Received 28 September 2023, accepted 12 November 2023, date of publication 20 November 2023,
date of current version 28 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334650

RESEARCH ARTICLE

Enhancing Secret Data Detection Using Convolutional Neural Networks With Fuzzy Edge Detection

NTIVUGURUZWA JEAN DE LA CROIX[1,2], (Member, IEEE),
TOHARI AHMAD[1], (Member, IEEE), AND FENGLING HAN[3], (Senior Member, IEEE)
[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2]African Center of Excellence in Internet of Things, College of Science and Technology, University of Rwanda, Kigali, Rwanda
[3]School of Computing Technologies, RMIT University, Melbourne, VIC 3000, Australia

Corresponding author: Tohari Ahmad (tohari@if.its.ac.id)

This research was supported by Institut Teknologi Sepuluh Nopember, grant no. 1721/PKS/ITS/2023, 1368/PKS/ITS/2023.

ABSTRACT Progress in Deep Learning (DL) has introduced alternative methods for tackling complex challenges, such as the steganalysis of spatial domain images, where Convolutional Neural Networks (CNNs) are employed. In recent years, various CNN architectures have emerged, enhancing the precision of detecting steganographic images. Nevertheless, current CNNs encounter challenges related to the inadequate quality and quantity of available datasets, high imperceptibility of low payload capacities, and suboptimal feature learning processes. This paper proposes an enhanced secret data detection approach with a CNN architecture that includes convolutional, depth-wise, separable, pooling, and spatial dropout layers. An improved fuzzy Prewitt approach is employed for pre-processing the images prior to being fed into CNN to address the issues of low payload capacity detection and dataset quality and quantity in learnability of the image features. Experimental results, which achieved an overall accuracy and F1-score of 99.6 and 99.3 per cent, respectively, to detect a steganographic payload of 0.5 bpp hidden with Wavelet Obtained Weights (WOW), show a significant outperformance over the state-of-the-art methods.

INDEX TERMS Convolutional neural networks, fuzzy logic, information security, network infrastructure, network security, spatial domain, steganalysis.

I. INTRODUCTION

Steganography stands as both a method and an artistic endeavour to conceal confidential communication within seemingly ordinary digital content such as digital images [1], [2], audio [3], [4] and video [5]. Unlike steganography, steganalysis is counter-art aiming to identify whether concealed messages are present within publicly transmitted media [6]. Over the last few decades, steganalysis and steganography have maintained a symbiotic relationship, being interchangeably employed and mutually fostering each other's advancement. Steganography within digital images has recently garnered significant attention due to its widespread adoption across various social media platforms [7].

Steganalysts use two distinct methodologies: targeted and universal steganalysis. Targeted steganalysis focuses on pre-emptively identifying stego images, resulting in specific steganographic techniques [8], and universal steganalysis focuses on detecting the stego images generated through various steganographic methods without prior knowledge of the exact algorithms employed [9]. In any steganographic strategy, the primary objective is to optimize the imperceptibility of the secret information into the content of an image, thereby preserving the integrity of the original cover image. The high imperceptibility of a stego image makes it hard and mainly impossible to discern the presence of the secret bits. Therefore, the primary endeavor of a steganalyst revolves around discriminating between these two states:

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

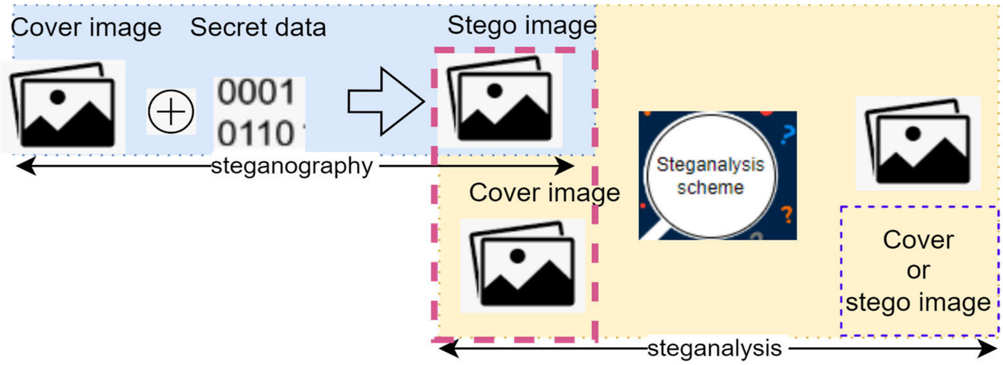

FIGURE 1. The general paradigm of steganography and steganalysis.

determining whether an image is an original cover or a stego image [10].

The general concept of steganography and steganalysis in images is illustrated in Fig. 1, with two main parts, one representing steganography, highlighted in blue, and another representing steganalysis, highlighted in orange. In the steganography part, the cover represents the original input image used as a carrier of the secret data. The output of the steganography part is the stego image, equivalent to a combination of the cover and the secret message. In the steganalysis part, the stego images resulting from the steganographic scheme are labeled as stego, and the cover images are labeled as cover to be considered as the inputs of a steganalysis scheme (as encircled in pink dashes). Based on the main objective of a steganalysis scheme of binary classifying images as cover or stego, a steganalysis scheme classifies images as cover or stego images.

The categorization of steganalysis predominantly hinges on the resulting outcomes of the procedure, which can be classified into four primary classes. Steganalysis, encompassing an image classification outcome, distinguishing between a cover and stego, is referred to as detective steganalysis [9]; a steganalysis approach aimed at revealing the concealed data's positions is recognized as locative steganalysis [11]. A steganalysis strategy that seeks to identify the payload size is referred to as quantitative steganalysis [8], and forensic steganalysis denotes a steganalysis procedure to extract the concealed bits of a confidential message [12]. In line with steganalysis to detect the existence of secret data, several machine learning (ML) methods have been suggested in the field of steganalysis [13], [14]. These approaches involve a two-step process comprising feature extraction and classification. Importantly, there is no form of backward communication between these two stages.

Nevertheless, the steganalysis methodologies developed using ML algorithms yielded unsatisfactory outcomes for the complete spectrum of steganalysis duties, primarily attributed to the fundamental logic inherent in these ML techniques. Lately, researchers have turned to DL models to enhance the overall efficacy of steganalysis outcomes. In contrast to conventional ML models, DL models facilitate bidirectional communication between the feature extraction and classification phases, enabling the conception of novel strategies to execute steganalysis operations on digital images. DL-based steganalysis models encompass architectures such as deep neural networks (DNN) and convolutional neural networks (CNN). These architectures link the processes of feature extraction and classification through backward communication into a unified phase [9], [10], [15]. The methods based on CNNs showcased that enhancing the feature extraction phase significantly augmented the efficacy of model generation for image classification. This substantial enhancement played a crucial role in elevating the quality of steganalysis performance.

Moreover, several other researchers proposed methods to combine the CNNs with other approaches [16], [17] that preprocess the images used in the binary classification to improve the results in steganalysis tasks. However, the proposed methods exhibit certain drawbacks regarding classification accuracy due to several problems, which include the lack of quality dataset and extensive training dataset, the inconsistency of the feature learning process, and the use of low payload by steganography practitioners, showing the need for further enhancement to effectively mitigate the risk of undetected secret communication, which could potentially have detrimental implications for companies, governmental institutions, and the community in general. This aspect assumes paramount significance within the realm of image digital forensics.

The research gaps identified in the existing steganalysis models are mainly founded on the risk of the inability to reveal all possible covert communication within digital images with certainty. Explicitly, this research addresses two main research problems, namely, the high imperceptibility of stego images resulting from low payload capacities and the issue of low learnability for features of the stego images. This paper proposes a solution based on combining CNN with a mathematical paradigm known as fuzzy logic to address the current research gaps. Benefiting from the ability of fuzzy logic to optimistically contribute to the classification problems [18], [19], [20], [21], [22], [23], we pre-process the cover images to make them the best version of images for classification because we remove the unwanted regions. In our CNN, we refer to [6] and [24] to integrate the depth-wise separable convolutions combined with other functions to design a new CNN with reduced dimensionality, which showed outperformance in stego image detection.

The contributions and the novelty of our work, which make it outperform the existing works, are summarized in the following points:

1) Enhancing low payload detection: By operating on the edge levels of the inquiry images, we achieve improved detection of low payload steganography based on the contribution of the training samples selection and approaches used in learning for the performance of a steganalysis algorithm. Most state-of-the-art models yield low payloads such as 0.2 bpp and 0.4 bpp; we evaluate our model with payload capacities from the lowest payload of 0.05 bpp.
2) Improving feature learning: Using convolutional structures in DL frameworks proves advantageous in capturing the relationships between adjacent pixel values within an inquiry image. Nevertheless, CNNs typically amalgamate local layer data through techniques like pooling operations or convolutional layer scaling when incorporating global features. Therefore, in this work, we develop an algorithm that handles global information in the process, enhancing the effectiveness of the feature learning process.

The structure of the next parts of this article is outlined as follows: Section II introduces the pertinent existing research. In Section III, we elaborate on our methodology. Section IV presents a comprehensive set of experimental results to substantiate the efficacy and efficiency of our proposed approach. Finally, Section V serves as the conclusion of this article.

II. RELATED WORKS

A. FUZZY-BASED EDGE DETECTION ALGORITHMS

Fuzzy logic is a mathematical paradigm that addresses classification issues resulting in impreciseness and uncertainties with data. Fuzzy consists of intricate and ever-changing situations that find better characterization through descriptive language and nuanced interpretations rather than strict mathematical representations. Fuzzy logic has found its primary practical application as process controllers in numerous fields, particularly in Japan and Europe [18]. In recent applications, fuzzy logic has been widely applied in metaheuristic tasks [25], [26], regularizing the environment for smart agriculture [27], [28], covering preprocessing for enhanced payload capacity in steganographic applications [29], [30]. Unlike classical logic, which is based on sharp true-false differentiations, fuzzy logic aims to emulate human logic by performing representations in non-linear ways. Fuzzy logic frequently employs linguistic terms that deviate from conventional binary logic. Unlike the binary approach, fuzzy logic enables gradual representations within a continuous environment, facilitating the expression of varying levels of impreciseness.

Zadeh [19] invented fuzzy sets to address dissatisfaction with classical (crisp sets) sets. Fuzzy logic permits the utilization of set membership, allowing their elements to belong to one or more classes simultaneously based on the degree of membership. The scope of these sets is influenced by human reasoning, as it hinges on the concept or user implementing them. The fuzzy logic type 1 (FT1), which is selected in this work based on state-of-the-art works such as [21], consists of a set B from the universe U ranging from 0 to 1, which belongs to a function that is continuous mathematically expressed as $\mu_B{:}U \to [0, 1]$. Let B, the fuzzy membership function noted as $\mu_B(u)$, we mathematically express the function B as of (1).

$$B = \{(u,\ \mu_B(u))|\, u\epsilon U\} \tag{1}$$

The representation of a fuzzy set primarily uses one of the three membership functions (MFs), such as the Gaussian membership function (GMF), trapezoidal membership function (TraMF), and triangular membership function (TriMF). In this work, we use GMF to detect the edges of images with $\{a \text{ and } \partial\}$ the parameters used to explain our relation in (2) mathematically. a is used to represent the average membership function and ∂ represents the amplitude.

$$Gaussian\,(u;\, a, \partial) = e^{-\frac{1}{2}\left(\frac{u-a}{\partial}\right)^2} \tag{2}$$

Moreover, the logic of the fuzzy inference system is founded on the if-then rules based on fuzzy reasoning [23] implemented with fuzzy sets. A Fuzzy Inference System (FIS) comprises a database alongside a reasoning process that deduces a logical conclusion based on the inputs, outputs, and knowledge stored within the database. Well-known fuzzy inference systems include Tsukamoto, Takagi and Sugeno [31], and Mamdani [20].

Techniques for image processing encompass a range of digital image manipulations aimed at concealing or emphasizing details and targeted patterns, improving image light, and eliminating noise resulting from external factors such as camera sensor artefacts or motion during image capture. This process involves applying an operation within a pixel window (kernel) that traverses the images, uniformly altering their content to generate a new image [32]. Considering the variable f as an input image, for the edge's detection function in (3), k is a kernel with n rows and m columns.

$$g\,(u, y) = \sum\nolimits_{m=-i}^{i} \sum\nolimits_{c=-j}^{j} k(n, m) f(u+n, y+m) \tag{3}$$

Tasks like edge detection are executed to decrease the volume of data within an image. Techniques for edge detection are employed to recognize abrupt shifts in the brightness gradations of the image, enabling the detection of boundaries. The Roberts, Sobel, and Prewitt operators [33], [34], [35] stand out as widely used methods for edge detection.

TABLE 1. **Inputs to the function *f* to compute for edges in an iimage.**

Cartesian Coordinates		
(u-1, v-1)	(u-1, v)	(u-1, v+1)
(u, v-1)	(u, v)	(u, v+1)
(u+1, v-1)	(u+1, v)	(u+1, v+1)

TABLE 2. **Slopes (coefficients) associated with each matrix position.**

Cartesian Coordinates		
μ_1	μ_2	μ_3
μ_4	μ_5	μ_6
μ_7	μ_8	μ_9

These operators concentrate on computing the gradient of an image using the initial derivative. This is achieved by applying a convolution operation that estimates the gradient and provides the first derivative along the horizontal and vertical axes.

The conventional Sobel and Prewitt operators operate similarly. Each utilizes a 3 × 3 gradient operator within a local neighbourhood, as adapted from [36]. However, their distinction lies in the convolutional process, where they employ different masks. In (4) and (5), we establish the masks for the Prewitt operator in a convoluted manner applied to a grayscale image. These equations pertain to *Prewittu* and *Prewittv*, respectively. In comparison, the masks utilized within the Sobel operator are presented in (6) for *Sobelu* and (7) for *Sobelv*.

$$P_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

$$P_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{5}$$

$$S_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{6}$$

$$S_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{7}$$

The filter employs a pair of distinct kernels on an image to produce gradients. This leads to the gradient along the x-axis represented as g_u stated in (8) for horizontal orientation and the gradient along the y-axis noted as g_v stated in (9) for the vertical orientation. Table 1 shows the cartesian coordinates of the inputs to the function f to compute edges in an image. The coordinates in u represent the horizontal axis coordinates, and v represents the vertical axis coordinates. To compute for gradients in all axes, *kernelu* stands for the mask *sobelu* or *prewittu* and *kernelv* represents the mask *sobelv* or *prewittv*.

$$g_u = \sum\nolimits_{i=1}^{i=3} \sum\nolimits_{j=1}^{j=3} kernelu_{i,j} * f_{u+i-2,v+j-2} \tag{8}$$

$$g_v = \sum\nolimits_{i=1}^{i=3} \sum\nolimits_{j=1}^{j=3} kernelv_{i,j} * f_{u+i-2,v+j-2} \tag{9}$$

To derive the magnitude of gradients $G_{[f(u,v)]}$, we use the relation in (10), which encompasses the outcomes of computations involving g_u and g_v, which stem from the input to f via (8) and (9).

The Morphological Gradient stands as an edge-detection method that evaluates the initial derivative of an image across its four orientations: vertical, the diagonals (0°, 45°, 90°, and 135°), and horizontal. Illustrated in Fig. 2, the gradients are denoted by the variables G_1, G_2, G_3 and G_4. The procedure for computing these gradients is as follows: G_i (with i ranging from 1 to 4) signifies the edge's direction (gradient). This computation is executed using a 3 × 3 kernel following (11), (12), (13), and (14). In the (15), μ_i corresponds to the slope (coefficient) associated with each matrix position as depicted in Table 2, with f symbolizing the input representation, utilizing the x-axis for columns and the y-axis for rows. The value of the edge is denoted by the variable "*MG*," and it is computed following (16) [22].

$$G_{[f(u,v)]} = \sqrt{g_u^2 + g_v^2} \tag{10}$$

$$G_1 = \sqrt{(\mu_5 - \mu_2)^2 + (\mu_5 - \mu_8)^2} \tag{11}$$

$$G_2 = \sqrt{(\mu_5 - \mu_4)^2 + (\mu_5 - \mu_6)^2} \tag{12}$$

$$G_3 = \sqrt{(\mu_5 - \mu_1)^2 + (\mu_5 - \mu_9)^2} \tag{13}$$

$$G_4 = \sqrt{(\mu_5 - \mu_3)^2 + (\mu_5 - \mu_7)^2} \tag{14}$$

$$\mu_i = \begin{cases} \mu_{1=f(u-1,v-1)} \\ \mu_2 = f(u, v-1) \\ \mu_3 = f(u+1, v-1) \\ \mu_4 = f(u-1, v) \\ \mu_5 = f(u, v) \\ \mu_6 = f(u+1, v) \\ \mu_7 = f(u-1, v+1) \\ \mu_8 = f(u, v+1) \\ \mu_9 = f(u+1, v+1) \end{cases} \tag{15}$$

$$GM = G_1 + G_2 + G_3 + G_4 \tag{16}$$

B. STATE-OF-THE-ART IN SPATIAL DOMAIN IMAGE STEGANALYSIS

Tan and Li [37] introduced the initial instance of employing deep learning in steganalysis in 2014. Their method involved unsupervised learning using a series of Auto-Encoders to train a Convolutional Neural Network (CNN). Subsequently, supervised learning was applied by first employing a High Pass Filter (HPF) to preprocess the image. This step aimed to amplify the steganographic noise power introduced through the data concealment steps. The detection rates for stego images exhibited a reduction of around 17% compared to the

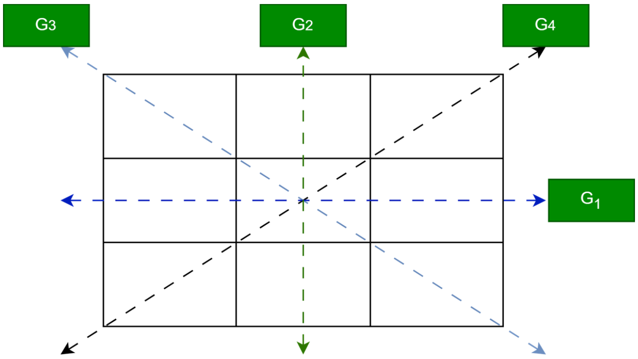

FIGURE 2. Directions for the considered gradients, G_1 represents the vertical gradient, G_2 represents the horizontal gradient, and both G_3 and G_4 represent the diagonal gradients.

results achieved through Spatial Rich Models (SRM) [38]. The rates were roughly 11% greater than those obtained using the Subtractive Pixel Adjacency Matrix (SPAM) [39]. Later, in 2015, Qian et al. [10] introduced the inaugural CNN utilizing a supervised learning methodology. This CNN architecture comprised five convolutional layers and featured a distinctive activation function called Gaussian Activation. The identification rates for steganographic images decreased by around 4% compared to the findings from SRM [38]. Moreover, these rates were approximately 10% higher than the results of utilizing the Subtractive Pixel Adjacency Matrix (SPAM) [39].

In 2016, Pibre et al. [40] built upon the groundwork laid by Qian, introducing two novel neural network architectures. The initial architecture featured a 2-layer CNN, while the second model consisted of a Fully Connected Neural Network (FNN) comprising two layers. Xu et al. [41] introduced a CNN architecture akin to Qian's, incorporating five convolutional layers. In contrast to the former model, Xu et al. introduced an additional absolute value layer (ABS) and employed a 1 × 1 convolutional kernel to enhance statistical modeling, resulting in improved outcomes. Taking their proposed CNN as a foundation, Xu et al. employed it as a foundation [42] to train datasets of CNNs. This approach aimed to attain enhanced training parameters and refine their detection outcomes. During that same year, Qian et al. explored Transfer Learning [43], involving the transfer of parameters from one CNN model, initially trained on stego images with substantial payload, to another network tailored for detecting stego images with small payloads. Although this approach yielded enhanced results compared to the previous models without Transfer Learning, it still fell short of outperforming conventional algorithms. These advancements were predominantly executed within the spatial domain. Following this, researchers shifted their attention to performing steganalysis using Deep Learning techniques within the frequency domain, explicitly focusing on the JPEG format.

In 2017, Zeng et al. [44] introduced a CNN-based model to conduct steganalysis on images in the JPEG format. They employed an approach influenced by RM for preprocessing, which was applied to extensive image collections sourced from ImageNet [45]. The results achieved closely paralleled the findings documented in the existing literature. Concurrently, Chen, Fridrich, and their team developed a fresh network utilizing Phase Split, drawing inspiration from the JPEG compression procedure [46]. Employing a CNN assembler, they achieved notably superior results compared to the prevailing state-of-the-art methods. Another noteworthy advancement involved the incorporation of transitions between distinct convolutional layers, drawing inspiration from models like ResNet [47], [48]. This technique facilitated the creation of more intricate CNN architectures, enhancing the convergence of networks and subsequently elevating detection precision. This progression led to an approximate 10% enhancement in detection outcomes compared to previously documented results.

In 2018, a novel CNN was introduced within the spatial domain by Yedroudj et al. [49]. CNN amalgamated the most favorable attributes of its predecessors. It integrated an array of input filters for preliminary processing, drawing inspiration from SRM's feature extraction. Furthermore, the model encompassed five convolutional layers, incorporated Batch Normalization, featured Truncation Linear Unit (TLU) activation units and expanded the training dataset's scale. These combined improvements yielded superior outcomes than those documented in the existing literature. In a subsequent work [50], Tsang et al. took Ye's CNN as a foundation and adapted it to facilitate the classification of high-resolution steganographic images. This adaptation involved training the network with low-resolution images. Boroumand et al. [47] introduced a novel CNN design to minimize the reliance on techniques like SRM filters in the preprocessing stage. CNN operates effectively in both the spatial and JPEG domains.

Similarly, Zhang et al. [51] proposed an innovative CNN architecture that optimizes the filters' weights in the preprocessing layer. This optimization strategy aims to enhance the strength of steganographic noise while concurrently diminishing the impact of the content in an image. The network employs distinct convolutions to independently capture residue channels and spatial correlations for improved feature representation. Additionally, the approach incorporates Spatial Pyramid Pooling (SPP) [52] to introduce local features. This augmentation enhances feature representation capabilities and enables accommodation for diverse image sizes.

In the recent five years, since 2019, several works have been proposed to improve the results of steganalysis in digital images, taking foundation in those primary works that have been done since 2014. Boroumand et al. [47] 2019 proposed a method to address the issue of hand-designed elements such as utilization of predetermined or limited convolutional kernels, heuristic initialization for kernel parameters, employment of threshold linear units to emulate truncation found in rich models, feature map quantization, and consideration of JPEG phase. Their work presents a profound residual structure crafted to mitigate the reliance on ad hoc techniques and externally imposed components. This comprehensive architecture delivers cutting-edge detection precision for both spatial-domain and JPEG-based steganography. The pivotal element of this devised architecture involves an extensively extended initial segment of the detector. This segment specializes in "computing noise residuals," where the pooling mechanism has been deactivated to avert the attenuation of the stego signal. Extensive experimental assessments underscore the exceptional prowess of this network, exhibiting remarkable enhancement, particularly within the realm of JPEG processing.

A further elevation in performance is noted by introducing the selection channel as an additional channel. This same year, Hu et al. [53] also proposed an innovative approach for self-directed steganalysis, leveraging visual attention and deep reinforcement learning to discern adaptive steganography within JPEG images. Initially, a visual attention mechanism was employed to designate a specific region within the image. Subsequently, through reinforcement learning, a continuous decision-making process is executed, creating a summarized region. This sequential methodology guides the deep learning model to concentrate on regions conducive to effective steganalysis while disregarding less informative regions. The outcomes encompass an enhanced quality of the training dataset and an augmented steganalysis detection capacity, achieved by substituting misclassified training images with their corresponding summarized regions. In 2020, Zhang introduced a new method based on learning selection channels one year later. Their method involves the holistic learning of selection channels in an integrated fashion. Their steganalysis framework encompasses two main components: the selection channel and the steganographic data detection networks. These two components are cohesively trained. The selection channel undertakes the task of identifying and outputting the selection channels employed by the steganalysis network. The latter, equipped with these learned selection channels, predicts the ultimate steganalysis outcomes. Through diverse experimental scenarios, their results illustrate a noteworthy enhancement in detection accuracy achieved by the acquired selection channels. This improvement is significantly pronounced when dealing with content-adaptive confidential data concealment.

In 2021, several steganalysis methods were proposed; among them, we can cite a model for the detection of spatial content-independent and content-adaptive steganographic algorithms through universal steganalysis, employing normalized features obtained from components derived via empirical mode decomposition has been proposed by Arivazhagan et al. [54]. Moreover, in 2022, in line with improving the results in the steganalysis of digital images using machine and deep learning methods, it continued to be much more interesting to researchers in information security. Fu et al. [55] proposed a novel CNN meticulously designed to amplify the potency of pertinent features, thereby augmenting the precision of detection within spatial domain steganalysis. The formulated model encompasses a triad of distinct modules: noise extraction, analysis, and classification. A pivotal element within the noise extraction and analysis modules is integrating a channel attention mechanism by incorporating SE (Squeeze-and-Excitation) modules into the residual blocks. Convolutional pooling is adopted instead of average pooling to refine feature aggregation further. Comprehensive empirical findings substantiate the pronounced efficacy of their model, surpassing the previously established counterparts like [6], [24], and [56] in steganographic payload detection accuracy.

Recently, several works have tried to improve the performance of steganalysis; nevertheless, few combine fuzzy logic and CNN. Referring to Subsection A of Section II, among the three main methods to identify the fuzzy edges in a digital image, namely Prewitt, Sobel, and morphological gradients, we chose to use the Prewitt method in this article because it showed a superior performance as referred to [36]. In [11] an algorithm has been proposed to detect the location of the steganographic data in digital images based on fuzzy correlation maps for classification based on the results of this work to detect steganography and also departing from the significance of the results that fuzzy logic has yielded in digital images classification [21], [36], [57], this work has been illuminated to use fuzzy logic as a preprocessing operation for images to be fed in the CNN for binary classification of images in

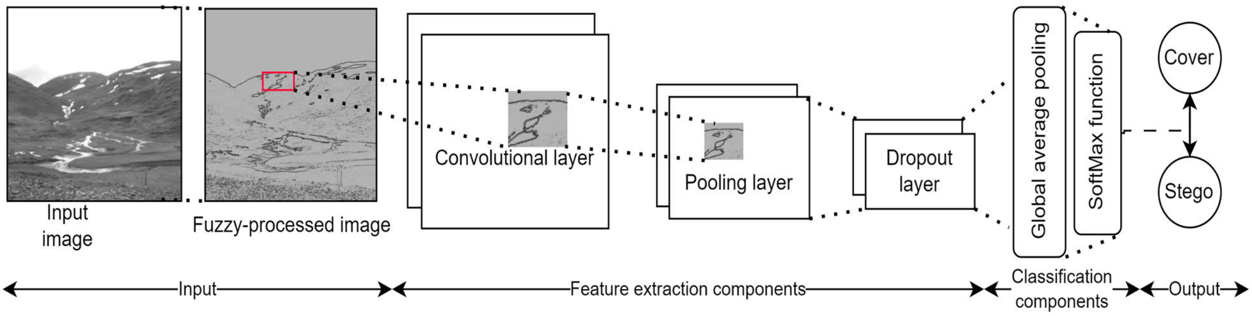

FIGURE 3. The general approach's depiction, the "Input" part consists of input preprocessing which contributes to overcoming the issues of the training dataset quality and quantity, and the remaining parts contribute in classification performance by mainly addressing the issues of both the poor features' learnability and the low payload detection.

cover or steganography holding. The position of this work, unlike [11], is to use Prewitt fuzzy edges identification to preprocess the inquiry images that are then used for training, validating, and testing our CNN.

III. PROPOSED METHOD

The main objective of this method is to provide a steganalysis model that outperforms the existing steganalysis methods to detect the presence of hidden data. This work combines fuzzy logic and CNN to classify the inquiry images into cover or stego images. Among our approaches, we include filters that utilize fuzzy edge-detection principles. These (termed edge-detection filters in this instance) enable assessing a pixel's association with an image's boundary or a consistent area while factoring in indistinctness. The general approach's depiction can be seen in Fig. 3 and is detailed in the subsequent description.

A. PREPROCESSING IMAGES WITH FUZZY LOGIC

This section explains the membership functions we use, the fuzzy rules, and the strategy employed to execute the approaches for fuzzy edge detection.

1) EDGE-DETECTION VIA FUZZY PREWITT APPROACH

The process of obtaining the fuzzy Prewitt edge-detection technique resembles the Sobel approach, contingent upon the chosen operator for application. This alteration only involves the mask, as identified earlier in Subsection A of Section II and denoted through the relations in (4) – (10). Nevertheless, this study adopted the Prewitt method due to its superior performance, demonstrated in [36]. Our inference system comprises fuzzy type-I with Mamdani fuzzy inference (FIS), featuring a pair of input and a single output membership function with three fuzzy rules. The overall model for our method is detailed by Algorithm 1 and elaborated in the following steps, underscoring that the depicted numerical outcomes, serving as an illustration, are computed utilizing the fuzzy Prewitt technique.

1) Step 1: Getting an input image: The input images taken by seven cameras are obtained from the commonly used dataset adopted from [58]. Some sample images are illustrated in Fig. 4 to showcase the various textures of images from the source dataset.
2) Step 2: Acquire the data required as inputs for the FIS: We consider fuzzy type -1 inference with two input variables, namely, the gradient along the x-axis (g_u) got by (8), and the gradient along the y-axis (g_v) got by (9). The labels for the input membership functions are D_u and D_v for the g_u and g_v respectively. The pair of inputs correspond to Gaussian membership functions as formulated in (3). The D_u input is discretized into three membership functions, each associated with the linguistic terms: '$Low - D_u$', '$Mid - D_u$', '$High - D_u$'; similarly, the D_v input is characterized by three membership functions denoted as: '$Low - D_v$', '$Mid - D_v$', '$High - D_v$'. The parameters are established based on the gradients of each image: lower values are computed through (17), higher values are obtained via (18), intermediate values are derived from (19), and σ values for the D_u and D_v gradients are determined using (20).

TABLE 3. Rules knowledge base for the proposed model's edges detection.

Input 1: *Du*	Operation	Input 1: *Du*	Output
$Low - D_u$	And	$Low - D_u$	Background
$Mid - D_u$	Or	$Mid - D_u$	Edge
$High - D_u$	Or	$High - D_u$	Edge

$$Low - value = \begin{cases} Low - D_u = \min(D_u) \\ Low - D_v = \min(D_v) \end{cases} \tag{17}$$

$$High - value = \begin{cases} High - D_u = \max(D_u) \\ High - D_v = \max(D_v) \end{cases} \tag{18}$$

$$Mid - value = \begin{cases} Mid - D_u \\ \quad = \dfrac{(Low - D_u + High - D_u)}{4} \\ Mid - D_v \\ \quad = \dfrac{(Low - D_v + High - D_v)}{4} \end{cases} \tag{19}$$

$$\sigma = \begin{cases} \sigma - D_u = \dfrac{High - D_u}{4} \\ \sigma - D_v = \dfrac{High - D_v}{4} \end{cases} \tag{20}$$

The Gaussian membership function parameters related to the gradient along the x-axis, namely, $\mu Low - D_u(x)$, $\mu Middle - D_u(x)$, and $\mu High - D_u(x)$ are obtained through the formulae (21)–(23) and visually represented in Fig. 5. The specific values obtained are depend on: $Low - D_u = 0$, $Mid - D_u = 127.50$, $High - D_u = 255$, and $\sigma - D_u = 127.50$ for the D_u input.

$$\mu Low - D_u(x) = \text{Exp}\left[-\frac{1}{2}\left(\frac{x-0}{127.50}\right)^2\right] \tag{21}$$

$$\mu Middle - D_u(x) = \text{Exp}\left[-\frac{1}{2}\left(\frac{x-127.50}{127.50}\right)^2\right] \tag{22}$$

$$\mu High - D_u(x) = \text{Exp}\left[-\frac{1}{2}\left(\frac{x-255}{127.50}\right)^2\right] \tag{23}$$

The Gaussian membership functions pertaining to the gradient along the y-axis, namely, $\mu Low - D_v(y)$, $\mu Middle - D_v(y)$, and $\mu High - D_v(y)$ are formulated within (24)–(26) and visually depicted in Fig. 6. The fundamental values used are as follows: $Low - D_v = 0$, $Mid - D_v = 127.50$, $High - D_v = 255$, and $\sigma - D_v = 127.50$.

$$\mu Low - D_v(y) = \text{Exp}\left[-\frac{1}{2}\left(\frac{y-0}{127.50}\right)^2\right] \tag{24}$$

$$\mu Middle - D_v\left(y\right) = \text{Exp}\left[-\frac{1}{2}\left(\frac{y-127.50}{127.50}\right)^2 \right] \quad (25)$$

$$\mu High - D_v\left(y\right) = \text{Exp}\left[-\frac{1}{2}\left(\frac{y-255}{127.50}\right)^2 \right] \quad (26)$$

3) Step 3: Generating the output: The FIS comprises a single output designated as "Edges," which is split into two linguistic designations: "Background" and "Edge." In the context of this study, the output (Edges) is subjected to normalization within a spectrum ranging

FIGURE 4. Sample input images with various textures got from the Break Our Steganographic System Base version 1. 01 [58].

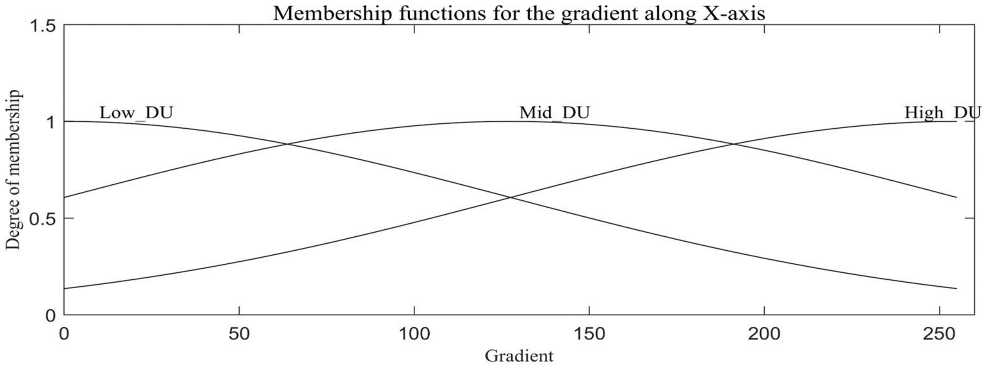

FIGURE 5. D_u (Gradient along x-axis) input membership function.

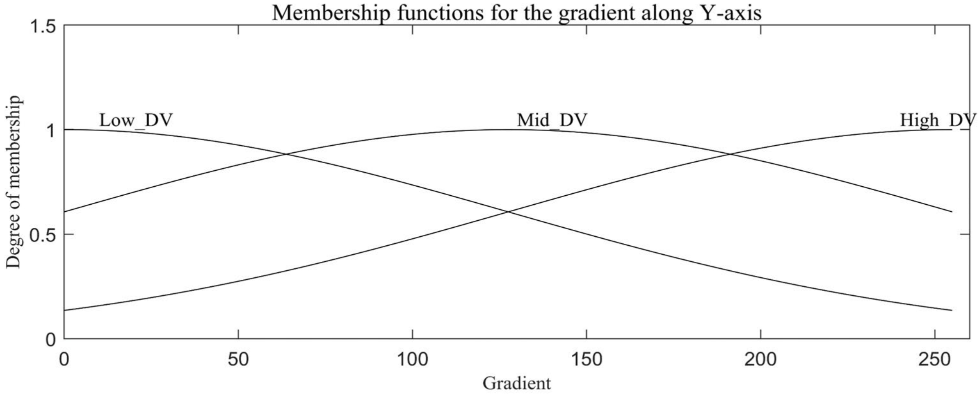

FIGURE 6. D_v (Gradient along y-axis) input membership function.

from 0 to 1. The pivotal value for the Background membership function is indicated as α Background = 0, while for the Edge membership function, it is determined as αEdge = 1. The σ value for both membership functions is computed using (27).

$$\sigma_{Output} = \frac{abs(\alpha Background - \alpha Edge)}{2} \tag{27}$$

The parameters pertaining to the output membership function are expressed within the relations (28) and (29), accounting for the specific values of $\sigma Background = 0$, $\sigma Edge = 1$, and $\sigma_{output} = 1$.

$$\mu\_Background_{(x)} = Exp\left[-\frac{1}{2}(\frac{\times - (0)}{1})^2\right] \tag{28}$$

$$\mu\_Edge_{(x)} = Exp\left[-\frac{1}{2}(\frac{\times - 1)}{1})^2\right] \tag{29}$$

4) Step 4: Generating our FIS's fuzzy rules: The conceptual framework is depicted using three fuzzy rules (refer to Table 3).

B. CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORK

The architecture of the proposed CNN is summarized in Appendix II. Our CNN is based on three stages: CNN's preprocessing stage, the stage for feature extraction, and the classification stage. The following subsections describe the three main parts of our CNN illustrated by Fig. 8 as presented in the Appendix Section.

1) PREPROCESSING PART

In this phase, convolution is performed using 30 filters [6], [38], [59], with dimensions (5, 5). These filters remain unaltered during the training phase, rendering the layer non-trainable. The arrangement of the convolutional layers during this phase is as such: maintaining "same" padding and utilizing strides of (1, 1). This setup utilizes 30 filters, the details of which are expounded upon later, and employs a **3 × *TanH*** activation function mathematically expressed in (30).

$$3 \times TanH\ (a) = 3 \times \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{30}$$

YE-Net's [45] incorporation of 30 filters was employed for image preprocessing within the structure. These filters have showcased notable performance in priming the images for ensuing feature extraction efficiency. Normalization of these 30 filters is carried out based on the highest absolute value inherent to each filter.

Existing networks in [49] and [59] adopt the TLU activation function on their initial layer. Nonetheless, its efficacy is not universal across all architectural setups, so the activation function was not used in [6] and [24]. Referring to the experimentation for the tests involving the TLU, ReLU, and TanH functions conducted in [56], where the best outcomes were attained through the utilization of the TanH activation function, we consider using this same function scaled by a factor of three, coupled with the specification that the first layer remains non-trainable. TLU and TanH exhibit analogous forms; nevertheless, TanH displays a more gradual curve. The performance with the ReLU function lacked significance. Consequently, during the preprocessing phase, the selected activation function is 3TanH, operating within the range of -3 to 3, which yields optimal performance.

2) FEATURE EXTRACTION PART

Within this phase, different layers used include 2-dimensional convolutional layers (2D-Convs.), 2-dimensional separable convolutional layers (2D Sep-Convs.), and 2-dimensional depth-wise convolutional layers (2D Dep-Convs.). Each layer is flexible to fine-tune the parameters and filters to improve the network's performance. Moreover, this stage integrates shortcuts employing addition, and following Batch Normalization (BN), Average Pooling (Avg-pool) operations were introduced to diminish dimensionality, structured with a pool size of (2, 2) and strides set at (2, 2). We use six convolutional layers with filters measuring (3, 3) while terminating in this phase are two additional layers utilizing a kernel size of (1, 1). We use the exponential linear unit (ELU) mathematically expressed in (31) in all convolutional and separable layers.

$$ELU\ (a) = \begin{cases} a & if\ a > 0 \\ \alpha\left(e^a - 1\right) & if\ a \leq 0 \end{cases} \tag{31}$$

The hyperparameter of the ELU function (α) handles the saturation values of the ELU to manipulate the negative inputs. In this scenario, it was established at a value of 1. Specific attributes of this activation function mitigate the vanishing gradients and tend to approach negative saturation as the argument diminishes. Strides of (1, 1) and the same padding are applied across all convolutional operations. In this phase, the initial pair of 2D-Convs employs 30 filters, followed by four subsequent layers with 60 filters each.

Moreover, 2D Sep-Convs are embedded within the network within the shortcuts, characterized by 30 and 60 filters, a (3, 3) shaped kernel size, strides of (1, 1), uniform padding, and a 3 × depth multiplier. Preceding each 2D Sep-Conv layer is a 2D Dep-Conv layer characterized by a kernel size of (1, 1). Upon the culmination of this stage, a global average pool is executed to prime the features for the subsequent classification process.

3) CLASSIFICATION PART

The classification phase simplifies the outcome derived from the global average pooling layer. Moreover, streamlining this phase involves omitting dense layers, a strategy that mitigates the risk of overfitting. The ultimate Batch Normalization, featuring dimensions of 16 × 16 × 2, is succeeded by a 2D global average pooling process that yields two values. Subsequently, predictions are derived utilizing the SoftMax function.

IV. EXPERIMENTATION AND RESULTS

A. PREPROCESSING IMAGES WITH FUZZY LOGIC

We employ popular content-adaptive techniques, namely Minimizing the Powerful Detector (MiPOD), Spatial UNIversal WAvelet Relative Distortion (S-UNIWARD), and Wavelet Obtained Weights (WOW), for embedding within the spatial domain. We implement these methods using their Matlab versions alongside Syndrome Trellis Codes (STCs). Our simulation involves utilizing a distinct random key for each embedding process, avoiding misuse of the C++ codes. This approach prevents using a static and uniform embedding key, as outlined in [40]. We evaluate our steganalysis model against existing cutting-edge techniques. This includes a comparison with the results achieved in [6], [24], and [56]. To ensure an equitable assessment, we conduct tests for all these steganalysis methodologies using identical subsampled images sourced from the well-known database Break Our Steganographic System Base version 1.01 (BOSSBase v.1.01) [58].

Because of the constraints posed by our GPU-based computing infrastructure and the limited time available, our experimentation was carried out on images sized 256 × 256 pixels. This approach aligns with the methodology followed in previous studies [9], [24]. We resized all the original 512 × 512 images to the target size of 256 × 256 using the *imresize*() function within the Matlab software suite, utilizing its default settings. Our resized images are the ones we use in data embedding, and after getting the stego images, we preprocess the inquiry images comprising the cover and stego images with fuzzy edge-detection using the Matlab fuzzy toolbox. The partition of our 256 × 256 dataset is such that we use 50% of the 10000 pairs (covers/ stego pairs) for the training phase, 40% for the testing phase, and 10% for the validation phase.

B. HYPER-PARAMETERS SETTING FOR THE CNN

We employ a batch size of 16 in CNN to optimize the available resources, and the network's training process necessitates 50 epochs to learn from the provided payload effectively. During this training, the chosen optimizer is Adam, configured with specific parameters: a learning rate of 0.001, a beta 1 value of 0.9, a beta 2 value of 0.999, a decay rate of 0.0, and an epsilon value set at 1e-08. Besides the initial preprocessing layer, the convolutional layers employ a kernel initializer termed "glorot uniform." Within the CNN architecture, a categorical cross-entropy loss function is applied to accommodate the classification of the two distinct classes.

The Batch Normalization configuration incorporates specific settings: a momentum value of 0.2 and an epsilon setting of 0.001. The parameters include a 'center' parameter set to True and a 'scale' parameter set to False, both of which are trainable. The 'fused' parameter remains at its default value of None, while 'renorm' is set to False, without any renormalization clipping. The momentum for renormalization is established at 0.4, and no adjustment is applied to this configuration. Each of the 30 high-pass SRM filters undergoes normalization using the maximum absolute value. A padding set to 'same' is employed across all layers similarly.

TABLE 4. Recall rate results obtained with the proposed method.

Targeted Algorithm Payload Size (in bpp)	MiPOD	S-UNIWARD	WOW
0.05	50.2	59.0	50.0
0.1	74.9	69.8	76.8
0.2	96.1	80.9	90.0
0.3	90.2	88.0	89.0
0.4	98.4	93.2	96.4
0.5	98.2	95.0	99.8

TABLE 5. F1-score results obtained with the proposed method.

Targeted Algorithm Payload Size (in bpp)	MiPOD	S-UNIWARD	WOW
0.05	48.2	49.8	50.4
0.1	68.7	65.9	74.1
0.2	85.3	80.8	88.8
0.3	88.1	86.0	92.2
0.4	95.2	92.8	94.8
0.5	94.6	96.6	99.3

TABLE 6. Accuracy results obtained with the proposed method.

Targeted Algorithm Payload Size (in bpp)	MiPOD	S-UNIWARD	WOW
0.05	50.6	49.8	52.2
0.1	70.9	68.8	76.0
0.2	86.8	81.6	92.0
0.3	89.0	86.2	94.0
0.4	96.4	94.8	98.6
0.5	97.2	96.6	99.6

C. EVALUATION METRICS

To comprehensively assess the effectiveness of our method, we analyze three distinct metrics: The sensitivity ($R_{(i)}$) otherwise called recall rate, the classification accuracy ($Accuracy_{(i)}$), and F1-score ($F1 - score_{(i)}$), which provides a balanced assessment of our CNN's precision and recall for the binary classification task.

Defining True Positive (TP) as the count of stego images correctly classified as stego, False Positive (FP) as the count of cover images inaccurately classified as stego, True Negative (TN) as the count of cover images correctly classified as covers, and False Negative (FN) as the count of stego images inaccurately classified as covers, the $R_{(i)}$ is computed following (32), the $Accuracy_{(i)}$ is computed from (33), and the $F1-score_{(i)}$ is got from the relation (34).

$$R_{(i)} = \frac{TP}{TP+FN} \times 100\% \tag{32}$$

$$Accuracy_{(i)} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \tag{33}$$

$$F1-score_{(i)} = \frac{2TP}{2TP+FP+FN} \tag{34}$$

D. RESULTS

To assess the effectiveness of the suggested models, we put them into practice using Matlab for the fuzzy logic side and Python for the CNN side, utilizing the TensorFlow framework in conjunction with the Keras API. To evaluate our approach against the state-of-the-art methods, we present outcomes aligned with the metrics specified in Sub-section C of Section IV. This enables us to make a comparative assessment and ascertain qualitative and quantitative enhancements. In Table 4, we record the obtained results in terms of the recall rate (R(i)) to detect the steganographic payloads with sizes ranging from 0.05 bpp to 0.5 bpp under the adaptive steganographic algorithms, namely, MiPOD, S-UNIWARD, and WOW. Tables 5 and 6 contain the results obtained in F1-score and accuracy, respectively. Based on the results obtained regarding the recall rate, as reported in Table 4, it is worth noting that our model effectively detects hidden data in WOW, particularly at higher payload sizes. The detection of the hidden data under MiPOD also performs well, especially at higher payload sizes, while detection of the secret data concealed with the S-UNIWARD is competitive at lower payload sizes but becomes less effective as the payload size increases.

Based on the data in Table 5, it is important to note that with our method, detecting the steganographic payload hidden under WOW outperforms those with both MiPOD and S-UNIWARD across all payload sizes, as indicated by its higher F1-Scores. MiPOD performs well, particularly at higher payload sizes, while S-UNIWARD competes more effectively as the payload size increases but lags WOW in most cases. It is also worth noting that the data in Table 6 show that our method achieves outperforming results in detecting the data embedded using WOW by consistently outperforming the detection of the data embedded using both MiPOD and S-UNIWARD across all payload sizes, as indicated by the achieved higher accuracy values. The detection of MiPOD performs reasonably well, particularly at larger payload sizes, while detecting those hidden under S-UNIWARD competes more effectively as the payload size increases but still lags WOW in most cases.

TABLE 7. Accuracy results obtained with the proposed method under different types of images.

	Targeted Algorithm / Payload Size (in bpp)	MiPOD	S-UNIWARD	WOW
With non preprocessed images	*0.05*	48.8	46.6	50.0
	0.1	72.4	68.8	76.0
	0.2	84.4	79.1	90.0
	0.3	87.9	85.4	91.3
	0.4	94.0	93.2	94.9
	0.5	96.2	95.7	97.3
With the images preprocessed with the proposed fuzzy edge detection-	*0.05*	50.6	49.8	52.2
	0.1	70.9	68.8	76.0
	0.2	86.8	81.6	92.0
	0.3	89.0	86.2	94.0
	0.4	96.4	94.8	98.6
	0.5	97.2	96.6	99.6

TABLE 8. Comparison of our results to the ones reported from the state-of-the-art methods.

	Targeted Algorithm / Payload Size (in bpp)	S-UNIWARD	WOW
Results in [6]	*0.2*	79.3	90.2
	0.4	93.1	94.4
Results in [24]	*0.2*	71.4	76.9
	0.4	80.5	84.1
Results in [57]	*0.2*	73.6	76.9
	0.4	87.1	84.1
Results with our method	*0.2*	81.6	92.0
	0.4	94.8	98.6

To demonstrate the effectiveness of the fuzzy preprocessing operation proposed in our method, we conduct comparative experimentation to show the results obtained with the proposed CNN when working with images with and without fuzzy preprocessing. Table 7 contains the obtained results in terms of accuracy for an ablation study

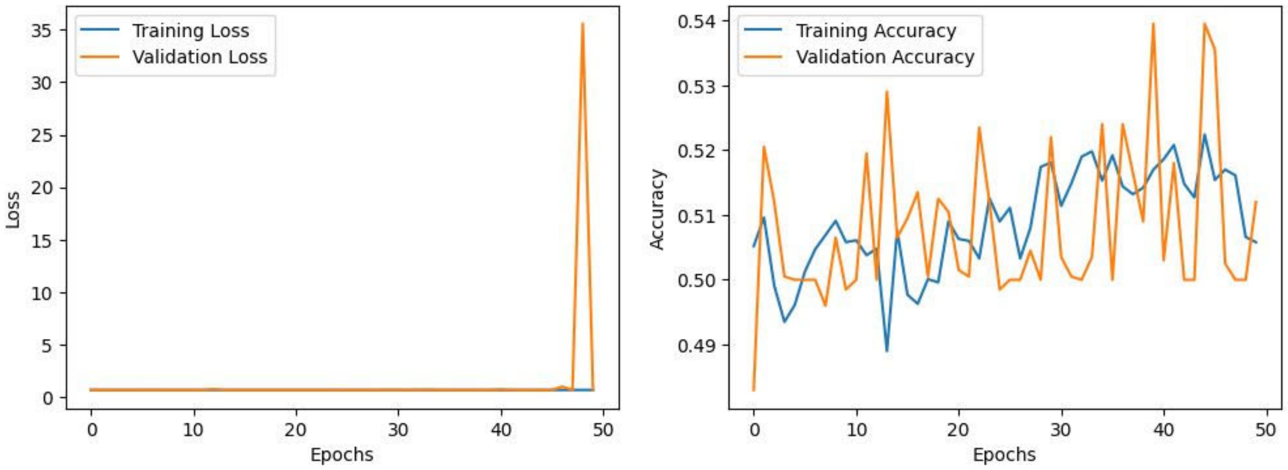

FIGURE 7. Training and validation loss and accuracy graphs for detecting the payload of size 0.05 bpp under WOW steganographic algorithm.

Algorithm 1 Algorithm to Detect the Edges With Our Model

```
Choosing the gradient calculating operator
  Utilizing Prewitt kernels (Referring to (4) for $Prewitt_u$ and (5) for $Prewitt_v$)
Getting the input inquiry image (here labeled as $f$)
Reading the dimensions (columns and rows) of the inquiry image $f$
    $[columns, rows] = size(f)$
Computing the conventional gradient associated with the preferred fuzzy filter.
$[D_u, D_v] = zero(rows, columns)$// Creating a matrix initialized with zeros, matching the //dimensions of' f ', to hold the gradient values.
    for $i = 0$ to (the total number of rows);
        for $j = 0$ to (the total number of columns);
            $D_u[i, j] = \sum kernel_u \times f[i : i + 3, j : j + 3]$ // Referring to (8)
            $D_v[i, j] = \sum kernel_v \times f[i : i + 3, j : j + 3]$ // Referring to (9)
        end
    end
Formulating the necessary fuzzy controller by utilizing the fuzzy rules outlined in Table 3
Applying Gaussian membership functions (as described in (21)-(26)) to perform the fuzzification of the pair of input gradients, $D_u$ and $D_v$
Deriving the output Edges using the chosen controller
Rendering the output of the controller crisp by removing fuzziness (Output Edges defuzzification)
```

of our method. The table contains the yielded results when using the images without any preprocessing operation before being fed to the CNN and the achieved results with images preprocessed with our method based on the fuzzy Prewitt paradigm. It is worth noting that, as generally proved by the previously discussed results in the previous tables, the detection of WOW is consistently superior to that of both MiPOD and S-UNIWARD across both scenarios, demonstrating its robustness. MiPOD and S-UNIWARD benefit from the proposed fuzzy edge detection-based preprocessing method, but WOW still exhibits the highest accuracy, which justifies the efficiency of the proposed method to detect the steganographic payload.

To compare our results with the results obtained in the recent works in [6], [24], and [56] performance over the state-of-the-art, we present in Table 8 a consolidation of the results obtained with our method and the results reported in the previous works under S-UNIWARD and WOW algorithms with the payload capacities 0.2 and 0.4 bits per pixel. The table data demonstrate that the results obtained with "our method" consistently outperform the ones reported in the state-of-the-art methods referenced in [6], [24], and [56]. This suggests that "our method" is competitive and may represent an improvement in spatial domain image steganalysis compared to the state-of-the-art methods mentioned in the references.

V. CONCLUSION

Utilizing CNNs, rather than relying on traditional handcrafted features and an ensemble classifier trained on the Rich Model, presents a notably superior performance for steganalysis researchers.

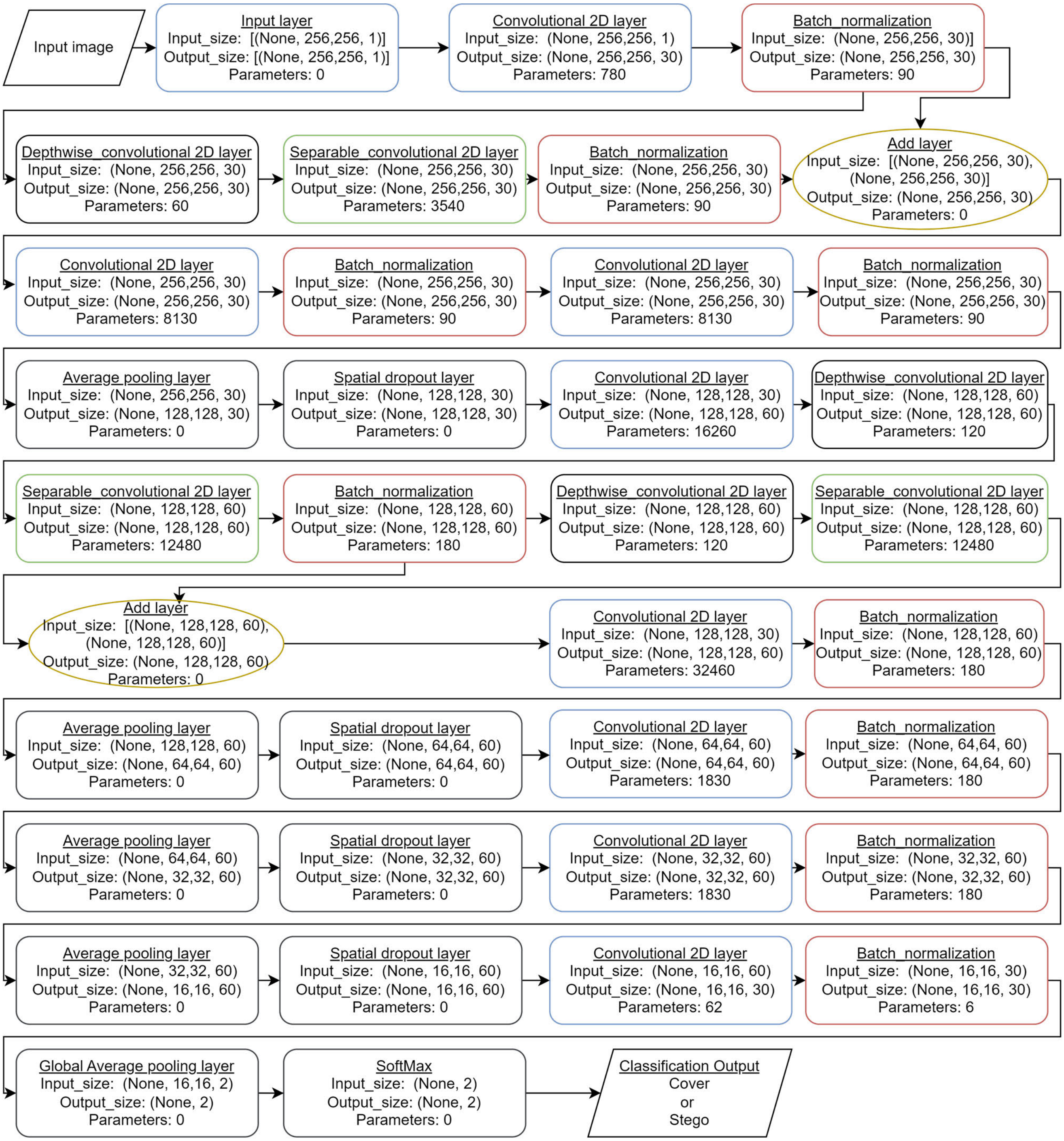
Input image
Input layer
Input_size: [(None, 256,256, 1)]
Output_size: [(None, 256,256, 1)]
Parameters: 0
Convolutional 2D layer
Input_size: (None, 256,256, 1)
Output_size: (None, 256,256, 30)
Parameters: 780
Batch_normalization
Input_size: (None, 256,256, 30)]
Output_size: (None, 256,256, 30)
Parameters: 90
Depthwise_convolutional 2D layer
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 60
Separable_convolutional 2D layer
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 3540
Batch_normalization
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 90
Add layer
Input_size: [(None, 256,256, 30),
(None, 256,256, 30)]
Output_size: (None, 256,256, 30)
Parameters: 0
Convolutional 2D layer
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 8130
Batch_normalization
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 90
Convolutional 2D layer
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 8130
Batch_normalization
Input_size: (None, 256,256, 30)
Output_size: (None, 256,256, 30)
Parameters: 90
Average pooling layer
Input_size: (None, 256,256, 30)
Output_size: (None, 128,128, 30)
Parameters: 0
Spatial dropout layer
Input_size: (None, 128,128, 30)
Output_size: (None, 128,128, 30)
Parameters: 0
Convolutional 2D layer
Input_size: (None, 128,128, 30)
Output_size: (None, 128,128, 60)
Parameters: 16260
Depthwise_convolutional 2D layer
Input_size: (None, 128,128, 60)
Output_size: (None, 128,128, 60)
Parameters: 120
Separable_convolutional 2D layer
Input_size: (None, 128,128, 60)
Output_size: (None, 128,128, 60)
Parameters: 12480
Batch_normalization
Input_size: (None, 128,128, 60)
Output_size: (None, 128,128, 60)
Parameters: 180
Depthwise_convolutional 2D layer
Input_size: (None, 128,128, 60)
Output_size: (None, 128,128, 60)
Parameters: 120
Separable_convolutional 2D layer
Input_size: (None, 128,128, 60)
Output_size: (None, 128,128, 60)
Parameters: 12480
Add layer
Input_size: [(None, 128,128, 60),
(None, 128,128, 60)]
Output_size: (None, 128,128, 60)
Parameters: 0
Convolutional 2D layer
Input_size: (None, 128,128, 30)
Output_size: (None, 128,128, 60)
Parameters: 32460
Batch_normalization
Input_size: (None, 128,128, 60)
Output_size: (None, 128,128, 60)
Parameters: 180
Average pooling layer
Input_size: (None, 128,128, 60)
Output_size: (None, 64,64, 60)
Parameters: 0
Spatial dropout layer
Input_size: (None, 64,64, 60)
Output_size: (None, 64,64, 60)
Parameters: 0
Convolutional 2D layer
Input_size: (None, 64,64, 60)
Output_size: (None, 64,64, 60)
Parameters: 1830
Batch_normalization
Input_size: (None, 64,64, 60)
Output_size: (None, 64,64, 60)
Parameters: 180
Average pooling layer
Input_size: (None, 64,64, 60)
Output_size: (None, 32,32, 60)
Parameters: 0
Spatial dropout layer
Input_size: (None, 32,32, 60)
Output_size: (None, 32,32, 60)
Parameters: 0
Convolutional 2D layer
Input_size: (None, 32,32, 60)
Output_size: (None, 32,32, 60)
Parameters: 1830
Batch_normalization
Input_size: (None, 32,32, 60)
Output_size: (None, 32,32, 60)
Parameters: 180
Average pooling layer
Input_size: (None, 32,32, 60)
Output_size: (None, 16,16, 60)
Parameters: 0
Spatial dropout layer
Input_size: (None, 16,16, 60)
Output_size: (None, 16,16, 60)
Parameters: 0
Convolutional 2D layer
Input_size: (None, 16,16, 60)
Output_size: (None, 16,16, 30)
Parameters: 62
Batch_normalization
Input_size: (None, 16,16, 30)
Output_size: (None, 16,16, 30)
Parameters: 6
Global Average pooling layer
Input_size: (None, 16,16, 2)
Output_size: (None, 2)
Parameters: 0
SoftMax
Input_size: (None, 2)
Output_size: (None, 2)
Parameters: 0
Classification Output
Cover
or
Stego

FIGURE 8. Architecture of the proposed CNN.

This paper focuses on designing a steganalysis model combining the fuzzy Prewitt approach with CNN to improve the detection accuracy of the steganalysis methods against the steganographic algorithms. The advantageous contribution of the proposed method focuses on (1) Improving the detection of low payload steganography by working with the edges of the inquiry images, which enhances performance through refined training sample selection and learning approaches in the steganalysis algorithm. We assess our model's performance at even lower payload capacities, such as 0.05 bpp. (2) Enhancing feature learning by introducing an algorithm that seamlessly incorporates global information into the process, thereby improving the efficiency of feature learning. (3) Mitigating the challenges posed by images of diverse data complexities and enhancing the extraction of valuable data from the dataset images by elevating the dataset's quality through preprocessing using fuzzy logic. (4) Tackling the challenge of an extensive training dataset by employing the fuzzy Prewitt approach to preprocess the inquiry images and obtain an efficient steganalysis framework capable of performing effectively even when dealing with a restricted number of training samples, such as employing just one dataset.

The experimental results demonstrate that the proposed method is promising in addressing the problems we aimed at, namely the dataset quality and size, the feature learning process, and the detection of low payload capacity (See Fig. 7, where the accuracy of the model to detect the considered lowest payload capacity, 0.05 bpp in WOW algorithm, achieves a maximum of 52 %). To sum up, it is worth noting that the proposed method outperforms the state-of-the-art methods in terms of a considered evaluation metric, accuracy, as of Table 8.

In our future research endeavours, we intend to implement the method we have put forth on alternative datasets, such as those comprising real-time images and images of arbitrary size. This will enable us to investigate and assess the performance of our model in addressing different image classification challenges. Furthermore, we aspire to enhance the same model to pinpoint the exact altered pixels within stego images by combining certain features from our approach with techniques proposed in [11].

APPENDIX

In this Section, we present detailed information regarding the two capital components of our research. Appendix I includes a detailed description of the proposed algorithm to detect the edges of an image. Appendix II comprises an architecture of the proposed CNN with in-depth details of our new architecture, which encompasses all the components of our model to detect the images altered by the addition of any steganographic payload.

A. APPENDIX I

Within this Subsection, we present a comprehensive description of our novel algorithm to detect the edges of images entitled Algorithm 1. This algorithm, pertaining to our preprocessing phase, plays one of the central roles in our contribution by solving the two issues with the datasets, namely, the quality and the quantity of the training samples.

B. APPENDIX II

Within this Subsection, we illustrate a visual representation of the proposed method's design and architecture labeled as Fig. 8. Detailed explanations are provided in Section III of this work entitled 'Proposed Method.' This architecture showcases in depth all technical aspects that contribute to realizing our contributions, namely, the feature learning optimization and the ability to detect low payload steganography, as evidenced in the results illustrated in Fig. 7.

ACKNOWLEDGMENT

The authors would like to thank all the laboratory and research group members for their valuable contributions and support.

REFERENCES

[1] A. J. Ilham, T. Ahmad, N. J. D. L. Croix, P. Maniriho, and M. Ntahobari, "Data hiding scheme based on quad general difference expansion cluster," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 12, no. 6, p. 2288, Nov. 2022, doi: 10.18517/ijaseit.12.6.16002.

[2] N. J. D. L. Croix, C. C. Islamy, and T. Ahmad, "Reversible data hiding using pixel-value-ordering and difference expansion in digital images," in *Proc. IEEE Int. Conf. Commun., Netw. Satell. (COMNETSAT)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Nov. 2022, pp. 33–38, doi: 10.1109/COMNETSAT56033.2022.9994516.

[3] I. B. Prayogi, T. Ahmad, N. J. D. L. Croix, and P. Maniriho, "Hiding messages in audio using modulus operation and simple partition," in *Proc. 13th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Oct. 2021, pp. 51–55, doi: 10.1109/ICTS52701.2021.9609028.

[4] M. M. Amrulloh and T. Ahmad, "Fuzzy logic and the greatest common divisor on audio-based data hiding method," *Int. Rev. Model. Simul.*, vol. 15, no. 3, p. 172, Jun. 2022, doi: 10.15866/iremos.v15i3.22235.

[5] T. Ahmad and A. N. Fatman, "Improving the performance of histogram-based data hiding method in the video environment," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1362–1372, Apr. 2022, doi: 10.1016/j.jksuci.2020.04.013.

[6] J. D. L. C. Ntivuguruzwa and T. Ahmad, "A convolutional neural network to detect possible hidden data in spatial domain images," *Cybersecurity*, vol. 6, no. 1, p. 23, Sep. 2023, doi: 10.1186/s42400-023-00156-x.

[7] M. S. Hossen, T. Ahmad, and N. J. D. L. Croix, "Data hiding scheme using difference expansion and modulus function," in *Proc. 2nd Int. Conf. Innov. Technol. (INOCON)*, Mar. 2023, pp. 1–6, doi: 10.1109/INOCON57975.2023.10100991.

[8] S. T. Veena and S. Arivazhagan, "Quantitative steganalysis of spatial LSB based stego images using reduced instances and features," *Pattern Recognit. Lett.*, vol. 105, pp. 39–49, Apr. 2018, doi: 10.1016/j.patrec.2017.08.016.

[9] N. J. D. L. Croix and T. Ahmad, "Toward hidden data detection via local features optimization in spatial domain images," in *Proc. Conf. Inf. Commun. Technol. Soc. (ICTAS)*, Mar. 2023, pp. 1–6, doi: 10.1109/ICTAS56421.2023.10082736.

[10] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Proc. SPIE*, vol. 9409, Mar. 2015, Art. no. 94090J, doi: 10.1117/12.2083479.

[11] N. J. D. L. Croix and T. Ahmad, "Toward secret data location via fuzzy logic and convolutional neural network," *Egyptian Informat. J.*, vol. 24, no. 3, Sep. 2023, Art. no. 100385, doi: 10.1016/j.eij.2023.05.010.

[12] M. Dalal and M. Juneja, "Steganography and steganalysis (in digital forensics): A cybersecurity guide," *Multimedia Tools Appl.*, vol. 80, no. 4, pp. 5723–5771, Feb. 2021, doi: 10.1007/s11042-020-09929-9.

[13] J. Lopez-Hernandez, R. Martinez-Noriega, M. Nakano-Miyatake, and K. Yamaguchi, "Detection of BPCS-steganography using SMWCF steganalysis and SVM," in *Proc. Int. Symp. Inf. Theory Appl.*, Dec. 2008, pp. 1–5, doi: 10.1109/ISITA.2008.4895497.

[14] M. Płachta, M. Krzemień, K. Szczypiorski, and A. Janicki, "Detection of image steganography using deep learning and ensemble classifiers," *Electronics*, vol. 11, no. 10, p. 1565, May 2022, doi: 10.3390/electronics11101565.

[15] B. Bashir and A. Selwal, "Towards deep learning-based image steganalysis: Practices and open research issues," in *Proc. Int. Conf. IoT Based Control Netw. Intell. Syst. (ICICNIS)*, Jul. 2021, pp. 1–9, doi: 10.2139/ssrn.3883330.

[16] S. Huang, M. Zhang, Y. Ke, X. Bi, and Y. Kong, "Image steganalysis based on attention augmented convolution," *Multimedia Tools Appl.*, vol. 81, no. 14, pp. 19471–19490, Jun. 2022, doi: 10.1007/s11042-021-11862-4.

[17] X. Han and T. Zhang, "Spatial steganalysis based on non-local block and multi-channel convolutional networks," *IEEE Access*, vol. 10, pp. 87241–87253, 2022, doi: 10.1109/ACCESS.2022.3199351.

[18] F. M. McNeill and E. Thro, "THE fuzzy world," in *Fuzzy Logic*. Amsterdam, The Netherlands: Elsevier, 1994, pp. 1–22, doi: 10.1016/B978-0-12-485965-4.50007-9.

[19] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, pp. 111–127, Sep. 1997, doi: 10.1016/S0165-0114(97)00077-8.

[20] W. J. M. Kickert and E. H. Mamdani, "Analysis of a fuzzy logic controller," *Fuzzy Sets Syst.*, vol. 1, no. 1, pp. 29–44, Jan. 1978, doi: 10.1016/0165-0114(78)90030-1.

[21] Q. Liu, T. Qiao, M. Xu, and N. Zheng, "Fuzzy localization of steganographic flipped bits via modification map," *IEEE Access*, vol. 7, pp. 74157–74167, 2019, doi: 10.1109/ACCESS.2019.2920304.

[22] G. E. Martínez, C. I. Gonzalez, O. Mendoza, and P. Melin, "General type-2 fuzzy sugeno integral for edge detection," *J. Imag.*, vol. 5, no. 8, p. 71, Aug. 2019, doi: 10.3390/jimaging5080071.

[23] C. Venugopal, S. P. Devi, and K. S. Rao, "Predicting ERP user satisfaction—An adaptive neuro fuzzy inference system (ANFIS) approach," *Intell. Inf. Manage.*, vol. 2, no. 7, pp. 422–430, 2010, doi: 10.4236/iim.2010.27052.

[24] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1138–1150, 2020, doi: 10.1109/TIFS.2019.2936913.

[25] G. Salloum and J. Tekli, "Automated and personalized nutrition health assessment, recommendation, and progress evaluation using fuzzy reasoning," *Int. J. Hum.-Comput. Stud.*, vol. 151, Jul. 2021, Art. no. 102610, doi: 10.1016/j.ijhcs.2021.102610.

[26] J. D. L. C. Ntivuguruzwa, "Fuzzy inference-based prediction model for an IoT-based water and pasture localization for pastoralists," *Int. J. Res. Eng. Appl. Sci.*, vol. 11, no. 1, Feb. 2021.

[27] N. J. D. L. Croix, M. Didacienne, S. Louis, J. T. Philander, and T. Ahmad, "Internet of Things based controlled environment for the production of shiitake mushroom," in *Proc. IEEE Int. Conf. Blockchain Distrib. Syst. Secur. (ICBDS)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Sep. 2022, pp. 1–6, doi: 10.1109/ICBDS53701.2022.9936039.

[28] N. J. D. L. Croix, M. Didacienne, and S. Louis, "Fuzzy logic-based shiitake mushroom farm control for harvest enhancement," in *Proc. 10th Int. Symp. Digit. Forensics Secur. (ISDFS)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Jun. 2022, pp. 1–6, doi: 10.1109/ISDFS55398.2022.9800832.

[29] N. J. D. L. Croix, C. C. Islamy, and T. Ahmad, "Secret message protection using fuzzy logic and difference expansion in digital images," in *Proc. IEEE Nigeria 4th Int. Conf. Disruptive Technol. Sustain. Develop. (NIGERCON)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Apr. 2022, pp. 1–5, doi: 10.1109/NIGERCON54645.2022.9803151.

[30] I. Théophile, N. J. D. L. Croix, and T. Ahmad, "Fuzzy logic-based steganographic scheme for high payload capacity with high imperceptibility," in *Proc. 11th Int. Symp. Digit. Forensics Secur. (ISDFS)*, May 2023, pp. 1–6, doi: 10.1109/ISDFS58141.2023.10131727.

[31] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985, doi: 10.1109/TSMC.1985.6313399.

[32] N. Yadav, V. Singh, A. Rani, and S. Goyal, "An improved hyper smoothing function-based edge detection algorithm for noisy images," *J. Intell. Fuzzy Syst.*, vol. 38, no. 5, pp. 6325–6335, May 2020, doi: 10.3233/JIFS-179713.

[33] A. Rosenfeld, "Image analysis: Problems, progress and prospects," *Pattern Recognit.*, vol. 17, no. 1, pp. 3–12, Jan. 1984, doi: 10.1016/0031-3203(84)90031-1.

[34] J. H. Pujar and D. S. Shambhavi, "A novel digital algorithm for Sobel edge detection," in *Proc. Int. Conf. Bus. Admin. Inf. Process.*, 2010, pp. 91–95, doi: 10.1007/978-3-642-12214-9_16.

[35] R. A. Kirsch, "Computer determination of the constituent structure of biological images," *Comput. Biomed. Res.*, vol. 4, no. 3, pp. 315–328, Jun. 1971, doi: 10.1016/0010-4809(71)90034-6.

[36] C. Torres, C. I. Gonzalez, and G. E. Martinez, "Fuzzy edge-detection as a preprocessing layer in deep neural networks for guitar classification," *Sensors*, vol. 22, no. 15, p. 5892, Aug. 2022, doi: 10.3390/s22155892.

[37] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Dec. 2014, pp. 1–4, doi: 10.1109/APSIPA.2014.7041565.

[38] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012, doi: 10.1109/TIFS.2012.2190402.

[39] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010, doi: 10.1109/TIFS.2010.2045842.

[40] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch," Nov. 2015, *arXiv:1511.04855*.

[41] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016, doi: 10.1109/LSP.2016.2548421.

[42] G. Xu, H.-Z. Wu, and Y. Q. Shi, "Ensemble of CNNs for steganalysis: An empirical study," in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur.* New York, NY, USA: Association for Computing Machinery, Jun. 2016, pp. 103–107, doi: 10.1145/2909827.2930798.

[43] Y. Qian, J. Dong, W. Wang, and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. Washington, DC, USA: IEEE Computer Society, Sep. 2016, pp. 2752–2756, doi: 10.1109/ICIP.2016.7532860.

[44] J. Zeng, S. Tan, B. Li, and J. Huang, "Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis," in *Proc. IS T Int. Symp. Electron. Imag. Sci. Technol.* Springfield, VA, USA: Society for Imaging Science and Technology, 2017, pp. 44–49, doi: 10.2352/ISSN.2470-1173.2017.7.MWSF-324.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet Classification With Deep Convolutional Neural Networks*. Accessed: Sep. 11, 2023. [Online]. Available: http://code.google.com/p/cuda-convnet/

[46] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale JPEG image steganalysis using hybrid deep-learning framework," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1200–1214, May 2018, doi: 10.1109/TIFS.2017.2779446.

[47] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019, doi: 10.1109/TIFS.2018.2871749.

[48] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 10437–10453, May 2018, doi: 10.1007/s11042-017-4440-4.

[49] M. Yedroudj, F. Comby, and M. Chaumont, "Yedrouj-Net: An efficient CNN for spatial steganalysis," Feb. 2018, *arXiv:1803.00407*.

[50] C. F. Tsang and J. Fridrich, "Steganalyzing images of arbitrary size with CNNs," *Electron. Imag.*, vol. 30, no. 7, pp. 121-1–121-8, Jan. 2018, doi: 10.2352/issn.2470-1173.2018.07.mwsf-121.

[51] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Efficient feature learning and multi-size image steganalysis based on CNN," Jul. 2018, *arXiv:1807.11428*.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," 2014, *arXiv:1406.4729*.

[53] D. Hu, S. Zhou, Q. Shen, S. Zheng, Z. Zhao, and Y. Fan, "Digital image steganalysis based on visual attention and deep reinforcement learning," *IEEE Access*, vol. 7, pp. 25924–25935, 2019, doi: 10.1109/ACCESS.2019.2900076.

[54] S. Arivazhagan, E. Amrutha, and W. S. L. Jebarani, "Universal steganalysis of spatial content-independent and content-adaptive steganographic algorithms using normalized feature derived from empirical mode decomposed components," *Signal Process., Image Commun.*, vol. 101, Feb. 2022, Art. no. 116567, doi: 10.1016/j.image.2021.116567.

[55] T. Fu, L. Chen, Z. Fu, K. Yu, and Y. Wang, "CCNet: CNN model with channel attention and convolutional pooling mechanism for spatial image steganalysis," *J. Vis. Commun. Image Represent.*, vol. 88, Oct. 2022, Art. no. 103633, doi: 10.1016/j.jvcir.2022.103633.

[56] T.-S. Reinel, A. H. Brayan, B. M. Alejandro, M.-R. Alejandro, A.-G. Daniel, A. J. Alejandro, B. A. Buenaventura, O.-A. Simon, I. Gustavo, and R.-P. Raúl, ''GBRAS-Net: A convolutional neural network architecture for spatial image steganalysis,'' *IEEE Access*, vol. 9, pp. 14340–14350, 2021, doi: 10.1109/ACCESS.2021.3052494.

[57] F. Yalcinkaya and A. Erbas, ''Convolutional neural network and fuzzy logic-based hybrid melanoma diagnosis system,'' *Elektronika Elektrotechnika*, vol. 27, no. 2, pp. 55–63, Apr. 2021, doi: 10.5755/j02.eie.28843.

[58] P. Bas, T. Filler, and T. Pevný, '''Break our steganographic system': The ins and outs of organizing boss,'' in *Proc. Int. Workshop Inf. Hiding*, 2011, pp. 59–70, doi: 10.1007/978-3-642-24178-9_5.

[59] J. Ye, J. Ni, and Y. Yi, ''Deep learning hierarchical representations for image steganalysis,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017, doi: 10.1109/TIFS.2017.2710946.

NTIVUGURUZWA JEAN DE LA CROIX (Member, IEEE) received the B.Sc. degree in computer science and systems from the National University of Rwanda, Rwanda, the master's degree in information technology from the University of Madras, India, the P.G.Dip. degree in education from the University of Kigali, Rwanda, the master's degree in the Internet of Things and embedded computing systems from the University of Rwanda. He is currently pursuing the Ph.D. degree in computer science with Institut Teknologi Sepuluh Nopember (ITS), Indonesia. His current research interests include steganography, steganalysis, and deep learning for data security in public networks. He is a Reviewer of several journals, including IEEE Access.

TOHARI AHMAD (Member, IEEE) received the B.Sc. degree in computer science from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, the master's degree in information technology from Monash University, Australia, and the Ph.D. degree in computer science from RMIT University, Australia. He was a consultant for some international companies. In 2003, he moved to ITS, where he is currently a Professor. His current research interests include network security, information security, data hiding, and computer networks. His awards and honors include the Hitachi Research Fellowship and JICA Research Program to conduct research in Japan. He is a reviewer of several journals.

FENGLING HAN (Senior Member, IEEE) received the bachelor's degree in control theory and application from the Harbin Ship-Building Engineering Institute of Technology, China, the master's degree in automatic control engineering from the Harbin Institute of Technology, China, and the Ph.D. degree in computer and electronic engineering from RMIT University, Australia. Her current research interests include complex networks, industrial electronics, and cyber security. She has been involved in and leading research projects awarded by the Australia Research Council and the Victoria Government. She is an associate editor and a reviewer for top IEEE journals.