) Loss vs. Epochs



(b) Accuracy vs. Epochs



(c) F1-score vs. Epochs

FIGURE 14. Training dynamics.

TABLE 2. Classification performance for the proposed model.

Class	ACC	PRE	REC	SPE	F1	MCC
Any	0.9632	0.8940	0.8432	0.98327	0.86786	0.8470
Epidural	0.9972	0.7537	0.5908	0.99911	0.66236	0.6659
Intraparenchymal	0.9854	0.9020	0.7822	0.99569	0.83787	0.8326
Intraventricular	0.9897	0.8786	0.8220	0.99585	0.84935	0.8445
Subarachnoid	0.9751	0.7390	0.7120	0.98784	0.72525	0.7124
Subdural	0.9618	0.6644	0.8035	0.9725	0.7273	0.7107
Overall	0.9802	0.8545	0.7603	0.9928	0.7899	0.7714

correctly predicted negative instances out of all actual negative instances. The F1-score, a metric derived from precision and recall, proves to be a valuable measure for evaluating the model's performance in this context. Table 2 highlights that intraventricular hemorrhage achieved the highest F1-score, while epidural hemorrhage obtained the lowest F1-score.

V. CONCLUSION

In this paper, we developed a deep learning-based CAD system for intracranial hemorrhage classification from CT scan of brain. We used windowing method for data preprocessing and optimized the DenseNet architecture hyperparameters using Bayesian optimization. Since the dataset is quite large, we first optimized the DenseNet architecture for a smaller subset of the dataset and obtained the best performing model. Then we retrained the best model on the larger dataset. According to our experimental results, the proposed model demonstrates the potential for accurate detection of hemorrhage from CT scans. Although our proposed method has some limitations, including the limited number of CT scans containing epidural hemorrhages and the lack of patient medical histories, it also has some advantages. By making

accurate and reliable diagnoses, it can assist doctors and improve patient outcomes. Our future work will involve testing the model's efficacy in clinical settings and refining it based on feedback from medical professionals. Our aim is to gain acceptance of deep learning based models within the medical community, with the potential to positively impact decision-making, patient care, and outcomes.

ACKNOWLEDGMENT

The authors would like to acknowledge Centennial Research Grant (CRG), University of Dhaka, Dhaka, Bangladesh, for funding this project. They would also like to acknowledge Raiyaan Abdullah, Raihan Kabir, and Maria Mim for their support in this project.

REFERENCES

[1] A. Majumdar, L. Brattain, B. Telfer, C. Farris, and J. Scalera, "Detecting intracranial hemorrhage with deep learning," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 583–587.

[2] A. Tharek, A. S. Muda, A. Baseri Hudi, and A. Baseri Hudin, "Intracranial hemorrhage detection in CT scan using deep learning," *Asian J. Med. Technol.*, vol. 2, no. 1, pp. 1–18, Jan. 2022.

[3] Ö. F. Ertuğrul and M. F. Akıl, "Detecting hemorrhage types and bounding box of hemorrhage by deep learning," *Biomed. Signal Process. Control*, vol. 71, Jan. 2022, Art. no. 103085.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[5] P. I. Frazier, "A tutorial on Bayesian optimization," 2018, *arXiv:1807.02811*.

[6] P. D. Chang, E. Kuoy, J. Grinband, B. D. Weinberg, M. Thompson, R. Homo, J. Chen, H. Abcede, M. Shafie, L. Sugrue, C. G. Filippi, M.-Y. Su, W. Yu, C. Hess, and D. Chow, "Hybrid 3D/2D convolutional neural network for hemorrhage evaluation on head CT," *Amer. J. Neuroradiology*, vol. 39, no. 9, pp. 1609–1616, Sep. 2018.

[7] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, and P. Warier, "Deep learning algorithms for detection of critical findings in head CT scans: A retrospective study," *Lancet*, vol. 392, no. 10162, pp. 2388–2396, Dec. 2018.

[8] H. Lee, S. Yune, M. Mansouri, M. Kim, S. H. Tajmir, C. E. Guerrier, S. A. Ebert, S. R. Pomerantz, J. M. Romero, S. Kamalian, R. G. Gonzalez, M. H. Lev, and S. Do, "An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets," *Nature Biomed. Eng.*, vol. 3, no. 3, pp. 173–182, Dec. 2019.

[9] M. Grewal, M. M. Srivastava, P. Kumar, and S. Varadarajan, "RADNet: Radiologist level accuracy using deep learning for hemorrhage detection in CT scans," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 281–284.

[10] N. T. Nguyen, D. Q. Tran, N. T. Nguyen, and H. Q. Nguyen, "A CNN-LSTM architecture for detection of intracranial hemorrhage on CT scans," 2020, *arXiv:2005.10992*.

[11] K. Jnawali, M. R. Arbabshirani, N. Rao, and A. A. Patel, "Deep 3D convolution neural network for CT brain hemorrhage classification," in *Proc. SPIE*, vol. 10575, 2018, pp. 307–313.

[12] W. Kuo, C. Häne, P. Mukherjee, J. Malik, and E. L. Yuh, "Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 45, pp. 22737–22745, Nov. 2019.

[13] R. Li, C. Xiao, Y. Huang, H. Hassan, and B. Huang, "Deep learning applications in computed tomography images for pulmonary nodule detection and diagnosis: A review," *Diagnostics*, vol. 12, no. 2, p. 298, Jan. 2022.

[14] J. D. L. Araújo, L. B. da Cruz, J. O. B. Diniz, J. L. Ferreira, A. C. Silva, A. C. de Paiva, and M. Gattass, "Liver segmentation from computed tomography images using cascade deep learning," *Comput. Biol. Med.*, vol. 140, Jan. 2022, Art. no. 105095.

[15] G. Salvaggio, G. Cutaia, A. Greco, M. Pace, L. Salvaggio, F. Vernuccio, R. Cannella, L. Algeri, L. Incorvaia, A. Stefano, M. Galia, G. Badalamenti, and A. Comelli, "Deep learning networks for automatic retroperitoneal sarcoma segmentation in computerized tomography," *Appl. Sci.*, vol. 12, no. 3, p. 1665, Feb. 2022.

[16] P. G. Mikhael, J. Wohlwend, A. Yala, L. Karstens, J. Xiang, A. K. Takigami, P. P. Bourgouin, P. Chan, S. Mrah, W. Amayri, Y.-H. Juan, C.-T. Yang, Y.-L. Wan, G. Lin, L. V. Sequist, F. J. Fintelmann, and R. Barzilay, "Sybil: A validated deep learning model to predict future lung cancer risk from a single low-dose chest computed tomography," *J. Clin. Oncol.*, vol. 41, no. 12, pp. 2191–2200, Apr. 2023.

[17] N. Alves, M. Schuurmans, G. Litjens, J. S. Bosma, J. Hermans, and H. Huisman, "Fully automatic deep learning framework for pancreatic ductal adenocarcinoma detection on computed tomography," *Cancers*, vol. 14, no. 2, p. 376, Jan. 2022.

[18] M. R. Islam and M. Nahiduzzaman, "Complex features extraction with deep learning model for the detection of COVID19 from CT scan images using ensemble based machine learning approach," *Exp. Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116554.

[19] S. V. Kogilavani, J. Prabhu, R. Sandhiya, M. S. Kumar, U. Subramaniam, A. Karthick, M. Muhibbullah, and S. B. S. Imam, "COVID-19 detection based on lung CT scan using deep learning techniques," *Comput. Math. Methods Med.*, vol. 2022, pp. 1–13, Feb. 2022.

[20] V. Ravi, H. Narasimhan, C. Chakraborty, and T. D. Pham, "Deep learning-based meta-classifier approach for COVID-19 classification using CT scan and chest X-ray images," *Multimedia Syst.*, vol. 28, no. 4, pp. 1401–1415, Aug. 2022.

[21] A. E. Flanders et al., "Construction of a machine learning dataset through collaboration: The RSNA 2019 brain CT hemorrhage challenge," *Radiol., Artif. Intell.*, vol. 2, no. 4, Jul. 2020, Art. no. e209002.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[24] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[25] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.

[26] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 1–10.

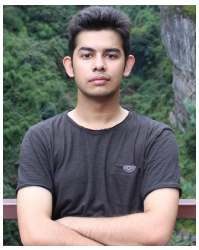
[27] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–12.

[28] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. D. Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[29] E. Eells, "Review: Bayes's theorem," *Oxford Mind*, vol. 113, no. 451, Jul. 2004.

SHIFAT E. ARMAN received the B.Sc. and M.Sc. degrees in robotics and mechatronics engineering from the University of Dhaka. He joined the University of Dhaka, as a Lecturer, in 2022. Prior to this, he worked as a Contractual Lecturer in BRAC University, a Research Assistant in the University of Dhaka, and a Visiting Research Assistant in Bangabandhu Sheikh Mujibur Rahman Agricultural University (BSMRAU). His research interests include neural architecture search (NAS), application of artificial intelligence in medicine, agriculture, finance, and GIS and remote sensing.

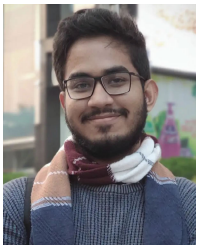
SAYED SAMINUR RAHMAN received the bachelor's degree in computer science from A. P. J. Abdul Kalam Technological University, India. He is currently a Software Engineer with health tech startup. His research interests include computer vision (CV), natural language processing (NLP), reinforcement learning (RL), and semi-supervised learning (SSL).



NILOY IRTISAM received the B.Sc. degree from the Department of Robotics and Mechatronics Engineering, University of Dhaka. He is currently a Research Assistant with the University of Dhaka. His current research interests include deep learning, computer vision, medical image analysis, medical robotics, and the application of artificial intelligence in robotics.

SHAMIM AHMED DEOWAN received the bachelor's degree in mechanical engineering from the Islamic University of Technology (IUT), the master's degree in sensor systems technology (mechatronics) from the Karlsruhe University of Applied Sciences, Germany, and the Ph.D. degree in chemical and materials engineering from the University of Calabria, Italy. He is currently an Associate Professor with the Department of Robotics and Mechatronics Engineering, University of Dhaka. He has published 20 international journal articles, seven books or book chapters, and 30 international conference papers. He holds an international patent in nanotechnology. His research interests include medical robotics, process engineering, and control and automation. He is a member of the Alumni Association of German Universities Bangladesh (AAGUB), the European Membrane Society (EMS), the American Society of Mechanical Engineers (ASME), and the Institution of Engineers, Bangladesh (IEB).

MD. ARIFUL ISLAM received the B.Sc. and M.Sc. degrees in electrical and electronic engineering from the University of Dhaka, in 2015 and 2017, respectively. He has been a Faculty Member with the Department of Robotics and Mechatronics Engineering, University of Dhaka. His research interests include machine learning, deep learning, fuzzy logic systems, medical robots, robotics in blue economy, and hybrid electric vehicle modeling.



SAADMAN SAKIB is currently pursuing the B.Sc. degree in robotics and mechatronics engineering with the University of Dhaka. As a Student with the University of Dhaka, he is dedicated to learning and exploring new technologies and techniques in the field of robotics and mechatronics engineering. His research interests include using deep learning to detect diseases from medical imaging, natural language processing, computational mathematics, reinforcement learning, and intelligent control of mechanical systems.

MEHEDI HASAN is currently a Senior Student with the Department of Robotics and Mechatronics Engineering, University of Dhaka. He has completed a wide range of courses, including artificial intelligence, robot vision, introduction to machine learning, introduction to robotics, microcontroller, and programmable logic controller. He is also deeply focused on working with microcontroller and the IoT-based projects. He is also working on soft robotics in his undergraduate project course. He was also an active Volunteer and a Mentor with the Bangladesh Robot Olympiad and the International Robot Olympiad. He has participated in a number of competitions held at different universities, such as ROBOLUTION 2018 held at MIST and the Intra-Department SOCCER BOT Competition. Recently, his team won Second Place in the Student Project Category in a Poster Competition at the Seminar on "Robotics in Bangladesh: Academia and Industry Initiatives." In this time, he has also attended three days of training on "Modular Production System (MPS) and CIROS Software."