

A Blockchain Based Framework for Stomach Abnormalities Recognition

Muhammad Attique Khan¹, Inzamam Mashood Nasir¹, Muhammad Sharif², Majed Alhaisoni³, Seifedine Kadry⁴, Syed Ahmad Chan Bukhari⁵ and Yunyoung Nam^{6,*}

¹Department of Computer Science, HITEC University, Taxila, 47040, Pakistan

²Department of Computer Science, COMSATS University Islamabad, Wah Campus, 47080, Pakistan

³College of Computer Science and Engineering, University of Ha'il, Ha'il, Saudi Arabia

⁴Department of Mathematics and Computer Science, Beirut Arab University, Lebanon

⁵Division of Computer Science, Mathematics and Science, Collins College of Professional Studies, St. John's University, New York, USA

⁶Department of Computer Science and Engineering, Soonchunhyang University, Asan, South Korea

*Corresponding Author: Yunyoung Nam. Email: ynam@sch.ac.kr

Received: 30 July 2020; Accepted: 19 October 2020

Abstract: Wireless Capsule Endoscopy (WCE) is an imaging technology, widely used in medical imaging for stomach infection recognition. However, a one patient procedure takes almost seven to eight minutes and approximately 57,000 frames are captured. The privacy of patients is very important and manual inspection is time consuming and costly. Therefore, an automated system for recognition of stomach infections from WCE frames is always needed. An existing block chain-based approach is employed in a convolutional neural network model to secure the network for accurate recognition of stomach infections such as ulcer and bleeding. Initially, images are normalized in fixed dimension and passed in pre-trained deep models. These architectures are modified at each layer, to make them safer and more secure. Each layer contains an extra block, which stores certain information to avoid possible tempering, modification attacks and layer deletions. Information is stored in multiple blocks, i.e., block attached to each layer, a ledger block attached with the network, and a cloud ledger block stored in the cloud storage. After that, features are extracted and fused using a Mode value-based approach and optimized using a Genetic Algorithm along with an entropy function. The Softmax classifier is applied at the end for final classification. Experiments are performed on a private collected dataset and achieve an accuracy of 96.8%. The statistical analysis and individual model comparison show the proposed method's authenticity.

Keywords: Stomach abnormalities; deep learning; blockchain; optimization; softmax



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Stomach infections are the most common nowadays. These infections include polyp, ulcer, and bleeding [1]. In 2019, about 22% of the adult population having gastric conditions in the United States. A total of 27,510 new stomach cancer cases are estimated, and 11,140 deaths are occurred due to these cases. In 2018, 160,820 deaths occurred due 319,160 new stomach cancer cases [2]. Stomach cancer is the third leading cause of death [3]. Colorectal or bowel cancer causes an average of 694,000 deaths in developing countries [4,5]. Esophageal cancer is the seventh common cancer disease in mature humans [6]. The early identification of gastric infections can improve the survival rate from 19% to 80% [7]. Therefore, humans' mortality rate can be decreased if infections are treated at an early stage [8].

Wireless Capsule Endoscopy (WCE) is widely utilized to recognize stomach infections. WCE is an imaging technique in the medical field. In WCE, a small camera captures the images of the gastrointestinal (GI) tract. The physicians detect and recognize the infections from these WCE images. However, this WCE technology has limitations such as an expert is required and consumption of time for infection recognition [9]. Several researchers developed the Computer-Aided Diagnostic (CAD) systems [10]. Using computer vision and image processing techniques along WCE can decrease the overall cost and time for infection recognition [11]. Many supervised Machine-Learning (ML) based CAD systems have developed and helped physicians identify the abnormalities in WCE images. Several researchers focused on deep feature extraction from pre-trained deep learning models such as AlexNet [12], ResNet [13], and VGG-16 [14]. Researchers extract different handcrafted features, including geometric, shape, color, and texture, along with deep CNN features to present their model. Color features descriptors have shown its importance in gastric infection recognition. Rajaei et al. [15] extract Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) features to classify the WCE images. The fundamental steps to develop these CAD systems for detecting abnormalities in WCE images are feature extraction, feature selection, and classification. Different methods which are utilized for the extraction features include color features [16], Scale Invariant Feature Transform (SIFT) [17], texture features, and many others. However, all extracted features may not be helpful in WCE images analysis. Therefore, the selection of best features is important, and several techniques have been developed, such as principal component analysis (PCA), linear discriminant analysis [18], and genetic algorithm (GA) [19]. A good feature selection method collects the best subset of features and reduces the classification time. The deep learning models save features, and these features are transformed in between several layers; therefore, it is a chance of disturbing few important features. The more recent, the entrance of blockchain in the machine learning, play a success for securing data. It is an absolute and encrypted database technology with a continuous growing list of blocks [20]. As a new technology, the researcher tries to implement this in several sectors and medicine is one of them. The importance of blockchain in the medical sector is needed when the number of patients are increasing with a high ratio. In this work, we are employing hash functions to secure the in-between CNN layer features. Our major contributions are as follows:

Blockchain technology in CNNs has been implemented to form a Secure CNN model called SecureCNN. The architecture of CNN is modified at each layer, to make it safer and more secure. For this purpose, we implemented an existing approach [21]. Each CNN model layer contains an extra block, which stores certain information to avoid possible tampering, modification attacks, and layer deletions. This information includes a) encrypted inputs and outputs of previous and current layers; b) public keys of all layers and private keys of neighboring layers, and c) weights of current and next layers. This information is stored in multiple blocks, i.e., a block attached

to each layer, a ledger block, attached with the network, and a cloud ledger block, stored at the cloud storage. The proposed CNNs are intelligent enough to detect any sort of tempering, either on the parameter level or on the network level, and perform restoration steps to avoid it. Later, we also optimize the features of CNN models using the Genetic Algorithm (GA) and passed optimal output in a Softmax classifier.

2 Related Work

This section explains the challenges of classifying and detecting ulcers in the past. Different Gastrointestinal (GI) tract infections include ulcer, polyp, esophagitis, and bleeding. The ulcer is the most common disease from all these infections [22]. Several techniques were proposed to effectively classify or detect anomalies in endoscopic images [23]. WCE images are used to classify abdominal infected gastrointestinal tracks using a novel automated method. A saliency estimated method called Color Features based Low-level and High-level Saliency (CFbLHS) is proposed to extract the frames from dataset videos. Transfer Learning (TL) has been an active technique in many domains to extract the deep features, which later proved vital in all computer-aided classification systems [24,25]. A pre-trained CNN network DenseNet has been used to extract the deep features using the TL technique and is fine-tuned using Kapur's entropy. Tsallis based entropy has been used to extract the 50% top-level features. The proposed method achieved an overall classification accuracy of 99.5% on the selected controlled WCE dataset [26]. An automated classification method using deep features is proposed, which combined densely connected models with non-local attention mechanisms. Contextual and relevant information is extracted by combining attention blocks with dense cascade blocks. The medical staff has annotated the used dataset twice per image on which the classification accuracy of 96.79% is achieved. Simultaneously, the ROC is noted at an average of 0.93 for the deep learning model [27]. A novel technique for automated localization and detection of gastrointestinal anomalies was proposed using the endoscopy images, in which training is carried out using the weakly annotated images. This training was performed using the image level semantic labels instead of pixel-level annotations, making it a cost-effective technique to analyze huge repositories of endoscopy. The proposed technique enabled the detector to detect the location's anomalies on the input image. The proposed technique's main steps included classifying the abnormal or normal image using, CNN model, detecting salient features using deep layers and deep saliency detection, and localizing the anomalies using Iterative Cluster Unification (ICU). The derived information from CNN was used to detect salient from the Pointwise Cross-Feature-Maps (PCFMs) in ICU. The proposed model achieved an average AUC of 88% on publicly available datasets [10]. The gastrointestinal disease was classified and detected using an automated diagnosis method on WCE images. In the proposed method, HSI color-space is utilized before the contour segmentation before implementing a saliency method in the YIQ color-space. The images are then fused using the proposed posterior probability maximization technique. The resultant images were then used to extract the GLCM, LBP, SVD features, which were serially fused to form a single feature vector. The proposed technique was tested on a private dataset containing 9,000 healthy, bleeding, and ulcer images. The proposed technique achieved an overall classification accuracy of 100% using the 10-fold cross-validation [18]. A pre-processing technique of edge enhancement was proposed to make images more appropriate to extract the features. Initially, the edges of the image were calculated using an edge extraction operator and then a brightness lookup table was utilized to calculate the edge map by applying addition or subtraction operation. The proposed technique achieved a classification accuracy of 95.55% on WCE images [28].

3 Proposed Methodology

Blockchain (BC) is a decentralized technology that uses distributed ledgers to record different transactions between users [29]. These users can either be people or systems or even algorithms. The transactions stored so that it remains permanent, could not be tempered, and can easily be verified upon a single request. Various crypto-currencies have used BC technology as basic building blocks. The Convolutional Neural Networks (CNNs) and BCs are not directly related. Still, these technologies can form a more secure structure in many real-time applications, i.e., machine security, health-care, and surveillance. The BCs are powerful and secure due to the following characteristics:

- Transitive Hash
- Encryption at every level
- Decentralization

The transitive hashes and encryption techniques disallow any algorithm's tempering, i.e., feature extraction, fusion, feature matching, and feature optimization. Transitive hashes will try to find any change at any level to trigger a notification highlighting an illegal change at a specific node or layer of algorithm and that specific node or layer can be restored to a previously valid state. The decentralized nature ensures that the whole algorithm is not stored on a single network, and no one, at any level, can deceive the algorithm. These properties can be used to propose a secure and safe CNN. Thus, blockchain stands as a favorite candidate for secure and safe CNN. The encryption can be carried out using symmetric or asymmetric key algorithms. Symmetric encryption algorithms use only one key to encrypt or decrypt the message, thus leaves a loophole. Anyone with the key can easily decrypt the message and change or delete it accordingly. In an asymmetric algorithm, two keys, i.e., public and private, are used to encrypt and decrypt the plain text [30]. The public key is openly distributed, while the private key is kept secretive. Anyone can encrypt receiver's message using their public key, but this encrypted message can only be decrypted using their private key. Asymmetric encryption improves the security level but reduces the overall speed of the process [31]. Asymmetric Encryption (AE) is applied during the implementation of BC enabled CNNs.

A Smart Contract (SC) is a code that can guarantee credible and secure transactions. SCs are also used to track the creation and updating of all transactions. The biggest advantage of these SCs is that it does not require any third-party API; thus the data cannot be compromised by any other external agent. SCs can be implemented at multiple levels on CNN to make it safer and more secure. All the inputs and outputs of the network can be saved in a ledger using an SC and can verify or restore the network inputs at any stage. Several SCs are formed during the proposed CNN, called Layer Ledger Block (LLB) with each layer, and stores the current and next layer's information. Another SC, called Central Ledger Block (CLB), is formed with all information about every layer of the network. CLB is stored within the network as local storage and a CLB copy is also stored on the cloud storage. The CLB at cloud storage keeps syncing with the local CLB. LLBs randomly update CLB so that the intruder cannot predict the order of each layer. The overall structure of SecureCNN is shown in Fig. 1.

The structure of SecureCNN is inspired by the architecture of BC itself, where blocks are connected in the form of an ever-growing linked list. The only difference is that the BC technology includes an infinite number of blocks and ledgers. Simultaneously, the SecureCNN has a finite number of blocks and ledgers, which solely depend upon the number of layers in a CNN network. A ledger block follows each layer of the network, which: a) Stores the parameter information of

the layer, b) Compute the output of current layer, c) Validate the output of the current layer and d) Update the ledger block of the current layer as well as the central ledger block. The structure of LLB and CLB is shown in Figs. 2 and 3, respectively.

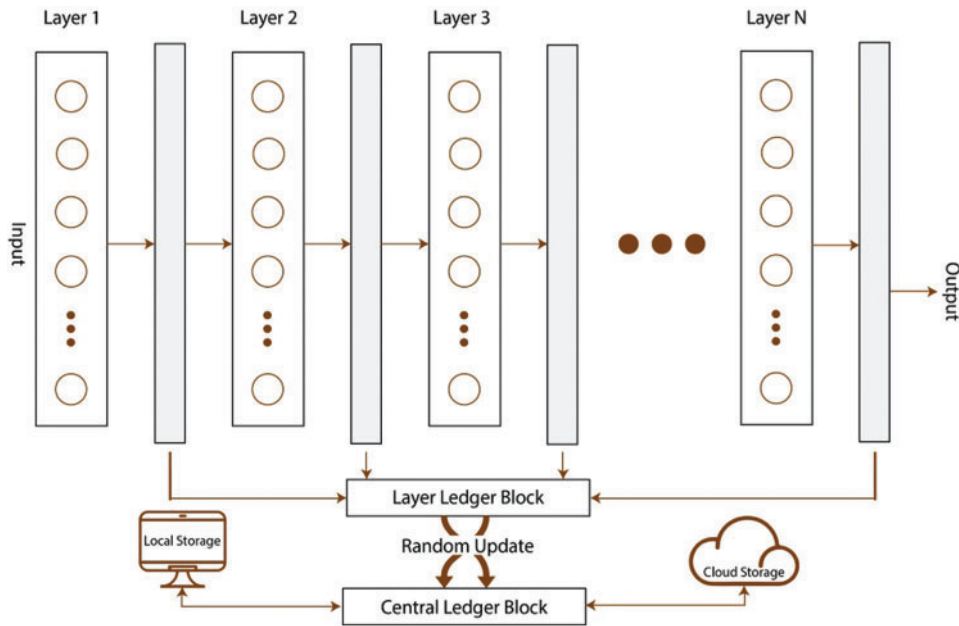


Figure 1: Architecture of SecureCNN

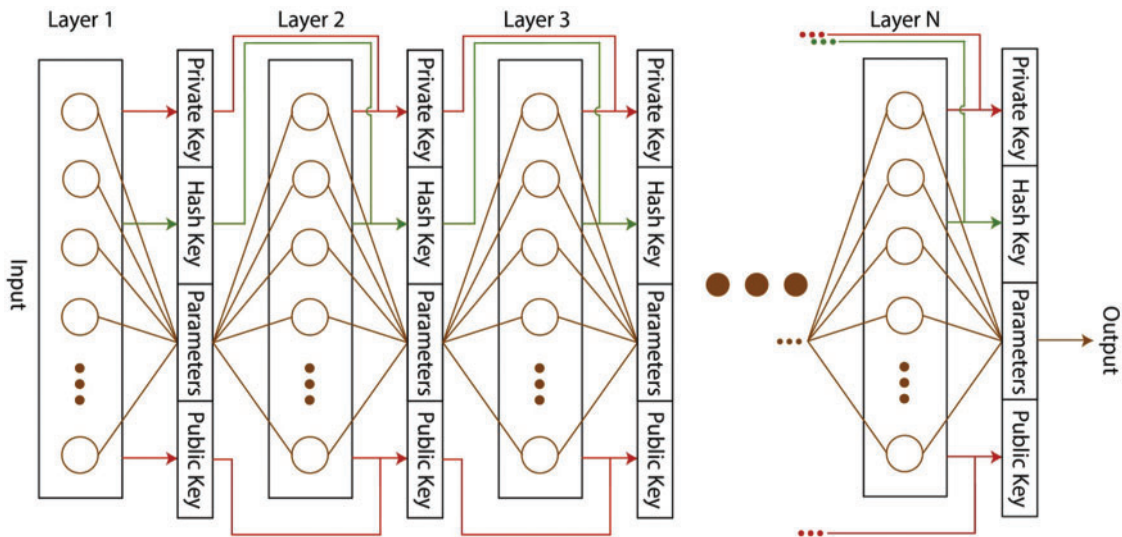


Figure 2: Structure of layer ledger block

Layer #	Hash _C	Hash _N	Public _C	Public _N	Private _C	Private _N	Layer _P	Layer _N	AE _C	w _C	b _C	Act _C	w _N	b _N	Act _N	Signature
3	Hash ₃	Hash ₄	Pub ₃	Pub ₄	Priv ₃	Priv ₄	2	4	AK ₃	w ₃	b ₃	Act ₃	w ₄	b ₄	Act ₄	Sign 3
6	Hash ₆	Hash ₇	Pub ₆	Pub ₇	Priv ₆	Priv ₇	5	7	AK ₆	w ₆	b ₆	Act ₆	w ₇	b ₇	Act ₇	Sign 7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	Hash ₁	Hash ₂	Pub ₁	Pub ₂	Priv ₁	Priv ₂	-	2	AK ₁	w ₁	b ₁	Act ₁	w ₂	b ₂	Act ₂	Sign 1
5	Hash ₅	Hash ₆	Pub ₅	Pub ₆	Priv ₅	Priv ₆	4	6	AK ₅	w ₅	b ₅	Act ₅	w ₆	b ₆	Act ₆	Sign 5

Figure 3: Structure of a central ledger block

The LLB is nothing but another layer of a CNN model with an identity weight matrix, zero bias, and identity function as an activation function. Thus, the output of the layer ledger block will be input itself. The LLB contains hashes of previous and current layers, private and public keys of current layers, public keys of immediate next and previous layers, and encrypted layer parameters of current and next layers. Hash generation and parameter encryption is achieved using a famous AE algorithm Data Encryption Standard (DES) [32]. Once a single layer is processed and provides output to the next layer, the whole process is treated as one transaction. The CLB contains information on all transactions random and associates each transaction with a signature. The purpose of randomizing this information is to make it more secure against the tempering. The central ledger block is a shared storage, which also stores the state of a model. A hash at a specific layer is calculated using the current and previous layer parameters and a hash of the previous layer. If it's the current layer, then the hash (H) of this layer can be calculated as:

$$H_i = \sigma(H_{i-1}, parameters_i, parameters_{i-1}) \quad (1)$$

Here, σ denotes the DES algorithm. These hash keys are stored in a central ledger block, which is later used to identify the tempered layer, in case of any tampering. The central ledger block has the information of every layer, which is stored randomly. Even the layers don't know their sequential orders in the central block. The layer ledger block calculates the authenticity parameter using the hash keys of the current and previous layers. This authenticity parameter has two possible values; true or false. If the value is set to true, the layer will take the output to the next layer. If the value is set to false, then the network has been compromised, and it will stop propagating the output to the next layer. It restores the parameters of previous and current layers and recalculates the hashes. This process will be repeated unless authenticity becomes valid again.

After the authenticity check, the central ledger block performs a) encryption of layer output through the public key of the next layer using DES, b) attach signature, and c) calculation of the hash of the next layer. After each update in the central block, each layer check either the update is verified through signature or not. The immediate next layer carries this verification. For any layer i , the signature $Sign_i$ of this layer can be calculated by encrypting the parameters of previous layer using the private key of current layer. If W_i denotes weights, I_i denotes the layer input and B_i denotes the bias, then the O_i using the activation function ρ can be calculated as $O_i = \rho((W_i \times X_i) + B_i)$. When encrypted using the public key of the next layer, this output becomes the input of that layer. $X_{i+1} = \sigma(O_i, Pub_{i+1})$. Using the parameters of current layer and hash of previous layer, the hash of the current layer is calculated as $H_i = \sigma(H_{i-1}, X_i, X_{i-1})$. Suppose the current layer updates the central ledger block using X_i, O_i, H_i and $Sign_i$. Then, at the next layer $i+1$, the verification process will be carried out by decrypting the previous layer's signature using the public key of the previous layer. If the signature matches, the outputs are valid otherwise,

the network is compromised. This signature verification ensures that the layer receives input from authorized layers.

The validation of any layer at any time can also be conducted. Suppose that layer i is tempered. Any layer is considered tempered if the signature of the current layer is not equal to decrypting the signature of the previous layer using the public key of previous layer or the input of the current layer X_i is not equal to the output of previous layer O_{i-1} . From this, we can conclude that either the O_i is not genuine, which implies that the previous layer $i-1$ is tempered or X_i are not genuine, which implies that the current layer i is tempered. O_{i-1} must be genuine as if it would have been tempered, the layer would never be able to produce an output, thus the current layer i is tempered.

3.1 Deep Learning Architecture Using SecureCNN

The Deep Learning (DL) architecture is used to classify the polyps into their related classes. The DL architecture consists of a pre-trained model AlexNet, InceptionV3, and DenseNet201. These models are transformed into SecureCNN models as per the proposed method to extract features of images. The extracted features are fused by the mode value-based serially approach and optimized using GA. Through GA, the most optimal features are selected and passed in the Softmax classifier for final classification. The overall flow of DL architecture is shown in Fig. 4.

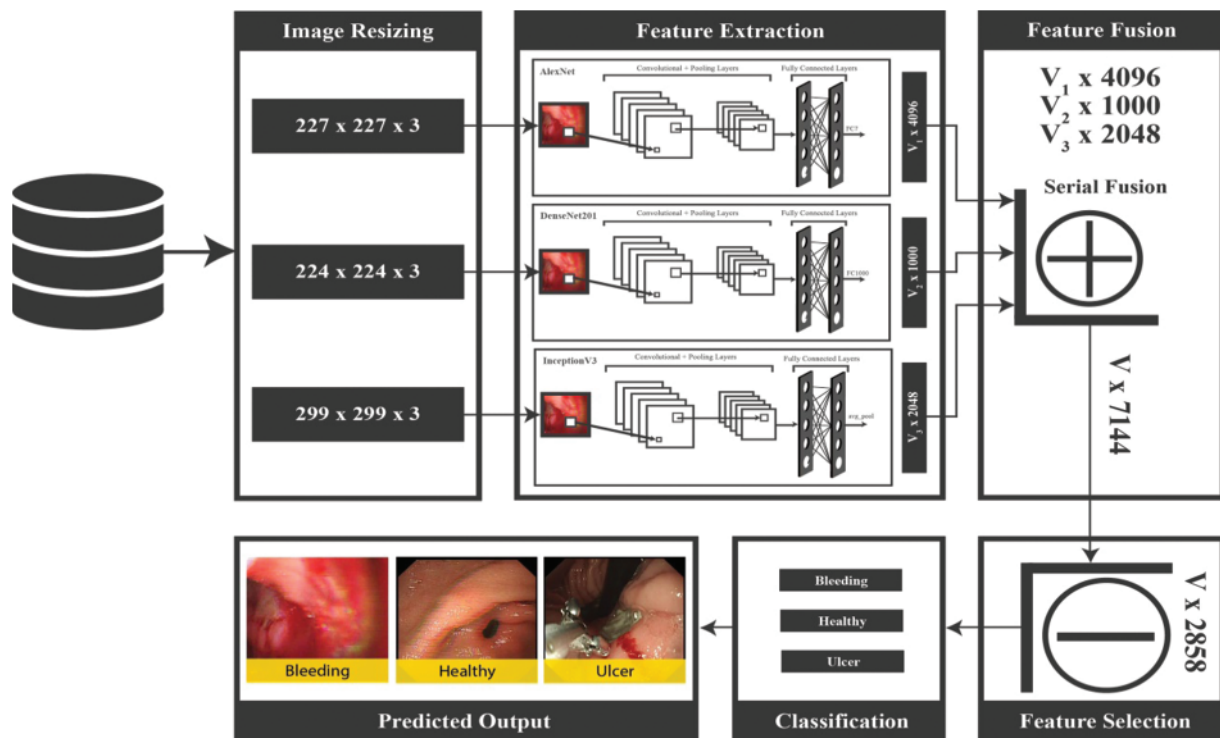


Figure 4: Proposed deep learning architecture for stomach infections recognition

3.1.1 Convolutional Neural Network (CNN)

The CNNs were proposed by [33] to classify the handwritten digits [34]. CNN models are inspired by the human mind’s biological structure, where neurons transfer the information from

one cell to another. The firing capacity and accuracy of neurons determine the intellectual level of a human. Similarly, the success of CNN's depends upon its learning and reducing the error-rate. The CNN architectures are made up of several layers, which perform different tasks at different levels. These networks always start with an input layer, which only accepts a certain specific size image. The input image is followed by multiple combinations of convolutional layers, pooling layers, ReLU layers, and normalization layers.

The last layers of CNN are used to extract the learned features and mostly include fully connected layers and the Softmax layer. The pre-trained networks are famous whenever we talk about the CNNs. These pre-trained networks differ from other Machine Learning (ML) networks, as pre-processed images are input instead of feature vectors. These models are trained in a supervised environment on large datasets like ImageNet.

AlexNet: It has eight (8) distinguished layers, out of which five connected convolutional layers are at the beginning with pooling layers, followed by three (3) fully-connected layers [35]. The output layer of this model is the Softmax layer, which is directly connected with the last fully-connected layer. The last layer is labeled as the FC8 layer, which fed the Softmax layer with 1000 size and softmax, which produces 1000 channels. Neurons of fully connected layers are directly attached to neurons of previous layers. Normalization layers are connected with the first and second layers. The fifth convolutional layer and response normalization layers have max-pooling layers. The output of every fully connected and convolutional layer has a ReLU layer. Input for this network is an RGB image of size $227 \times 227 \times 3$. The FC7 layer returned a feature matrix of dimension $V_1 \times 4096$.

DenseNet: It consists of a total of four dense blocks like dense block 1, dense block 2, dense block 3, and dense block 4 [36]. In the first three dense blocks, a transition layer is added for each, and for the last dense block, a classification layer is added. The output size of the first convolutional layer is 112×112 , where filter size 7×7 and stride 2×2 . After the convolutional layer, the max-pooling layer is added of pooling size 3×3 and stride 2. The global average pool layer of filter size 7×7 and fully connected (FC) layer are added in the classification layer. The FC layer returned a feature matrix of dimension $V_2 \times 1000$. Input for this network is an RGB image of size $224 \times 224 \times 3$.

Inception V3: It was presented as an enhanced adaptation of the ILSVRC-2014 Large Scale Visual Recognition Challenge [37]. The system was intended to decrease the computational expense while enhancing the characterization precision with the goal that computer-related applications can also be versatility ported with it. It achieves 22.0% best 1 and 6.1% best 5 error ratio on the ILSVRC-2012 characterization [38]. This model includes 346 layers, and the input for this network is an RGB image of size $299 \times 299 \times 3$. The "avg_pool" layer returned a feature matrix of dimension $V_3 \times 2048$.

Features Extraction: Features are extracted from three layers, such as FC seven layer of AlexNet, FC eight layer of Inception V3, and global average pool layer of DenseNet201. These models are trained using the destination transfer learning approach. The feature vector size of each selected layer is $V_1 \times 4096$, $V_2 \times 1000$, and $V_3 \times 2048$. These features are fused by a Mode value-based approach, discussed in the next section.

3.2 Tempering Attack

The primary purpose is to prevent tampering attacks against a learned model so that the performance and results are not compromised. To test the capabilities of proposed SecureCNN, a

tempering attack is proposed in this article, which tries to temper the learned model at different levels. The proposed attack justifies the integration of BC technology with CNNs. Algorithm 1 presents the pseudo-code for the proposed tempering attack.

Algorithm 1: Parameter tempering for a model

Input: model, parameters
Output: model_t

- 1: $i \leftarrow 0$
- 2: layers \leftarrow model.layers
- 3: attack_{type} \leftarrow [0, 1, 2]
- 4: if (attack_{type} == 0)
 - attack_{name} \leftarrow mild
 - Classes \leftarrow findSoftmaxLayer (model.layers)
 - Classes_{new} \leftarrow interchange (Classes)
 - findSoftmaxLayer (model.layers) = Classes_{new}
- End if
- 5: else if (attack_{type} == 1)
 - attack_{name} \leftarrow average
 - Perform Attack₁
 - Layers_{output} = FindOutputLayers(model)
 - Layer_{shape} \leftarrow Layers_{output}.shape
 - noise \leftarrow GaussianNoise(ρ , Layer_{shape})
 - w \leftarrow weights(Layers_{output})
 - w_{new} \leftarrow w + noise
 - weights (Layers_{output}) = w_{new}
- End if
- 6: else if (attack_{type} == 2)
 - attack_{name} \leftarrow severe
 - Perform Attack₂
 - for $i \leftarrow 0$ to size(Layers)
 - Layer_{shape} \leftarrow Layers_i.shape
 - noise \leftarrow GaussianNoise(ρ , Layer_{shape})
 - w \leftarrow weights(Layers_i)
 - w_{new} \leftarrow w + noise
 - weights (Layers_i) = w_{new}
 - end for
- end if
- 7: model_t = trainMod(model)

3.3 Features Fusion and Selection

Consider we have three feature vectors named AlexNet features, Inception V3 features, and Densenet201 features denoted by $\psi_1(k_1)$, $\psi_2(k_2)$, and $\psi_3(k_3)$. Where k_1 , k_2 , and k_3 represent the length of the extracted features. As we know, the length of the feature vector, mentioned in

Fig. 4. We first serially combined all features in one vector as follows:

$$\psi_{ki} = \begin{pmatrix} \psi_1(k1) \\ \psi_2(k2) \\ \psi_3(k3) \end{pmatrix}_{N \times Ki} \in \mathbb{R} \quad (2)$$

where, ψ_{ki} denotes serially combined vector, and K_i represents the size of the final fused vector. Later on, we organize all features in the highest value-based and for this, mode value is computed. Based on mode value, features are arranged in the highest order. Later, applied Genetic Algorithm of Entropy controlled Naïve Bayes (GAecNB) fitness function. The algorithm of GAecNB is given below:

Algorithm 2: Features Optimization using GAecNB

Input: Fused Feature Vector $\leftarrow \psi_{ki}$

Output: Optimal Feature Vector $\leftarrow \phi_{ki}$

1: Initialized Parameters

- Population = 100
- Iterations = 500
- MutationRate = 0.001
- Crossover = 0.8
- B = 6

2: Calculate Fitness

- $E = -\sum_{k=1}^N p(ki) \log P(ki)$
- $H(k) = P(K_1|K_2) = \frac{P(K_2|K_1)P(K_1)}{P(K_2)}$
- Calculate Error

3: Perform Uniform Cross Over

4: Generate Features using Roulette Wheel

5: Perform Mutation

6: Merge Populations

7: Sort best population

8: Best Chromosomes

Finally, the selected features are passed in the Softmax classifier for final classification. Mathematically, Softmax is formulated as follows:

$$SM = \frac{\exp(ki)}{\sum_i \exp(ki)} \quad (3)$$

When the feature vectors $V_1 \times 4096$, $V_2 \times 1000$, and $V_3 \times 2048$ are fused into a stand-alone vector; final size becomes $V \times 7144$. After applying GA, feature vector V becomes of size $V \times 2858$, 40% of fused features.

4 Experimental Results and Analysis

4.1 Experimental Setup

The proposed CNN model is trained on NVIDIA GeForce GTX 1080 with 6.1 computation capability, seven multiprocessors, and 1607–1733 MHz clock rate. The dataset is divided into two

parts: training and testing, using a traditional approach of 50–50. The CNN model is trained and testing using MATLAB 2019b. The Stochastic Gradient Descent with momentum (SGDM) algorithm represents the minibatch size training technique of 64. The learning rate is started at 0.01 and decreased after every 20 epochs by 10. The momentum is set at 0.4 and maximum epochs are set at 450. Cross-Entropy [39] is used as a suitable loss function as it has performed reasonably for many multiclass issues. For the CNN models, different output layers are selected to extract features. AlexNet model extracts the 4096 features against a single image on the FC7 layer; InceptionV3 extracts 2048 features for one image on the *avg_pool* layer; the densenet201 model extracts 1000 features against one image on the fc1000 layer. For the hand-crafted features, the input image size is fixed to $250 \times 250 \times 3$.

4.2 Dataset

In this work, we used a Private Stomach dataset, originally collected by Liaqat et al. [9]. Later, the number of images are increased and reached up to above 5500 by Sharif et al. [23]. This dataset was originally collected in videos from POF Hospital, Wah Cantt, Pakistan. In this work, we further increase the images, and each class images are 5000. Three classes of the selected dataset are ulcer, Bleeding, and Healthy.

4.3 Fine Tuning of CNN Models

Transfer learning is an essential element for datasets, which does not have many images or classes [40]. For the smaller datasets, pre-trained networks are utilized as a feature extractor. Fine-tuning has shown promising results as compared to generic feature extraction using CNN [41]. The CNN models, i.e., AlexNet, InceptionV3, and DenseNet201, are already trained on large-scale dataset ImageNet having 1000 classes. The selected dataset only has three classes, so the softmax layer of these pre-trained models is updated by replacing 1000 with 3. But this change forces the network to start training process with some random weights on each layer. The training accuracy and training loss of fine-tuning can be observed in Fig. 5. The softmax layer's learning rate increases exponentially in transfer learning as it must learn the new features quickly. The pre-trained models are fine-tuned over a mini-batch size of 64, weight decay of 0.005, and momentum of 0.7. A Gaussian distribution with 0.01 standard deviation is used to initialize the weights of the softmax layer containing three (3) classes. For 450 iterations, a dropout size of 0.5 is fixed to avoid overfitting. The CNN models are trained over 70%, 15%, 15% ratio for training, and testing and validation.

4.4 Classification Results

The classification results are computed using several experiments, where features from standalone networks, fused features, and optimized features were utilized to obtain results. All the networks are used as simple CNNs, as well as SCNNs. Tab. 1 shows the classification results in different experiments. The results on both CNNs and SCNNs remain almost the same with an extremely low variation. All pre-trained networks are used with and without fine-tuning to compare the impact. In the first experiment, AlexNet, without fine-tuning got a classification accuracy of 78.3% with 21.7% FNR. The model was trained in 214.3 s and the average prediction time remained at 1.63 s. When AlexNet was used with fine-tuning, results improved 10.34% by achieving an accuracy of 86.4% and FNR of 13.6%. The training time decreased by 16.52% to 178.9 s and the average prediction time decreased 26.99% to 1.19 s. In the second experiment, InceptionV3 without fine-tuning got a classification accuracy of 83.8% with 16.2% FNR. The model was trained in 193.4 s and the average prediction time remained at 0.94 s. When

InceptionV3 was used with fine-tuning, results improved 4.42% by achieving 87.5% accuracy, FNR of 12.5%. The training time decreased by 12.20% to 169.8 s and the average prediction time decreased 11.70% to 0.83 s. In the third experiment, DenseNet201 without fine-tuning got a classification accuracy of 74.3% with 25.7% FNR. The model was trained in 187.1 s and the average prediction time remained at 1.23 s. When DenseNet201 was used with fine-tuning, results improved 14% by achieving 84.7% accuracy, FNR of 15.3%. The training time decreased by 12.61% to 163.5 s and the average prediction time decreased 20.23% to 0.98 s. In the fourth experiment, fused networks without fine-tuning got a classification accuracy of 89.4% with 10.6% FNR. As the model contains all features from the previous three pre-trained networks, training time increases to 497.2 s and the average prediction time remained at 0.78 s. When networks with fine-tuning were fused, results improved by achieving 90.3% accuracy, FNR of 9.7%. The training time decreased by 30.03% to 347.9 s and average prediction time decreased 19.23% to 0.63 s. In the final experiment, optimized features of networks without fine-tuning got classification accuracy of 92.4% with 7.6% FNR. The model was trained in 200.8 s and the average prediction time remained at 0.42 s. When optimized features of networks with fine-tuning are utilized, results improved 4.76% by achieving 96.8% accuracy, FNR of 3.2%. The training time decreased by 43.38% to 113.7 s and the average prediction time decreased 78.57% to 0.09 s. Fig. 6 shows the impact of selected features during optimization.

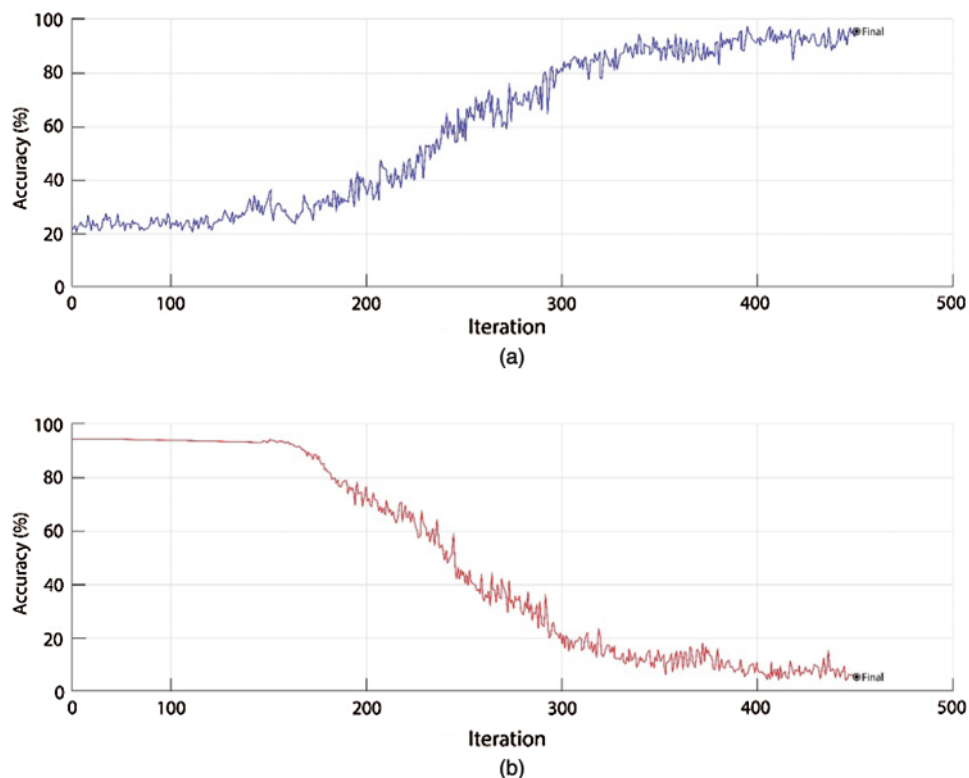
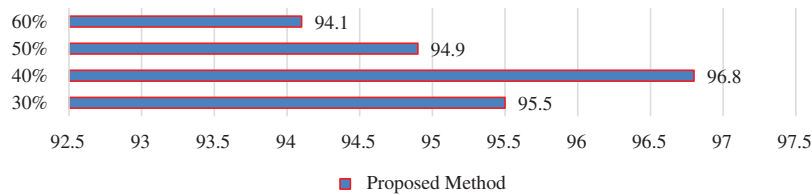


Figure 5: Training accuracy and training loss for AlexNet, VGG19, and DenseNet201

Table 1: Classification results on different experiments

Model	Fine tuning		Accuracy (%)	FNR (%)	Training time (s)	Avg. prediction time (s)
	No	Yes				
AlexNet (A)	✓		78.3	21.7	214.3	1.63
		✓	86.4	13.6	178.9	1.19
InceptionV3 (I)	✓		83.8	16.2	193.4	0.94
		✓	87.5	12.5	169.8	0.83
Densenet201 (D)	✓		74.3	25.7	187.1	1.23
		✓	84.7	15.3	163.5	0.98
Fused [A + I + D]	✓		89.4	10.6	497.2	0.78
		✓	90.3	9.7	347.9	0.63
Optimized CNN (OCNN) [A + I + D]	✓		92.4	7.6	200.8	0.42
		✓	96.8	3.2	113.7	0.09

**Figure 6:** Impact of selected features during feature optimization

Several experiments were carried during the optimization procedure to check the impact of the feature vector size. These experiments include selecting 30%, 40%, 50%, and 60% features of the final feature vector. It can be seen that the highest classification accuracy of 96.8 is achieved by selecting 40% of the features as compared to 95.5%, 94.9%, and 94.1% on 30%, 50%, and 60%, respectively. The reduction of features also decreased training time and, eventually, prediction time for any input image. Fig. 7 demonstrates the confusion matrices of experiments with the highest accuracies with fine-tuning.

It can be seen from the confusion matrices that True Positive Rate (TPR) and False Negative Rate (FNR) rates improved by performing fusion and optimization. The TPR remained on average at 87% for InceptionV3, 90% for fused features, and 96% for optimized features. While the Positive Predictive Values (PPV) and False Discovery Rate (FDR) also improved from 87% for InceptionV3, 90% for fusion, and 95% for optimization. Correctly and incorrectly predicted images are shown in Figs. 8 and 9 respectively. During the testing of the proposed method on selected dataset, few images were incorrectly classified, which degraded the proposed model's accuracy. All these images have incorrectly predicted labels on the image with a yellow background and correct labels under the image in black background.

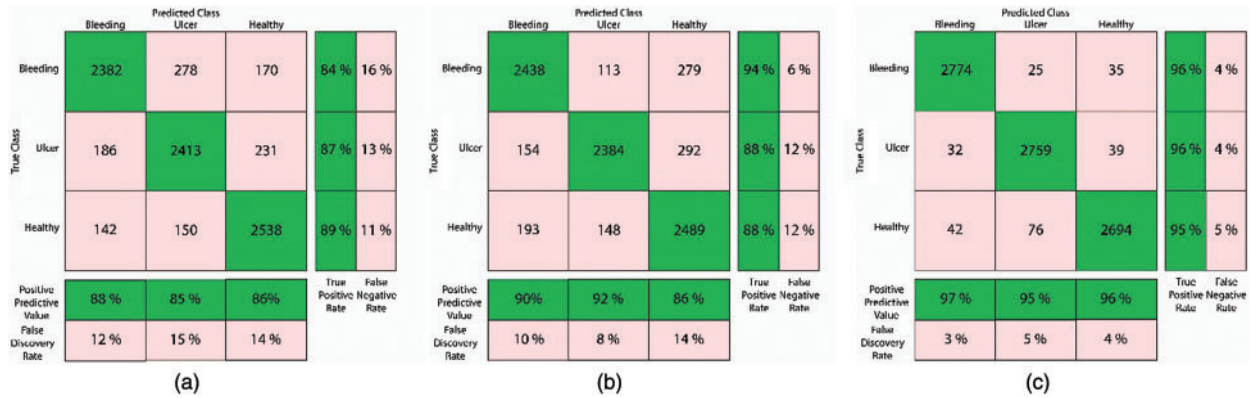


Figure 7: Confusion matrices of (a) InceptionV3; (b) Fusion of all networks; and (c) Optimized features

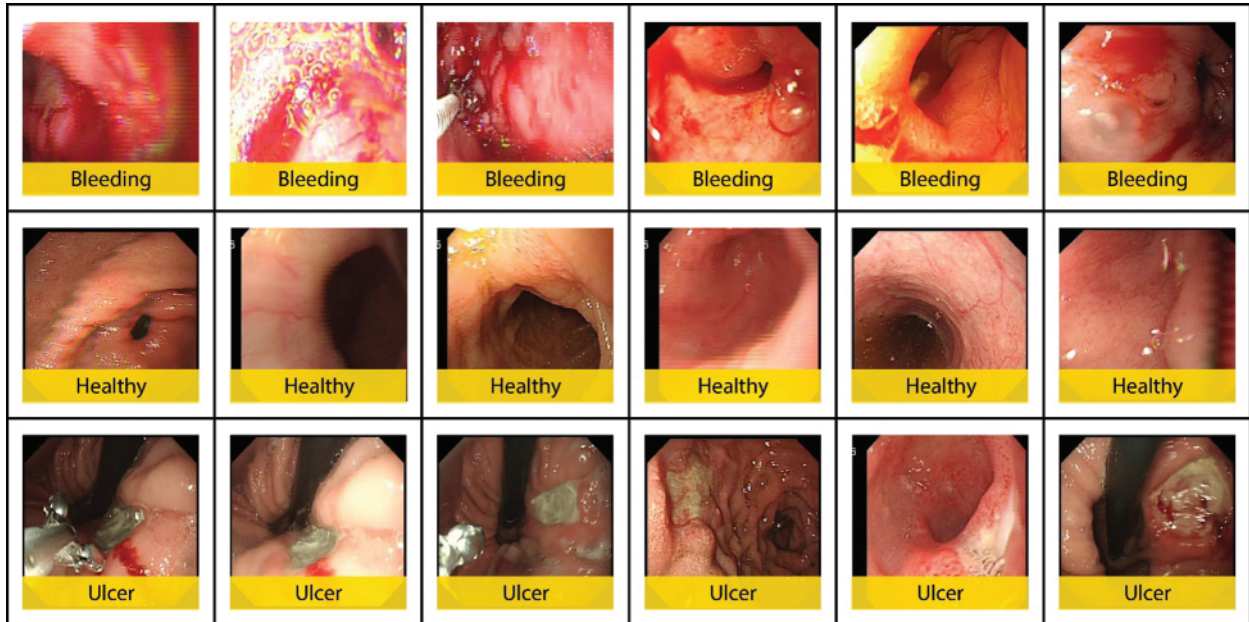


Figure 8: Correctly labeled images using proposed model

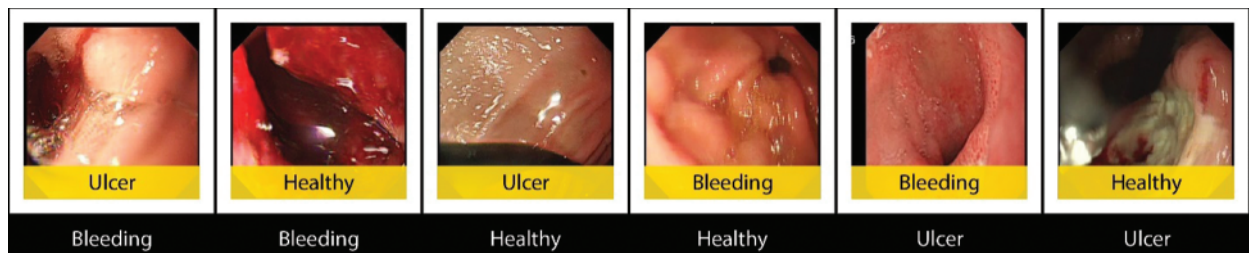


Figure 9: Incorrect predictions using the proposed model

Table 2: Impact of tempering attacks on models with and without blockchain

Attack type	Blockchain inclusion		Accuracy (%)
	No	Yes	
Mild	✓		83.7
		✓	96.7
Average	✓		59.1
		✓	96.3
Severe	✓		35.8
		✓	96.0

Table 3: Comparison with existing techniques

Model	Year	Accuracy (%)	Dataset size (images)	Training time (s)	Avg. prediction time (s)
Liaqat [9]	2018	98.3	442	–	–
Sharif [23]	2019	99.4	5,500	–	–
Khan [42]	2020	98.4	6,000	–	–
Proposed	2020	92.4	18,000	200.8	0.42
		96.8		113.7	0.09

During the experiments, the trained classifiers were modified with different kinds of tempering attacks. These attacks were carried out at different severity levels. The severity of attacks was categorized as mild, average, and severe attacks. In the mild attack, only the output classes were interchanged, while in the average attack, the output classes and weights of output layers were tempered. In the severe attack, the weights of all layers, sizes of filters, strides, output size of the output layer, and output classes are modified. The results of networks with and without blockchain inclusion are illustrated in [Tab. 2](#). A comparison with existing techniques, is presented in [Tab. 3](#). In this table, it is shown that the proposed approach works better as compared to exiting techniques.

5 Conclusion

An existing blockchain approach is implementing in this work to secure the CNN model for stomach infection classification. Three deep models are employing and secure through implemented blockchain framework and extract the features. Features are fused using serially mode value. Later on, we try to improve the GA using the proposed approach name GAecNB. Through this approach, selected optimal chromosomes known as features are obtained and passed in Softmax Classifier for final classification. Based on results, it can be observed that even the mild attack decreased the accuracy of the proposed model by 13.44%, and when the mild attack was performed on a network with blockchain, the results remain almost the same.

Similarly, the average and severe attacks decreased classification accuracies by 38.88% and 62.98%, respectively. These findings prove the authenticity of proposed secure models and their

robustness against the tempering attacks. In the future, SecureCNN can be made more secure by employing multiple hashing algorithms and intricate integration of LLBs with CNN.

Funding Statement: This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. A. Khan, M. A. Khan, F. Ahmed, M. Mittal, L. M. Goyal *et al.*, “Gastrointestinal diseases segmentation and classification based on duo-deep architectures,” *Pattern Recognition Letters*, vol. 131, pp. 193–204, 2020.
- [2] A. Majid, M. A. Khan, M. Yasmin, A. Rehman, A. Yousafzai *et al.*, “Classification of stomach infections: A paradigm of convolutional neural network along with classical features fusion and selection,” *Microscopy Research and Technique*, vol. 83, no. 5, pp. 562–576, 2020.
- [3] J. H. Lee, Y. J. Kim, Y. W. Kim, S. Park, Y. I. Choi *et al.*, “Spotting malignancies from gastric endoscopic images using deep learning,” *Surgical Endoscopy*, vol. 33, no. 11, pp. 3790–3797, 2019.
- [4] R. L. Siegel, K. D. Miller, S. A. Fedewa, D. J. Ahnen, R. G. Meester *et al.*, “Colorectal cancer statistics,” *CA: A Cancer Journal for Clinicians*, vol. 67, no. 3, pp. 177–193, 2017.
- [5] S. Namasudra, P. Roy, P. Vijayakumar, S. Audithan and B. Balusamy, “Time efficient secure DNA based access control model for cloud computing environment,” *Future Generation Computer Systems*, vol. 73, pp. 90–105, 2017.
- [6] N. Ghatwary, X. Ye and M. Zolgharni, “Esophageal abnormality detection using densenet based faster r-cnn with gabor features,” *IEEE Access*, vol. 7, pp. 84374–84385, 2019.
- [7] S. Menon and N. Trudgill, “How commonly is upper gastrointestinal cancer missed at endoscopy? a meta-analysis,” *Endoscopy International Open*, vol. 2, no. 2, E46, 2014.
- [8] Y. Fu, W. Zhang, M. Mandal and M. Q. H. Meng, “Computer-aided bleeding detection in WCE video,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 636–642, 2013.
- [9] A. Liaqat, M. A. Khan, J. H. Shah, M. Sharif, M. Yasmin *et al.*, “Automated ulcer and bleeding classification from WCE images using multiple features fusion and selection,” *Journal of Mechanics in Medicine and Biology*, vol. 18, no. 4, 1850038, 2018.
- [10] D. K. Iakovidis, S. V. Georgakopoulos, M. Vasilakakis, A. Koulaouzidis and V. P. Plagianakos, “Detecting and locating gastrointestinal anomalies using deep learning and iterative cluster unification,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 10, pp. 2196–2210, 2018.
- [11] M. Vasilakakis, A. Koulaouzidis, D. E. Yung, J. N. Plevris, E. Toth *et al.*, “Follow-up on: Optimizing lesion detection in small bowel capsule endoscopy and beyond: From present problems to future solutions,” *Expert Review of Gastroenterology & Hepatology*, vol. 13, no. 2, pp. 129–141, 2019.
- [12] S. Lu, Z. Lu and Y. D. Zhang, “Pathological brain detection based on alexnet and transfer learning,” *Journal of Computational Science*, vol. 30, pp. 41–47, 2019.
- [13] S. Targ, D. Almeida and K. Lyman, “Resnet in resnet: Generalizing residual architectures,” arXiv preprint arXiv: 1603.08029, 2016.
- [14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv: 1409.1556, 2014.
- [15] A. Rajaei and L. Rangarajan, “Wavelet features extraction for medical image classification,” *An International Journal of Engineering Sciences*, vol. 4, pp. 131–141, 2011.
- [16] S. Suman, F. A. Hussin, A. S. Malik, S. H. Ho, I. Hilmi *et al.*, “Feature selection and classification of ulcerated lesions using statistical analysis for WCE images,” *Applied Sciences*, vol. 7, no. 10, pp. 1097, 2017.

- [17] Y. Yuan, B. Li and M. Q. H. Meng, "WCE abnormality detection based on saliency and adaptive locality-constrained linear coding," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 149–159, 2016.
- [18] M. A. Khan, M. Rashid, M. Sharif, K. Javed and T. Akram, "Classification of gastrointestinal diseases of stomach from WCE using improved saliency-based method and discriminant features selection," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27743–27770, 2019.
- [19] A. S. Ghareb, A. A. Bakar and A. R. Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Systems with Applications*, vol. 49, pp. 31–47, 2016.
- [20] M. P. McBee and C. Wilcox, "Blockchain technology: Principles and applications in medical imaging," *Journal of Digital Imaging*, vol. 33, pp. 726–734, 2020.
- [21] X. Cheng, F. Chen, D. Xie, H. Sun and C. Huang, "Design of a secure medical data sharing scheme based on blockchain," *Journal of Medical Systems*, vol. 44, no. 2, pp. 52, 2020.
- [22] A. Goel, A. Agarwal, M. Vatsa, R. Singh and N. Ratha, "DeepRing: Protecting deep neural network with blockchain," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, 2019.
- [23] M. Sharif, M. Attique Khan, M. Rashid, M. Yasmin, F. Afza *et al.*, "Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–23, 2019.
- [24] N. Nida, M. Sharif, M. U. G. Khan, M. Yasmin and S. L. Fernandes, "A framework for automatic colorization of medical imaging," *IIOAB J*, vol. 7, pp. 202–209, 2016.
- [25] I. M. Nasir, M. A. Khan, M. Alhaisoni, T. Saba, A. Rehman *et al.*, "A hybrid deep learning architecture for the classification of superhero fashion products: An application for medical-tech classification," *Computer Modeling in Engineering & Sciences*, vol. 124, no. 3, pp. 1017–1033, 2020.
- [26] I. M. Nasir, M. Rashid, J. H. Shah, M. Sharif, M. Y. H. Awan *et al.*, *An Optimized Approach for Breast Cancer Classification for Histopathological Images Based on Hybrid Feature Set*. Current Medical Imaging, 2020.
- [27] M. A. Khan, M. Sharif, T. Akram, M. Yasmin and R. S. Nayak, "Stomach deformities recognition using rank-based deep features selection," *Journal of Medical Systems*, vol. 43, no. 12, pp. 329, 2019.
- [28] M. Sun, K. Liang, W. Zhang, Q. Chang and X. Zhou, "Non-local attention and densely-connected convolutional neural networks for malignancy suspiciousness classification of gastric ulcer," *IEEE Access*, vol. 8, pp. 15812–15822, 2020.
- [29] W. Yang, Y. Cao, Q. Zhao, Y. Ren and Q. Liao, "Lesion classification of wireless capsule endoscopy images," in *2019 IEEE 16th Int. Symp. on Biomedical Imaging*, Venice, Italy, pp. 1238–1242, 2019.
- [30] S. Namasudra, G. C. Deka, P. Johri, M. Hosseinpour and A. H. Gandomi, "The revolution of blockchain: State-of-the-art and research challenges," *Archives of Computational Methods in Engineering*, 2020.
- [31] S. Namasudra, C. Practice and Experience, "An improved attribute-based encryption technique towards the data security in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 3, e4364, 2019.
- [32] S. Namasudra and P. Roy, "Time saving protocol for data accessing in cloud computing," *IET Communications*, vol. 11, no. 10, pp. 1558–1565, 2017.
- [33] D. Coppersmith, "The data encryption standard and its strength against attacks," *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 243–250, 1994.
- [34] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, pp. 1995, 1995.
- [35] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard *et al.*, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, pp. 396–404, 1990.
- [36] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

- [37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.*, “Going deeper with convolutions,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. of the IEEE Conf. on Computer Vision and pattern recognition*, pp. 2818–2826, 2016.
- [39] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell *et al.*, “Densenet: Implementing efficient convnet descriptor pyramids,” arXiv preprint arXiv: 1404. 1869, 2014.
- [40] J. Shore and R. Johnson, “Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy,” *IEEE Transactions on Information Theory*, vol. 26, no. 1, pp. 26–37, 1980.
- [41] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.
- [42] M. A. Khan, S. Kadry, M. Alhaisoni, Y. Nam, Y. Zhang *et al.*, “Computer-aided gastrointestinal diseases analysis from wireless capsule endoscopy: A framework of best features selection,” *IEEE Access*, vol. 9, pp. 132850–132859, 2020.