

## Detection of colorization based image forgeries using convolutional autoencoder method

Soumyashree Muralidhar Panchal<sup>1</sup>, Asha Kethaganahalli Hanumanthiah<sup>2</sup>,  
Bindushree Channabasavaraju Doddasiddavanahalli<sup>3</sup>, Manju More Eshwar Rao<sup>4</sup>,  
Ambika Belekere Jayaramu<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru,  
Manipal Academy of Higher Education, Manipal, India

<sup>2</sup>Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru, India

<sup>3</sup>Department of Computer Science and Engineering, REVA University, Bengaluru, India

<sup>4</sup>Department of Computer Science and Engineering, PES University, Bengaluru, India

### Article Info

#### Article history:

Received Mar 15, 2024

Revised Jul 26, 2024

Accepted Jul 29, 2024

#### Keywords:

Blind forgery detection  
Convolutional autoencoder  
Deep learning  
Digital images  
Image forgery

### ABSTRACT

Recently, it has become difficult to recognize and easier to misuse digital images due to the large number of editing tools available. Detecting forgeries in images is crucial for security and forensic purposes. Therefore, this research implements a deep learning (DL) method of convolutional autoencoder (CAE) which improves colorization-based image forgery detection by leveraging spatial and color information, increasing the detection accuracy. At first, the pre-processed input forgery images are used with the wiener filtering-contrast restricted improved histogram equalization (WE-CLAHE) technique. Hybrid dual-tree complex wavelet trigonometric transform (H-DTCWT) and VGG-16 are used to extract effective features from the clustered data. Improved horse herd optimization (IHH) is employed to reduce the dimensionality of a feature. At last, the CAE model is implemented to significantly recognize the image forgery. The accuracy of CASIA V1 and GRIP datasets of 99.95% and 99.97%, respectively is achieved. Hence, this implemented method obtains a high forgery detection performance than the existing methods.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



### Corresponding Author:

Ambika Belekere Jayaramu

Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru

Manipal Academy of Higher Education

Manipal, 576104, Karnataka, India

Email: ambika.bj@manipal.edu

## 1. INTRODUCTION

Copy-move forgery (CMF) is a prevalent technique employed in image manipulation, aimed at deceiving viewers. Its primary objective is to either draw attention to certain elements or conceal specific information within an image [1]. As the repeated sections are taken from the same image, they share many attributes such as color patterns, noise levels and dynamic ranges. Because these repeating portions blend-in with the rest of the image, it becomes challenging for humans to differentiate between the duplicate parts and the original image [2], [3]. Splicing, retouching and copy-move are the most popular methods used in image forgeries. A section of an image is pasted and copied to a new position in the CMF [4]. This approach repeats some elements that are clear from the image by using noise, color, contrast or other factors. As a consequence, it becomes increasingly difficult to detect other types of forgeries such as splicing and retouching [5], [6]. It is hence necessary to apply certain processing techniques like rotation, scaling,

downsampling, JPEG compression and noise addition to increase trust in copy-move altered images. This work focuses on copy-move forgery detection (CMFD) methods as image CMF detection is a challenging issue [7], [8]. Since JPEG format is used by a majority of digital cameras and image processing software to encode digital images, it is crucial to detect the compressing history of JPEG images. Additionally, the compression history of an image reveals potential changes in a particular JPEG image [9], [10]. Copy-paste and splicing/image composite are the two forgeries that typically affect digital imagery. To identify copy-paste forgeries, existing academics provide many techniques that all follow an easy procedure of feature extraction and feature vector comparison. However, image splicing provides some inherent alliterations to the images that increase variations already present.

Any forgery detection method's main objective is to identify these inconsistencies in the images [11]. It is easy to produce fake images without leaving a trace with help of an image editing software, which is available for free in the market. These elements have increased the trust issue regarding the reliability of digital images in recent years. Therefore, there is a constantly increasing need for efficient digital image forgery detection methods [12], [13]. The process of producing CMTIF includes pasting and copying a section of the image and then post-processing the resultant image [14]. The essential properties including illumination behavior, noise incurred, and color state are, are essentially maintained in both the source and tampered images [15]. Due to a lack of digital media, malicious image-raising has serious negative effects on military, politics, academics and the real world. Thus, the need for an effective method of identifying image tampering and fraud is crucial [16], [17]. However, there is a major difference in low-level semantic notions, because most CMF detection systems primarily rely on scale-invariant feature transform (SIFT), a low-level visual representation of digital images [18]. Modern techniques like convolutional neural network (CNN), mobile net and ResNet50v2 are trained on large datasets, and automatically extract the possible features according to the development of deep learning (DL) [19]. Deep features are used for person identification, skin lesion classification and image quality assessment, or a few instances of CNN-based feature extractions [20].

Walia *et al.* [21] implemented scale and direction local binary pattern (SD-LBP), utilized for digital image manipulation detection. The implemented method was employed to learn deep and complex features from pre-processed images for classification into authentic and forged images. IMD 2020, CASIA v1, DVMM, and CASIA v2 were four datasets employed to evaluate this method that achieved high detection accuracy using ResNet50. Due to less image forgery localization, this method required an extension in localized forgery through providing input to the neural network. Srivastava and Yadav [22] implemented multiple local binary patterns (LBP) utilized for the detection of both splicing and CMF. An RGB image was converted into a Cb and Cr image, while YCbCr image components were extracted in this implemented multiple LBP method. CASIA v2.0, Columbia and CASIA v1.0 were three datasets utilized for effectively evaluating tampering detection, employed with SVM to provide good results in both small and large datasets. Due to their impact on localizing forged parts in spliced images, this method needed to combine DL and ML methods, in order to efficiently localize forged parts in spliced images.

Rao *et al.* [23] implemented a knowledge-based fuzzy approximation (KBFA) and hybrid grey wolf ant lion optimization (H-GWAL) (KBFA with H-GWAL) approach. H-GWAL was implemented to detect spliced images, while localized tampered region spliced images were utilized by the KBFA model. The KBFA algorithm was used to effectively detect and categorize images, whereas the detection accuracy performance was improved by the H-GWAL method. However, the implemented model only localized spliced images and did not properly classify the images. Sushir *et al.* [24] implemented a hybrid deep convolutional capsule autoencoder (hybrid DCCAE), which was utilized for significant imager forgery recognition. A wiener filtering-contrast limited increased histogram equalization was used to effectively pre-process the image. The implemented method attained high performance of the system, because of the efficient reliability and training model capability, thereby effectively detecting blind image forgeries. Nonetheless, the implemented hybrid DCCAE method had lower forgery detection and image recognition performance. Diwan *et al.* [25] implemented the Superpoint approach for detecting CMF in digital images, used as a key point detector in self-supervised images. GRIP, CMFD, MICC-F220, MICC-F2000, MIC-F220, CoMoFoD, CASIA V2.0, and COVERAGE were the eight datasets utilized to detect copy-move image forgery. The implemented method produced high results in images with different attacks, and also effectively detected CMF in a diverse range of forged images. Due to less effectiveness in forgery detection, this Superpoint approach faced computational complexity, along with challenges in real-time processing or large-scale applications.

Bibi *et al.* [26] presented multiple structures stacked autoencoders (SAE) method, which was based on CNN utilized for forgery detection. The SAE based CNN method was employed with multiple structures which contains two and three stacked autoencoders for decreasing the feature dimensions. Two CASIA datasets were utilized to evaluate the presented method, which provided high forgery detection accuracy on JPEG image. Nonetheless, the presented method had overfitting issue, due to the complex architecture of SAE. Alhaidery *et al.* [27] implemented region growing-merging segmentation (RGMS) approach, which

was utilized to detect CMF in digital images. The implemented RGMS utilized among block-based and keypoint-based methods effectively localized the duplicated regions. It additionally improved the detection and region of interest assisted in enhancing localization for small and large regions. Yet, this method possessed low capability to recognize the difference between the authentic and forged regions. The main contributions of this work are given as follows:

- A DL-based feature extraction approach is used to accurately detect forgery images, alongside reducing redundant data and providing valuable features, even in areas with lower contrast and huge-sized images along VGGNet and DTT.
- This study presents a technique for pre-processing images using a WE-CLAHE which enhances contrast and noise, and improves system efficiency through feature extraction.
- Convolutional autoencoder (CAE) is a DL-based classification framework that accurately detects forgery, while enhancing classification accuracy due to its enhanced capability.

The rest of the paper is structured follows: section 2 describes the proposed method. Section 3 explains the process of CAE-CNN, while section 4 shows the results and discussion, and conclusion of this work is given in section 5.

## 2. PROPOSED METHOD

In this work, a CAE model is implemented for colorization-based image forgery detection. This work includes CASIA V1 and GRIP datasets for collecting data, and WE-CLAHE used in the pre-processing phase to effectively eliminate noise from the input image. H-(DTCWT-DCT) and VGG-16 methods are used to extract the optimal features and then, IHH is employed to reduce the features' dimensionality. The implemented CAE model is utilized to significantly recognize image forgery. Figure 1 denotes the implemented method's block diagram.

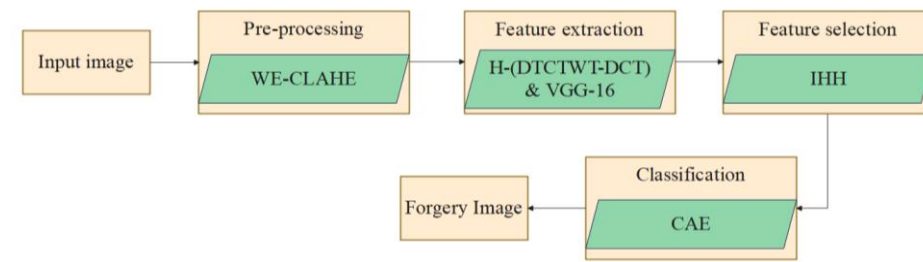


Figure 1. Implemented model's block diagram

### 2.1. Datasets

The work utilizes two datasets including CASIA V1 and GRIP to evaluate the implemented method. CASIA V1 [28] and GRIP [29] datasets are employed to collect the information. These two datasets are used to detect colorization-based image forgeries.

#### 2.1.1. CASIA V1 dataset

This CASIA V1 dataset consists of both fake and pristine images, where there are 921 tampered images and 800 authentic images. This dataset has 1,721 JPEG compressed images in total, of the size of 384×256. Thus, the 921 tampered images are divided into 469 spliced and 452 copy-move kind of tampered images. Adobe Photoshop software is used to make splice images, which are sections of one image that are connected to others. Figure 2 shows the sample images of the CASIA V1 dataset.



Figure 2. Sample images of the CASIA V1 dataset

**2.1.2. GRIP dataset**

This GRIP dataset was created by Cozzolino, containing 80 input images and 80 plain forgery images, which are accessible in 1024×768 in the format of PNG. In the format of PNG, the tampered portions are separately deposited. The tampered portion’s location is stored in the format of text in the forgery images. This dataset is divided into 3 parts with 50 real images without forgery images, 50 tampered images after translation, and scaling 20 sets of an images after rotation. Figure 3 represents the sample images of the GRIP dataset.



Figure 3. Sample images of GRIP dataset

**2.2. Pre-processing**

Following the collection of data, pre-processing is performed using the wiener filter-CLAHE. The income image size of both datasets contains 768×1024×3 for the GRIP dataset and 256×384×3 for the CASIA V1 dataset. An important aim of pre-processing image is to increase the quality of images and provide superior evaluations. To improve the accuracy of forgery detection, the unwanted distortions are compressed and the features are increased in the image. At first, to eliminate noise pre-processing is performed, and the image contrast is increased. This work utilizes wiener filter-CLAHE [30] to perform pre-processing. Wiener filter works on statistical models has a low pass filter which ensures that the signals and noise are stationary. Neighbourhood means and variance are calculated, and strong smoothing is used when the variation is at its minimum, while the minimal smoothing is applied when variation is at its maximum. This filter decreases the error between the original signal and estimated signal. If it is assumed that  $M'$  is the estimated image, and  $M$  is the original image, the error is measured using the (1), where the unexpected term is  $e$ , then (2) shows the minimum error function.

$$E^2 = e\{(M - M')^2\} \tag{1}$$

$$M'(f_1, f_2) = \left[ \frac{B^*(f_1, f_2)S'(f_1, f_2)}{|S'(f_1, f_2)|^2 B(f_1, f_2) + S'(f_1, f_2)} \right] C(f_1, f_2) \tag{2}$$

Where, a degraded image transform is  $C(f_1, f_2)$ , an estimated image in a frequency is  $M'(f_1, f_2)$ , and a degraded function transform is  $B(f_1, f_2)$ . Then, a non-grading image power spectral value is  $S'(f_1, f_2)$ , and  $B(f_1, f_2)$  conjugate term is  $B^*(f_1, f_2)$ . Pre-processed output images are passed as input to the feature extraction process.

**2.3. Feature extraction**

Following the image pre-processing, VGG-16 and H-(DTCWT–DCT) are utilized in this process of feature extraction. This process of feature extraction is performed to decrease dimensionality where a large pixel’s number is effectively expressed, so that the important parts of the image are collected. Currently, discrete wavelet transform is increased with DTCWT for significant properties. The CNN’s most recent vision, VVG-16 has a high learning capacity and is simple to assemble. From VGG 16, the most important features are extracted with deep neural networks processing the 16 layers. The abstraction of important features from VGG-16 and DTWCT is included in this feature extraction process. Low contrast areas are enhanced while the quality of the image is improved using the pre-processing technique. The important features are then effectively extracted from low contrast regions using the effective feature extraction techniques VGG-16 and H-(DTCWT–DCT), and the performance is enhanced by decreasing the feature dimensionality.

**2.3.1. H-(DTCWT–DCT)**

Directional selectivity in 2 or more dimensions and shift variance issues are addressed using DTCWT [31]. With the assistance of analytic wavelets, DTCWT achieves directional selectivity. Six directional subbands are generated by this model exposed in the directions including +12, +45, +75 real and

imaginary parts. Let filters be denoted as  $h_j(m)$ , and  $g_j(m)$ . The first  $F_{new}^{(l)}(e^{kw})$  and second  $S_{new}^{(l)}(e^{kw})$  are  $j^{th}$  phase filter bank response expressed in (3). To decompose the image  $g(a, b)$  using DTCWT for the translation's series, complex scaling dilations and functions of 6 CWT are provided in (4).

$$F_{new}^{(l)}(e^{kw}) = F\{F_{new}^{(l)}(e^{kw})\} \tag{3}$$

$$g(a, b) = \sum_{m \in \mathbb{Z}^2} a_{j\theta}, m^{\theta j\theta}, m^{(a,b)} + \sum_{m \in \mathbb{Z}^2} \sum_{k \geq k_0} \sum_{m \in \mathbb{Z}^2} a_j^\theta, m^{\theta j^\theta}, m^{(a,b)} \tag{4}$$

Where, the scaling term is denoted as  $m^{\theta j\theta}, m^{(a,b)}$ , and 6 wavelet terms are represented as  $m^{\theta j^\theta}, m^{(a,b)}$ , while the wavelets and scaling coefficients are shown as  $a_j^\theta$  and  $a_{j\theta}$ . At last,  $g(a, b)$  image is separated into  $s$  sub-bands of non-overlapping. Next, the highest and center frequency sub-bands are divided into blocks of  $4 \times 4$ . Next, the DCT transform function is appealed to each sub-band block, and the DCT transform is given in (5). At last, H-(DTCWT–DCT) coefficients are extracted from pre-processed images using H-(DTCWT–DCT).

$$g(c, d) = \sqrt{\frac{2}{M}} \sqrt{\frac{2}{N}} \alpha_c \alpha_d \sum_{c=0}^{M-1} \sum_{d=0}^{N-1} I(a, b) \cos \frac{(2a+1)c\pi}{2M} \cos \frac{(2a+1)d\pi}{2N} \tag{5}$$

**2.3.2. VGG-16**

The VGG-16's deep architecture enhances feature extraction for colorization-based image forgeries detection, while it is pre-trained on large datasets, hence increasing the model performance with minimal training data. The VGG-16 [32] has 16 layers, and it is an excellent vision network. Three 3 fully connected (FC) layers, 13 convolutional layers, and 16 learnable weights layers consist of 41 layers of  $3 \times 224 \times 224$  pixel VGG input. It has two types of filters:  $2 \times 2$  with max pooling layer's stride 2 and  $3 \times 3$  with convolutional layers' stride 1. The 64 and 128 filters are in the first and second convolutional layers. 256, 512, and 515 filters are in other convolutional layers. 1<sup>st</sup> and 2<sup>nd</sup> FC have 4,096 neurons, and at the end, 3 FC layers are produced. Last FC utilized for features is reduced to a thousand dimensions. This contains 138 million parameters approximately, and is a large network. Figure 4 illustrates the VGG-16 architecture.

H-(DTCWT–DCT) has a high directional selectivity feature that decomposes image texture from different directions and acquires descriptions of multiple feature information. Additionally, max-pooling and convolution are appealed for the first 4 blocks in VGG-16. This is because the output of the achieved features has a size of  $16 \times 16 \times 512$  and input image size is  $3 \times 224 \times 224$ . The extracted image features are passed as input to the feature selection phase.

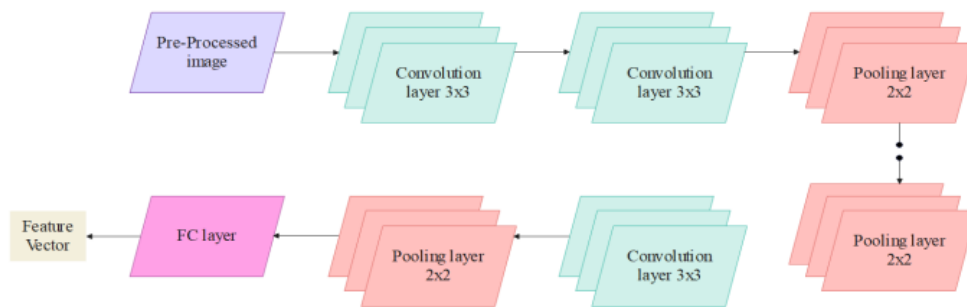


Figure 4. Architecture of VGG-16

**2.4. Feature selection**

Following the feature extraction, the feature dimensionality is decreased using the IHH. This IHH method increases the detection of colorization-based image forgery by effectively exploring solution space, resulting in faster convergence and high accuracy. One important procedure that ensures optimal feature collection to improve detection accuracy, speed up detection, lower error, and decrease feature dimensionality in feature selection. Metaheuristic algorithms extract important features from the feature sets. This optimization depends on the horse's characteristics, and it has phases that include Imitation (I), Grazing (G), Sociability (S), Defence (D) and Roam (R). Based on the following (6), the movement is provided to

horse at each iteration. Where a horse velocity vector is represented as  $\vec{W}_n^{iter,age}$ , the present iteration is denoted as  $iter$ , horse's age is showed as  $age$ , and  $n^{th}$  horse position is denoted as  $Y_n^{iter,age}$ .

$$Y_n^{iter,age} = \vec{W}_n^{iter,age} + Y_n^{(iter-1),age} \tag{6}$$

Grazing (G): the grazing field is developed by IHH optimization around each horse with  $h$  coefficient, and grazing model is expressed as (7) and (8). Where, the horse motion variable is denoted as  $\vec{G}_{r_n}^{iter,age}$ , and this expression is utilized for decreasing linearity with iterations of  $\sigma_g$ . Further, the upper and lower limit space in grazing is represented as  $UL$  and  $LL$ .

$$\vec{G}_{r_n}^{iter,age} = h_{iter}(LL + ULp) + Y_n^{(iter-1)} \tag{7}$$

$$h_n^{iter,age} = h_n^{(iter-1),age} \times \sigma_g \tag{8}$$

Hierarchy (H): horses observe hierarchy law in the middle age of  $\beta_i$  and  $\gamma_i$ , as proven by research. It is provided in (9) and (10). The horse's better location is denoted as  $Y_*^{(iter-1)}$ , and the effort of horse's better location is represented as  $\vec{N}_m^{iter,age}$ .

$$\vec{N}_m^{iter,age} = n_m^{(iter-1),age} [Y_*^{(iter-1)} - Y_n^{(iter-1)}] \tag{9}$$

$$n_m^{iter,age} = h_n^{(iter-1),age} \times \sigma_n \tag{10}$$

Sociability (S): it is defined as the number of observed horses aged between years 5 to 15 that express interest in a herd, as expressed in (11) and (12). Where, the horse orientation is denoted as  $P_m^{iter,age}$  and social vector motion is shown as  $\vec{P}_n^{iter,age}$ . With the  $\sigma_p$  factor,  $P_m^{iter,age}$  is reduced in each cycle.

$$\vec{P}_n^{iter,age} = P_m^{iter,age} \left[ \left( \frac{1}{M} \sum_{k=1}^M Y_k^{(iter-1)} \right) - Y_n^{(iter-1)} \right] \tag{11}$$

$$P_m^{iter,age} = P_m^{(iter-1),age} \times \sigma_p \tag{12}$$

Imitation (I): the horse characteristic is considered as  $j$  term in the algorithm. Other horses and the behavior of small horses do not convert their lives which is expressed as (13) and (14). Where, the decreased factor per cycle is denoted as  $\sigma_i$ , motion vector is denoted as  $\vec{Y}$ , and the position is shown as  $I_n^{iter,age}$ . The best position of the total horse is shown as  $Q_n$ .

$$\vec{I}_n^{iter,age} = I_m^{iter,age} \left[ \left( \frac{1}{Q_n} \sum_{k=1}^{Q_n} \vec{Y}_k^{(iter-1)} \right) - Y_n^{(iter-1)} \right] \tag{13}$$

$$I_n^{iter,age} = I_n^{(iter-1),age} \times \sigma_i \tag{14}$$

Defence (D): using the negative coefficients system of defense in IHH is determined in (15) and (16), utilized to move a horse beyond an unsuited place. Where, the horse with the worst position is represented as  $Q_n$ . The escape vector is denoted by  $\vec{D}_n^{iter,age}$ .

$$\vec{D}_n^{iter,age} = -d_n^{iter,age} \left[ \left( \frac{1}{Q_n} \sum_{k=1}^{Q_n} \vec{Y}_k^{(iter-1)} \right) - Y_n^{(iter-1)} \right] \tag{15}$$

$$d_n^{iter,age} = d_n^{(iter-1),age} \times \sigma_d \tag{16}$$

Roam (R): the horse is roaming and grazing to find food from one place to another. The roaming behavior is commonly viewed in young horses, and represented by (17) and (18). Where, the reduced term is shown as  $\sigma_r$ , and the local search velocity vector is represented by  $\vec{R}_n^{iter,age}$ . In IHH, (19) shows the horse velocity among ages 0 to 5 years.

$$\vec{R}_n^{iter,age} = r_n^{(iter,age)} pY^{(iter-1)} \tag{17}$$

$$r_m^{iter,age} = r_m^{(iter-1),age} \times \sigma_r \quad (18)$$

$$V_m^{iter} = \beta_i \times \left( g_m^{(iter-1),\delta_{wg}}(u + pl) [X_m^{(iter-1)}] \right) + (w - 1) \times \gamma_i \left[ l_m^{(iter-1),\delta_{wi}} \left[ \left( \frac{1}{p^N} \sum_{j=1}^p X_j^{(iter-1)} \right) - X^{(iter-1)} \right] + \left[ r_m^{(iter-1),\delta_{ww}} p X_m^{(iter-1)} \right] \right] \quad (19)$$

For selecting the meaningful features, the regulation parameters  $\beta_i$  and  $\gamma_i$  are used. Therefore, IHH is employed to select the optimal characteristics. Figure 5 illustrates the flowchart for IHH. The flow chart represents the feature selection method's detailed description. Following the parameter update, it satisfies evaluation criteria for the implemented performance; the procedure is repeated until achieving the optimal solution. The optimal features enhance the detection speed and performance of overall bling forgery detection. The output of selected features is passed as input to the classification process.

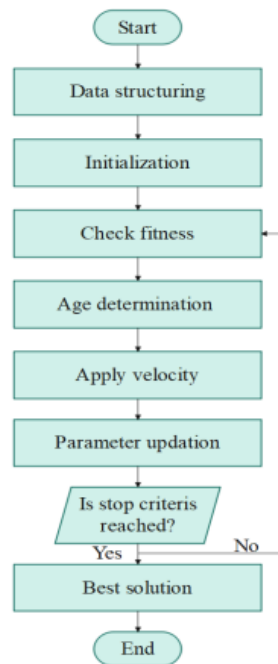


Figure 5. IHH optimization flowchart

### 3. CONVOLUTIONAL AUTOENCODER-CONVOLUTIONAL NEURAL NETWORK

After the feature selection, the implemented CAE-CNN (CAE-CNN) is utilized for the categorization process. By using convolutional layers to capture complex color patterns in images, CAE-CNN improves the accuracy of forgery detection and allows for strong differentiation between forgery and authentic images. At first, the training dataset is trained using CAE. The decoder component is discarded after the CAE completes its training procedure, whereas compressing first high-dimensional image dataset into a compressed dataset of image is carried out by using an encoder. At last, the CAE's encoder output (compressed dataset image) is utilized for training and feeding categorization of CNN including VGG, and ResNet.

In this consequence,  $C$  represents the model of CNN classification and  $l = \{l_1, l_2, \dots, l_N\}$ , where  $l_i \in \{0,1\} \forall i \in \mathbb{N}$ , as  $N$  refers total class output target in classification issue.  $x$  of initial training dataset is converted through  $E$  encoder into an encoded compressed 2D representation  $y$ .  $\hat{l} = \{\hat{l}_1, \hat{l}_2, \dots, \hat{l}_N\}$  is the raw output of CNN classification model, as given in (20). The reconstruction error in (21) is used to measure the CNN classification model's performance.

$$\hat{l} = C(y) \quad (20)$$

$$e_{CNN} = L_{CNN}(\hat{l}^{(k)}, l^{(k)}) \quad (21)$$



The difference in measurements like wide and cross-entropy loss functions are represented by the function  $L_{CNN}$ . Next, (22) shows the general form of the cost function. At last, optimal weight parameters are attained for the categorization model of CNN by reducing the  $J_{CNN}$  cost function concerning task classification.

$$J_{CNN} = \frac{1}{M} \sum_{k=1}^M L_{CAE} \left( D \left( E(l^{(k)}) \right), l(k) \right) \tag{22}$$

### 3.1. Implemented CAE’s architectural design

The implemented topology architecture of CAE is represented in Figure 6. At last, the CAE’s parameter settings configuration setup is denoted in Table 1. The symmetric architecture of the implemented CAE with 2D convolutional and deconvolutional layers of 4 batches comes after an activation function of rectified linear unit (ReLU). The deconvolutional (or transposed convolution) is performs the convolution layer’s reverse operation. It particularly converts information from a low-dimensional space to a high-dimensional one.

Most particularly, raw image dimensions  $H \times W \times 3$  are passed to 1<sup>st</sup> layer (2D conv1-ReLU1), which is also an income of CAE’s Encoder. The dimensions  $H/2 \times W/2$  develop 32 downsampled spatial feature maps, employing 32 filters of  $4 \times 4$  kernel size. Afterward, the outcome is passed into 2<sup>nd</sup> layer (2D Conv2-ReLU2) which is the output of Encoder with dimensions  $H/2 \times W/2 \times 3$  representation of compressed image, employing a dimension with 3 filters of  $2 \times 2$  kernel size. Using a smaller kernel size in 2<sup>nd</sup> layer makes it clear because feature maps from 1<sup>st</sup> layer’s output have smaller dimensional sizes than the input image. Likewise, third and fourth-layer CAE’s decoder components perform (2D Deconv4-ReLU4, 2D Deconv3-ReLU3) in an encoder’s reverse operation symmetric way.

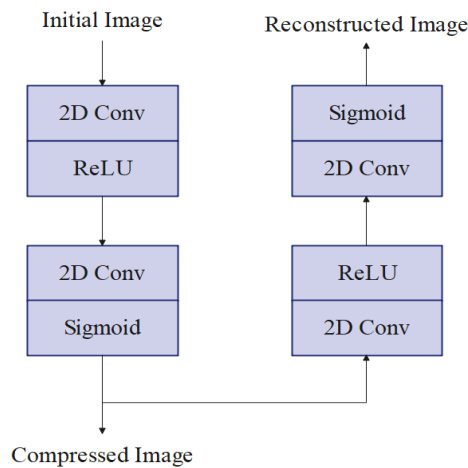


Figure 6. CAE topology’s architectural presentation

Table 1. CAE topology’s parameter settings

	Layers	Input Size	Kernel Size	Stride	Output Size
Encoder	2D Conv (E. Input)	$H \times W \times 3$	$4 \times 4 \times 32$	$2 \times 2 \times 1$	$H/2 \times W/2 \times 32$
	ReLU	$H/2 \times W/2 \times 32$	-	-	$H/2 \times W/2 \times 32$
	2D Conv	$H/2 \times W/2 \times 32$	$2 \times 2 \times 3$	$2 \times 2 \times 1$	$H/4 \times W/4 \times 3$
	Sigmoid (E. Output)	$H/4 \times W/4 \times 3$	-	-	$H/4 \times W/4 \times 3$
Decoder	2D Deconv3 (D. Input)	$H/4 \times W/4 \times 3$	$2 \times 2 \times 32$	$2 \times 2 \times 1$	$H/2 \times W/2 \times 32$
	ReLU	$H/2 \times W/2 \times 32$	-	-	$H/2 \times W/2 \times 32$
	2D Deconv4	$H/2 \times W/2 \times 32$	$4 \times 4 \times 3$	$2 \times 2 \times 1$	$H \times W \times 3$
	Sigmoid (D. Output)	$H \times W \times 3$	-	-	$H \times W \times 3$

Therefore, effective detection accuracy is achieved by using the approach provided, employing two different datasets. In comparison to the existing methods, the dimensionality failure and detection time are low. Hence, the implemented work is analyzed for the highly accurate use in blind image forgeries efficient detection. The following section provides a clear analysis of the performance of the implemented system.



## 4. RESULTS AND DISCUSSION

The implemented method is trained on two datasets, CASIA V1 and GRIP. Intel core i7 processor, Windows 10 (64-bit) operating system and 16 GB RAM are the system requirements for implementing the suggested method. This method's effectiveness is calculated in terms of specificity, recall, F1-score, precision, sensitivity and accuracy, explained below.

### 4.1. Evaluation parameters

The parameters like specificity, recall, F1-score, precision, sensitivity and accuracy are utilized to compute model performance. The parameters are expressed mathematically in the following (23), (24), (25), (26), (27) and (28). Where,  $FP$ ,  $FN$ ,  $TN$  and  $TP$  are denoted as false positive, false negative, true negative and true positive.

$$Precision = \frac{TP}{TP+FP} \quad (23)$$

$$Recall = \frac{TP}{TP+FN} \quad (24)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

$$F - score = \frac{Precision*Recall}{Precision+Recall} \quad (26)$$

$$Sensitivity = \frac{TP}{(TP+FN)} \quad (27)$$

$$Specificity = \frac{TN}{(FP+TN)} \quad (28)$$

### 4.2. Quantitative and qualitative analysis

This section provides an examination of the CAE method to determine the F1-score, specificity, recall, precision, sensitivity and accuracy. The CAE method's performance is evaluated using two datasets of CASIA V1 and GRIP. The effectiveness of the implemented model is represented in different tables below.

#### 4.2.1. Performance analysis using CASIA V1 dataset

Table 2 represents the presented feature extraction method's performance. The presented H-(DTCWT–DCT) and VGG-16 methods are compared with the current feature extraction methods including DTCWT, VGG-16, and complex wavelet transform (CWT) using the CASIA V1 dataset. The presented H-(DTCWT–DCT) and VGG-16 methods achieve the highest values of 90% precision, 91% recall, 93% accuracy, 85% F1-score, 92% sensitivity, and 89% specificity, in contrast to the existing feature extraction methods.

Table 3 denotes the presented feature selection method's performance. The presented IHH technique is compared with current feature selection models like the fruit fly optimization algorithm (FOA), artificial bee colony (ABC), and crow search algorithm (CSA) on the CASIA V1 dataset. The presented IHH method achieves the highest values of 92.21 % precision, 93.79% recall, 95.62% accuracy, 92.87% F1-score, 90.98% sensitivity and 94.26% specificity, when compared with other feature selection methods.

Table 2. Performance of feature extraction methods

Feature extraction methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
CWT	71	65	75	68	79	70
DTCWT	65	72	66	75	75	68
VGG-16	75	74	81	66	71	65
H-(DTCWT–DCT) and VGG-16	93	90	85	91	89	92

Table 3. Performance of feature selection methods

Feature selection methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
FOA	79.89	80.78	83.46	81.71	86.74	85.51
CSA	82.56	89.12	85.13	90.82	91.63	87.62
ABC	91.23	86.45	89.79	88.93	87.52	88.84
IHH	95.62	92.21	92.87	93.79	94.26	90.98

Table 4 shows the implemented model’s classification performance. The presented CAE model is compared with current classification models such as VGG-16, Inception and XceptionNet using the CASIA V1 dataset. The implemented CAE method attains the highest values of 99.53% precision, 99.62% recall, 99.95% accuracy, 99.82% F1-score, 99.70% sensitivity, and 99.89% specificity when compared to the other classification models.

Table 4. Performance of classification models

Classification methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
VGG-16	91.84	89.82	88.94	92.31	85.99	90.89
Inception	95.52	92.71	93.61	90.64	94.81	91.57
XceptionNet	93.28	90.93	94.37	88.97	95.78	95.26
CAE	99.95	99.53	99.82	99.62	99.89	99.70

**4.2.2. Performance analysis using GRIP dataset**

Table 5 represents the proposed method’s feature extraction performance. The presented H-(DTCWT–DCT) and VGG-16 methods are compared with current methods of feature extraction, including DTCWT, VGG-16 and CWT on the GRIP dataset. The presented H-(DTCWT–DCT) and VGG-16 methods attain superior values of 91% precision, 92% recall, 93% accuracy, 90% F1-score, 91% sensitivity and 92% specificity, in contrast to the existing methods of feature extraction.

Table 5. Performance of feature extraction methods

Feature extraction methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
CWT	89	70	88	79	85	87
DTCWT	90	89	75	84	89	79
VGG-16	78	85	86	90	76	81
H-(DTCWT–DCT) and VGG-16	93	91	90	92	92	91

Furthermore, Table 6 represents the proposed method’s feature selection performance. The presented IHH method is compared with current feature selection techniques like the fruit FOA, ABC, and CSA on the GRIP dataset. The presented IHH method achieves the highest values of 93.89% precision, 94.49% recall, 95.35% accuracy, 93.95% F1-score, 93.95% sensitivity, and 92.62% specificity, contrary to the existing feature selection methods.

Table 6. Performance of feature selection methods

Feature selection methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
FOA	87.36	88.35	82.93	85.96	79.98	90.45
CSA	92.25	82.59	85.81	86.85	86.89	83.97
ABC	89.14	90.78	89.50	91.74	81.72	85.84
IHH	95.35	93.89	92.75	94.49	92.62	93.95

Table 7 shows the implemented model’s classification performance. The presented CAE method is compared with current classification VGG-16, Inception, and XceptionNet models on the GRIP dataset. The implemented CAE method achieves the highest values of 99.59% precision, 99.70% recall, 99.97% accuracy, 99.48% F1-score, 99.62% sensitivity and 99.58% specificity, contrary to the existing classification models.

Table 7. Performance of classification models

Classification methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
VGG-16	85.90	80.97	91.68	89.98	95.17	87.84
Inception	93.84	92.50	92.18	90.65	93.28	89.95
XceptionNet	89.99	90.64	93.35	92.32	92.93	94.86
CAE	99.97	99.59	99.48	99.70	99.58	99.62

### 4.3. Comparative analysis

The implemented model's performance is analyzed by using parameters of precision, recall, F1-score, accuracy, specificity and sensitivity, as represented in this section. Table 8 denotes the accuracy of The existing and implemented methods on CASIA V1 and GRIP datasets. Table 9 displays the outcomes of the existing and implemented methods on both datasets. The suggested model outperforms other methods including SD-LBP [21], multiple LBP [22], KBFA with H-GWAL [23], hybrid DCCAE [24], Superpoint [25], SAE [26] and RGMS [27], as opposed to the pervious methods with the highest performance values.

Table 8. Comparative analysis of accuracy of existing and implemented methods

Datasets	Methods	Accuracy (%)
CASIA V1 dataset	SD-LBP [21]	99.31
	Multiple LBP [22]	98.2
	KBFA with H-GWAL [23]	99.56
	Hybrid DCCAE [24]	99.23
	SAE [26]	95.90
	CAE	99.95
GRIP dataset	Hybrid DCCAE [24]	98.07
	Superpoint [25]	N/A
	RGMS [27]	96.6
	CAE	99.97

Table 9. Comparative analysis of existing and implemented methods

Datasets	Methods	Precision (%)	F1-score (%)	Recall (%)	Specificity (%)	Sensitivity (%)
CASIA V1 dataset	Multiple LBP [22]	N/A	N/A	N/A	99.1	97.13
	KBFA with H-GWAL [23]	98.97	98.96	99.03	N/A	N/A
	CAE	99.53	99.82	99.62	99.89	99.70
GRIP dataset	Hybrid DCCAE [24]	98.07	98.75	98.75	N/A	N/A
	Superpoint [25]	N/A	96.93	N/A	N/A	N/A
	RGMS [27]	N/A	96.6	96.6	N/A	N/A
	CAE	99.59	99.48	99.70	99.58	99.62

### 4.4. Discussion

In this work, a CAE model is implemented for colorization-based image forgeries detection. The existing method's limitations and implemented approach's benefits are discussed in this section. Some limitations of the existing methods are that the SD-LBP [21] method has less forgery detected, hence needing to extend the detection of forgery by providing an income to neural network. Multiple LBP [22] method has an impact on localizing the forged part in spliced images, and so needs to combine other DL and ML methods to efficiently localize forged parts in spliced images. Hybrid DCCAE [24] method restricts in forgery detection performance due to the challenge in combining capsule networks with autoencoder, and also did not recognize the image accurately. Superpoint [25] approach had difficulties for large-scale applications. To overcome these issues, CAE model is implemented in this research. The implemented CAE combined with CNN effectively extracts the discriminative features from colorized images, enabling the forged regions detection based on the difference in colorization patterns. Through the abovementioned comparative analysis of Table 8, and as opposed to the existing methods like SD-LBP [21], multiple LBP [22], KBFA with H-GWAL [18], hybrid DCCAE [24], Superpoint [25], and SAE [26], the implemented CAE method achieves a superior accuracy of 99.95% on CASIS V1 dataset, while achieving 99.97% on the GRIP dataset. By using difference in colorization pattern, the implemented CAE method enables to effectively distinguish among regions that are forged and authentic, and also obtains commendable accuracy in forgery detection. As a further extension, more relevant databases can be included for retrieving the forgery images accurately.

## 5. CONCLUSION

In this paper, colorization-based image forgeries are detected using the implemented CAE model. The implemented CAEs are effectively utilized for colorization-based image forgery detection by deploying their ability to attain images' compact representations, while maintaining significant image features. At first, the income forgery images are pre-processed for efficient noise removal utilizing the WE-CLAHE method. The optimal features are extracted from the clustered data by employing H-DTCWT-DCT and VGG-16

methods. In addition, the feature dimensionality is decreased along IHH, so as to increase the accuracy of categorization. At last, the CAE model is developed for significant recognition of image forgery. This implemented model is trained on forgery and original images, and the performance is tested on two datasets that consist of ground truth, forgery and original images. The implemented CAE model is compared with existing methods including SD-LBP, multiple LBP, KBFA with H-GWAL, hybrid DCCAE, Superpoint, and SAE, based on metrics of precision, recall, F1-score, accuracy, specificity and sensitivity. When applied on CASIA V1 and GRIP datasets, the model obtains commendable accuracy values of 99.95% and 99.97%, respectively. Therefore, the implemented method proves to be most robust in forgery detection, as opposed to the existing methods. In the future, this work will use more relevant databases to retrieve the forgery images accurately.





## REFERENCES

- [1] S. Samir, E. Emary, K. El-Sayed, and H. Onsi, "Optimization of a pre-trained AlexNet model for detecting and localizing image forgeries," *Information (Switzerland)*, vol. 11, no. 5, p. 275, May 2020, doi: 10.3390/INFO11050275.
- [2] X. yang Wang, X. qi Wang, P. pan Niu, and H. ying Yang, "Accurate and robust image copy-move forgery detection using adaptive keypoints and FQGPCET-GLCM feature," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 2203–2235, May 2024, doi: 10.1007/s11042-023-15499-3.
- [3] V. Verma, D. Singh, and N. Khanna, "Block-level double JPEG compression detection for image forgery localization," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 9949–9971, Jun. 2024, doi: 10.1007/s11042-023-15942-5.
- [4] G. Tahaoglu, G. Ulutas, B. Ustubioglu, M. Ulutas, and V. V. Nabiyev, "Ciratefi based copy move forgery detection on digital images," *Multimedia Tools and Applications*, vol. 81, no. 16, pp. 22867–22902, Jan. 2022, doi: 10.1007/s11042-021-11503-w.
- [5] N. Kaur, N. Jindal, and K. Singh, "A deep learning framework for copy-move forgery detection in digital images," *Multimedia Tools and Applications*, vol. 82, no. 12, pp. 17741–17768, Oct. 2023, doi: 10.1007/s11042-022-14016-2.
- [6] J. S. Sujin and S. Sophia, "High-performance image forgery detection via adaptive SIFT feature extraction for low-contrast or small or smooth copy-move region images," *Soft Computing*, vol. 28, no. 1, pp. 437–445, Apr. 2024, doi: 10.1007/s00500-023-08209-6.
- [7] M. J. Kwon, S. H. Nam, I. J. Yu, H. K. Lee, and C. Kim, "Learning JPEG compression artifacts for image manipulation detection and localization," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1875–1895, May 2022, doi: 10.1007/s11263-022-01617-5.
- [8] S. Lu, X. Hu, C. Wang, L. Chen, S. Han, and Y. Han, "Copy-move image forgery detection based on evolving circular domains coverage," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 37847–37872, Apr. 2022, doi: 10.1007/s11042-022-12755-w.
- [9] L. Bertojo, C. Néraud, and W. Puech, "A very fast copy-move forgery detection method for 4K ultra HD images," *Frontiers in Signal Processing*, vol. 2, Jun. 2022, doi: 10.3389/frsip.2022.906304.
- [10] Y. Aydin, "Automated identification of copy-move forgery using Hessian and patch feature extraction techniques," *Journal of Forensic Sciences*, vol. 69, no. 1, pp. 131–138, Oct. 2024, doi: 10.1111/1556-4029.15415.
- [11] R. Hansda, R. Nayak, B. K. Balabantaray, and S. Samal, "Copy-move image forgery detection using phase adaptive spatio-structured SIFT algorithm," *SN Computer Science*, vol. 3, no. 1, Nov. 2022, doi: 10.1007/s42979-021-00903-2.
- [12] C. Wang, Z. Huang, S. Qi, Y. Yu, G. Shen, and Y. Zhang, "Shrinking the semantic gap: spatial pooling of local moment invariants for copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1064–1079, 2023, doi: 10.1109/TIFS.2023.3234861.
- [13] W. Li, C. Feng, L. Wei, and D. Wu, "Improving the generalization of face forgery detection via single domain augmentation," *Multimedia Tools and Applications*, vol. 83, no. 23, pp. 63975–63992, Jan. 2024, doi: 10.1007/s11042-023-17840-2.
- [14] Y. Z. Bahia, F. Meriem, and B. Messaoud, "Face spoofing detection using heterogeneous auto-similarities of characteristics," *Engineering Applications of Artificial Intelligence*, vol. 130, p. 107788, Apr. 2024, doi: 10.1016/j.engappai.2023.107788.
- [15] K. Sunitha, A. N. Krishna, and B. G. Prasad, "Copy-move tampering detection using keypoint based hybrid feature extraction and improved transformation model," *Applied Intelligence*, vol. 52, no. 13, pp. 15405–15416, 2022, doi: 10.1007/s10489-022-03207-x.
- [16] E. U. H. Qazi, T. Zia, and A. Almorjan, "Deep learning-based digital image forgery detection system," *Applied Sciences (Switzerland)*, vol. 12, no. 6, p. 2851, Mar. 2022, doi: 10.3390/app12062851.
- [17] A. K. Jaiswal and R. Srivastava, "Detection of copy-move forgery in digital image using multi-scale, multi-stage deep learning model," *Neural Processing Letters*, vol. 54, no. 1, pp. 75–100, Aug. 2022, doi: 10.1007/s11063-021-10620-9.
- [18] S. Walia, K. Kumar, and M. Kumar, "Unveiling digital image forgeries using Markov based quaternions in frequency domain and fusion of machine learning algorithms," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 4517–4532, Jul. 2023, doi: 10.1007/s11042-022-13610-8.
- [19] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong, and X. Cui, "DCU-Net: a dual-channel U-shaped network for image splicing forgery detection," *Neural Computing and Applications*, vol. 35, no. 7, pp. 5015–5031, Aug. 2023, doi: 10.1007/s00521-021-06329-4.
- [20] S. Agarwal and K. H. Jung, "Forensic analysis and detection using polycolor model binary pattern for colorized images," *Multimedia Tools and Applications*, vol. 83, no. 14, pp. 41683–41702, Oct. 2024, doi: 10.1007/s11042-023-16675-1.
- [21] S. Walia, K. Kumar, S. Agarwal, and H. Kim, "Using XAI for deep learning-based image manipulation detection with shapley additive explanation," *Symmetry*, vol. 14, no. 8, p. 1611, Aug. 2022, doi: 10.3390/sym14081611.
- [22] V. Srivastava and S. K. Yadav, "Digital image tampering detection using multilevel local binary pattern texture descriptor," *Journal of Applied Security Research*, vol. 17, no. 1, pp. 62–79, Apr. 2022, doi: 10.1080/19361610.2021.1883397.
- [23] A. Rao, C. S. Rao, and D. R. Cheruku, "Differentiating digital image forgeries and tampering localization by a novel hybrid approach," *Multimedia Tools and Applications*, vol. 81, no. 13, pp. 18693–18713, Mar. 2022, doi: 10.1007/s11042-022-12257-9.
- [24] R. D. Sushir, D. G. Wakde, and S. S. Bhutada, "Enhanced blind image forgery detection using an accurate deep learning based hybrid DCCAE and ADFC," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 1725–1752, 2024, doi: 10.1007/s11042-023-15475-x.
- [25] A. Diwan, D. Kumar, R. Mahadeva, H. C. S. Perera, and J. Alawatugoda, "Unveiling copy-move forgeries: enhancing detection with superpoint keypoint architecture," *IEEE Access*, vol. 11, pp. 86132–86148, 2023, doi: 10.1109/ACCESS.2023.3304728.
- [26] S. Bibi, A. Abbasi, I. U. Haq, S. W. Baik, and A. Ullah, "Digital image forgery detection using deep autoencoder and CNN features," *Human-centric Computing and Information Sciences*, pp. 1–17, 2021.
- [27] M. M. A. Alhaidery, A. H. Taherinia, and H. I. Shahadi, "A robust detection and localization technique for copy-move forgery in digital images," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 1, pp. 449–461, Jan. 2023, doi: 10.1016/j.jksuci.2022.12.014.

- [28] "CASIA Dataset," *Kaggle*, 2018. <https://www.kaggle.com/datasets/sophatvathana/casia-dataset>.
- [29] "GRIP dataset," [Online]. Available: [https://www.researchgate.net/figure/Copy-move-forgery-detection-results-on-the-GRIP-dataset-a-Original-image-b-Forged\\_fig4\\_367239091](https://www.researchgate.net/figure/Copy-move-forgery-detection-results-on-the-GRIP-dataset-a-Original-image-b-Forged_fig4_367239091).
- [30] S. Modak, J. Heil, and A. Stein, "Pansharpening low-altitude multispectral images of potato plants using a generative adversarial network," *Remote Sensing*, vol. 16, no. 5, p. 874, Mar. 2024, doi: 10.3390/rs16050874.
- [31] K. Rao, M. Bansal, and G. Kaur, "An effective CT medical image enhancement system based on DT-CWT and adaptable morphology," *Circuits, Systems, and Signal Processing*, vol. 42, no. 2, pp. 1034–1062, Sep. 2023, doi: 10.1007/s00034-022-02163-8.
- [32] G. Suganeshwari, R. Balakumar, K. Karuppanan, S. B. Prathiba, S. Anbalagan, and G. Raja, "DTBV: a deep transfer-based bone cancer diagnosis system using VGG16 feature extraction," *Diagnostics*, vol. 13, no. 4, p. 757, Feb. 2023, doi: 10.3390/diagnostics13040757.

## BIOGRAPHIES OF AUTHORS







**Soumyashree Muralidhar Panchal**     is currently working in Manipal University, Bangalore in the Department of Computer Science and Engineering. She has around 7 years of teaching Experience. She has obtained her Bachelor of Engineering (B.E), Master of Technology (M.Tech) and Doctor of Philosophy (Ph.D.) from Visvesvaraya Technological University (VTU), Belagavi. She has published various technical and research papers in national, international journals and conferences. Her research area is image processing using machine learning. She can be contacted at email: soumya.shree@manipal.edu.







**Asha Kethaganahalli Hanumanthaiah**     is currently working in Don Bosco Institute of Technology in the Department of Computer Science and Engineering. She has around 12 years of teaching experience from various institutes. She has obtained her Bachelor of Engineering (B.E), Master of Technology (M.Tech) and Doctor of Philosophy (Ph.D.) from Visvesvaraya Technological University (VTU), Belagavi. She has published various technical and research papers in national, international journals and conferences. Her areas of research include cloud computing, data mining, machine learning, and image processing. She can be contacted at email: asha.kh06@gmail.com.







**Bindushree Channabasavaraju Doddasiddavanahalli**     is currently working in REVA University, Bangalore in the Department of Computer Science and Engineering. She has around 12 years of teaching Experience from BNMIT, Jain University and REVA University, Bangalore. She has obtained her Bachelor of Engineering (B.E), Master of Technology (M.Tech) from Visvesvaraya Technological University (VTU), Belagavi and Doctor of Philosophy(Ph.D.) from REVA University. She has published various technical and research papers in national, international journals and conferences. Her areas of research include data mining, machine learning, and healthcare analytics. She can be contacted at email: binduojin@gmail.com.



**Manju More Eshwar Rao**     is currently working in PES University, Bangalore in the Department of Computer Science and Engineering. She has around 11+ years of teaching Experience from Alpha College of Engineering, REVA University and Gitam University, Bangalore. She has obtained her Bachelor of Engineering (B.E), Master of Technology (M.Tech) and Doctor of Philosophy (Ph.D.) from Visvesvaraya Technological University (VTU), Belagavi. She has published various technical and research papers in national, international journals and conferences. Her areas of research include cloud computing, data mining, and machine learning. She can be contacted at email: manjumore13@gmail.com.



**Ambika Belekere Jayaramu**     is currently working in Manipal Institute of Technology Bangaluru in the department of Computer Science and Engineering. She has around 14 years of teaching Experience from different universities. She has obtained her Bachelor of Engineering (B.E), Master of Technology (M.Tech) from Visvesvaraya Technological University (VTU), Belagavi and Doctor of Philosophy (Ph.D.) from REVA University. She has Published various technical and research papers in national, international journals and conferences. Her areas of research include networks, cloud computing, data mining, and machine learning. She can be contacted at email: ambika.bj@manipal.edu.