IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2021, accepted July 6, 2021, date of publication July 12, 2021, date of current version July 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3096240

Fusion of Handcrafted and Deep Features for Forgery Detection in Digital Images

SAVITA WALIA[1], KRISHAN KUMAR[1], MUNISH KUMAR[2], AND XIAO-ZHI GAO[3]
[1]University Institute of Engineering and Technology, Panjab University, Chandigarh 160014, India
[2]Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda 151001, India
[3]School of Computing, University of Eastern Finland, 70211 Kuopio, Finland

Corresponding author: Savita Walia (s.walia24@pu.ac.in)

ABSTRACT Content authentication of digital images has captured the attention of forensic experts and security researchers due to a multi-fold increase in the dissemination of multimedia data through the open and vulnerable Internet. Shrewd attackers successfully devise novel ways to challenge state-of-art forensic tools used for forgery detection in digital images. Feature engineering approaches have yielded up to 97% accuracy on benchmarked datasets. Deep learning approaches have shown promising results in various image classification problems but cannot find hidden patterns in digital images, which can reliably detect image forgeries. State-of-art accuracy of deep learning approaches for forgery detection is up to 98% on benchmarked datasets. The objective of the proposed approach is to further escalate the detection accuracy, pushing it near 100%. In this paper, a synergy of handcrafted features based on color characteristics and deep features using the image's luminance channel is employed to mine patterns responsible for accurate forgery detection. In the first Stream, 648-D Markov-based features are computed from the quaternion discrete cosine transform of the image. In the second Stream, the luminance channel of YCbCr colorspace is used to extract the Local Binary Pattern of the image. Further, local binary feature maps are fed to the pre-trained ResNet-18 model to obtain a 512-D feature vector called 'ResFeats' from the last layer of the model's convolutional base portion. The handcrafted features from Stream I and ResFeats from Stream II are combined to form an 1160-D feature vector. Further, classification is performed using a shallow neural network, and the method is tested on CASIA v1 and CASIA v2 datasets. The accuracy of the proposed fusion-based approach is 99.3% on benchmark datasets.

INDEX TERMS Image forensics, image manipulations, pre-trained networks, Markov process, local binary pattern, deep learning, machine learning.

I. INTRODUCTION

Due to the rise and inescapable utilization of web-based media platforms such as WhatsApp, Instagram, Facebook, YouTube, etc., there has been steep growth in the number of pictures being transferred and shared on these platforms. These digital pictures are utilized to spread data to an audience on a wide scale and consequently formulate a general opinion on a large scale. Due to the handiness of software and editing tools on the Internet, images are prone to fraudulent manipulations. Such images are disseminated on social media and even used in courtrooms, literature works, science and medicine, military, etc. Image forgery is used to refer to the act of manipulating images to showcase false information or to hide some helpful information from the images. The motive behind such manipulations can be various factors like earning money, disseminating rumors, or making false claims. Most regularly performed falsifications on digital pictures comprise splicing, retouching, and copy-move. In splicing [1], the counterfeit segment is taken from some other image, whereas in copy-move [2], the forged segment belongs to the same image. Retouching refers to the process of changing the appearance of the subject in the image. Retouching [3] is generally utilized in style photography to improve the people's skin in the picture to make their skin immaculate. Due to such manipulations, the intrinsic properties of the image, such as pixel correlations, chrominance, and luminance characteristics, become inconsistent. Various forgery detection methods [4] have been proposed in the literature to detect these types of inconsistencies. Previous studies are distributed

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

into two broad groups; Active and Passive approaches. Active methods [5] rely on prior information about the image under consideration. It includes a digital watermark or digital signatures embedded onto the image, and the watermark/signature is extracted at the receiver end to match with the original watermark/signature.

On the other hand, passive methods [6] are used when no prior information is available. The pixels of an image are used to determine the intrinsic changes in the images due to underlying manipulations. Some inherent characteristics must be extracted from images that are highly discriminable to distinguish forged and authentic images.

Researchers have experimented with various manually engineered features like keypoint-based [7]–[10], block-based features [11]–[14], pixel-based [15]–[17], etc., to detect these forgeries initially. Later, various deep learning methods are also investigated [18], [19], but results were not as per expectations for the forensics community because the available pre-trained architectures are specifically designed and trained for a different image classification task rather than forgery detection task. The concept of transfer learning [20] has recently been applied in various image processing applications using pre-trained deep neural networks. The trained layers of these deep neural networks are utilized to extract high-level features required to perform edge or corner detection in images. But these features are only capable of giving good performance for the problems these networks are trained for. In order to use these networks for other image processing tasks such as image forgery detection, deep features extracted from these networks needs to be combined with manually crafted features to enhance the accuracy. Further, to improve the detection accuracy, the combination of two streams of features has been proposed to form a more distinctive representation apt for content authentication of digital images. The method presented in the paper is novel in four different ways as follows:

1. This is the first approach for detecting image manipulations utilizing a blend of in-depth, high-level features and manually engineered image features. The detection accuracy has been improved by combining the deep high-level features and manually engineered image features compared to traditional state-of-the-art detection methods on CASIA v1 and CASIA v2 datasets.
2. Three channels of RGB color space are used in the case of handcrafted feature extraction, and the luma channel of YCbCr color space is used in the case of deep features to detect forgery in images. The complementary information in two different colorspaces has effectively characterized forgeries in digital images
3. We feed Scale and Orientation invariant local binary pattern maps of the image to the pre-trained ResNet-18 model instead of giving RGB images directly. The rich textural description capability of LBP helps to obtain a more meaningful and low-dimensional representation from the deep neural network.
4. A shallow neural network is trained using a fused feature vector for classification.

The remaining article is structured as follows. Section II explains the previous work in this area. Section III details handcrafted feature extraction and the method to extract deep features from the pre-trained neural network. In Section IV, the experimental setup and results are explained. Finally, concluding notes and findings of the experimental outcomes are given in Section V.

II. PREVIOUS WORK

As the proposed approach uses a blend of manually engineered features and deep high-level features so, it is necessary to go through the literature related to them. This section presents a brief review of existing approaches based on conventional methods and deep learning-based approaches in the domain of digital image forgery detection.

A. CONVENTIONAL METHODS

The Markov process [21] has been an absolute effective tool for detecting image manipulations in literature. A 2-Dimensional non-causal model has been proposed [22], representing the image as a 2-Dimensional signal and capturing the underlying dependencies between the current pixel and its neighbors. Cross-domain features are extracted using Blocks discrete cosine transform and discrete Meyer-wavelet transform. The approach's major setback is that the combined feature vector's dimensionality is very high (14240-D), and the approach reaches an accuracy of 93.36% on the splicing detection dataset. Multiple texture descriptors [23] are used to represent a single image. Various texture descriptors used are Local Phase Quantization, Local Binary Pattern, Binary Statistical Image Features, and Binary Gabor Pattern. The texture features are extracted from each sub-band after Steerable Pyramid Transform is applied. The Relief feature selection method is used to choose features from this enormous representation to produce a compact representation. For classification, Random Forest Classifier is used, which achieved 97% accuracy on the CASIA v2 dataset. The feature dimension before feature selection was 19680. This method's weakness is that the authors have chosen only a particular color channel from the YCbCr color space, which results in the loss of information.

Another method is proposed, where a combination of Local Binary Pattern (LBP), Histogram of Oriented Gradients, and Higher-Order Statistical Features are used and classified using an artificial neural network [24]. The complexity of the method is extremely high as LBP is calculated for every color channel, and still, the accuracy of the method is less than few state-of-art techniques. Again, Cb and Cr images from YCbCr color space are chosen to apply a multi-scale entropy filter [25]. After this, Local Phase Quantization is applied on these entropy filtered images. Classification is performed using an SVM classifier to achieve an accuracy of 95.41 on CASIA v1 and 98.33 on CASIA v2 datasets. Local binary patterns [26] are extracted from the Discrete

Wavelet Transform (DWT) domain of the image. The combinations of features from all four sub-bands of DWT are used for final representation. The method gives the best accuracy on the image's chrominance channel using a Support Vector Machine (SVM) classifier with 10-fold cross-validation. Markov feature extraction [27] for color images is proposed using threshold expansion and maximization. Again, Markov-based features [28] from two different domains are used to distinguish forged and authentic images. The method reached an accuracy of 93.55% on the CASIA v20 dataset.

B. DEEP LEARNING BASED METHODS

As of late, the quick expansion in deep learning-based techniques has turn out to be the most mainstream research point. Deep learning-based strategies have been generally utilized in various image processing applications such as classification [29], [30] identification [31]–[33], and segmentation [34]–[36] Deep-learning methods are better than conventional image handling methods, as they do not need the user to decide or calculate the image features beforehand. These methods can extricate features from images by self-learning through various convolutional and pooling layers in the network. The first CNN architecture, LeNet-5 [37], was proposed, and this network was applied to handwriting recognition in the MNIST dataset. The used images were in grayscale, and the size of the images was 32 × 32. The recognition accuracy of LeNet-5 is superior to that of conventional recognition systems. Then GPU was introduced in the CNN architecture through AlexNet [38]. Dropout [39] and Relu [40] were also added to the Deep-Neural Network (DNN) architecture to increase its accuracy. Liu and Deng *et al.* [41] proposed VGG-16, and it achieved a top-5 accuracy for testing of 92.5% in ImageNet. VGG networks are painfully slow to train, and their network architecture weights are substantial. Another architecture was proposed, i.e., GoogleNet [42], in which the Inceptionstructure was introduced into the network. The broadness of the network is expanded by utilizing different convolution kernel sizes to extract different features. 1 × 1 convolution layers are introduced to decrease the dimension, improving the accuracy when the network lessens the parameters. A residual network [43] is proposed that directly maps the low-level features to higher-level features by introducing the concept of identity connections or skip connections. The block that contains such connections is called the residual blocks. Residual Networks are analogous to networks with convolution, pooling, activation and fully-associated layers arranged over one another. The only distinction is that it has the identity connection between the layers. Through identity connections, the network straightforwardly tries to find out the output functions with no further support. ResNet Networks are faster than AlexNet and VGG-16 because of the lesser number of channels in the convolutional layers. The architecture of ResNet is made such that it is much denser as well as faster than other networks. ResNet-18 has roughly 11.7 million parameters whereas VGG-16 has 138 million parameters. These pre-trained architectures can also be used for extracting deep high-level features from different layers of the network.

A pre-trained AlexNet model is modified by adding few layers and trained over the CASIA v1 and CASIA v2 datasets [44]. These datasets are relatively very small to train such large models, whereas the authors have achieved an accuracy of 96.8% and 97.44% on CASIA v1 and CASIA v2 datasets, respectively. In another method [45], high pass filters in the Spatial Rich color model are used to feed to a 10 layer convolutional neural network. A complex set of features [46] derived from 3 Level 2-Dimensional Daubechies decomposition are obtained. Stacked AutoEncoders are used for learning these complex 450-dimensional feature vectors. The method managed to achieve an overall accuracy of 91.09%. Various research studies are available in the literature that used these pre-trained architectures for image forgery detection but lacked sufficient reasoning for obtained results.

Engineering of features in different domains can be a cumbersome task at times; therefore, due to the advent of deep learning methods, automatic learning of concerned features from the training images is observed to be an exemplary method to substitute the features carefully chosen by the user in many image processing applications. A new approach based on hybrid features that combine information from manually engineered and deep high-level features is proposed to benefit from automatic learning. The idea of fusion came from a recently published study [47] to apply face recognition systems in which authors have combined multi-level LBP features and 4096-Dimensional deep features from the Vggnet-19 model.

III. PROPOSED METHODOLOGY

In the proposed method, two streams are used to mine features and further combined to generate the most discriminable features. The handcrafted features and in-depth features can represent different kinds of information from input images, enhancing detection accuracy. Figure 1 shows the block diagram of two different streams used for feature representations. The details are provided:

A. STREAM I: HANDCRAFTED FEATURE ENGINEERING

Most of the Markov model-based approaches treat the image as a 1-D signal as per the literature. The conventional methods only portray the state conditions between adjacent states and specific directions (vertical, horizontal). In this approach, state dependencies along minor and major diagonals are also considered to represent the image better. The pseudocode for the handcrafted feature extraction is given in Algorithm 1. The image is segmented into blocks of size 8 × 8. The three color components are extracted and processed separately. Further, intra-block and inter-block differences are utilized to formulate the feature vector in vertical, horizontal, and diagonal directions on the quaternion discrete cosine transformation of an RGB image. The correlations between and within the blocks are considered along with significant

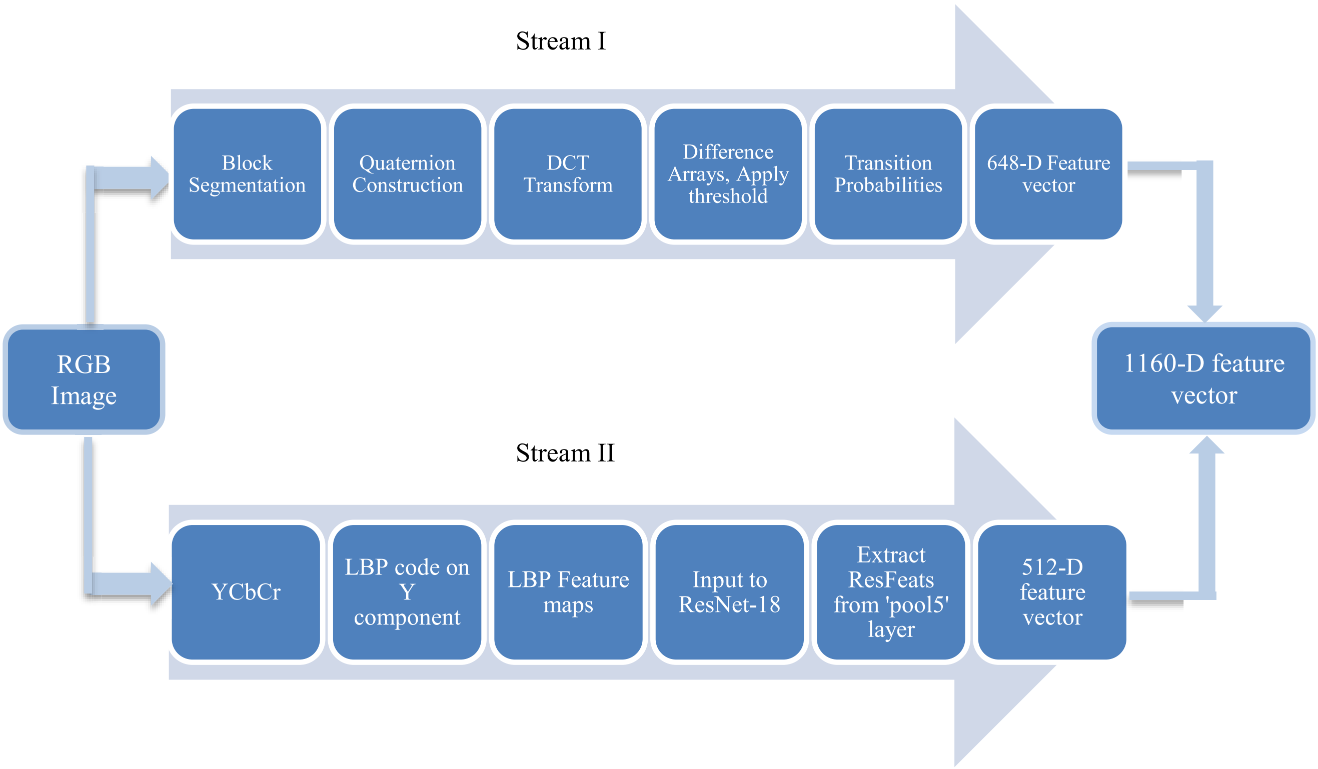

FIGURE 1. **Block diagram of proposed feature-fusion approach.**

```
Algorithm 1 Extract_handcrafted_features
Input: $I_{RGB}$: RGB image
Output: Vector of 648-dimensional handcrafted features, $H_f$
Begin
        Repeat
Load image_file()
Segment the image_file using 8 × 8 blocks
Construct a Quaternion from color channels of the image
Apply Forward DCT transform using Equation (6)
        Until Making_8 × 8_2D matrix;
        Block_Rearrangement();
        Compute_2D_FFT();
        FDCT_coefficients();
        Compute quantisation and dequantisation
        Compute DFT coefficients to further compute 2D IFFT
End Until_finished_image_file
For block_list
        Calculate $Q_V$, $Q_H$, $Q_D$, and $Q_{-D}$ using Equation (7) to
        Equation (10)
        Calculate $R_V$, $R_H$, $R_{D,}$ and $R_{-D}$ using Equation (11) to
        Equation (14)
        Calculate transitional probabilities using Equation (16)
        to Equation (23)
Return $H_f$ of size $(2T + 1) \times (2T + 1) \times 8$ for $T = 4$
```

diagonal and anti-diagonal quaternion Discrete Cosine Transform (DCT) coefficients. The ultimate feature vector H_f is made by calculating the correlations along every possible direction on DCT coefficients of the image.

1) QUATERNION CONSTRUCTION

The notion of quaternions is very widely used in pure and applied mathematics. The quaternions are employed in three-dimensional computer graphics, texture analysis, and computer vision applications. In color image processing, quaternions have proved to be efficient as they consider all the three-color channels, and a color image can be regarded as a vector field holistically. A quaternion is an extended complex quadruple having one fundamental part and three imaginary numbers and is of the form as in Equation (1).

$$\boldsymbol{a} + \boldsymbol{bi} + \boldsymbol{cj} + \mathbf{dk} \tag{1}$$

a, b, c and d correspond to values of continuous quantities where a is the non-imaginary part of the quaternion, and i, j, and k are the basic quaternion units that satisfy the Hamilton rule [48]. It also assures that $i^2 = j^2 = k^2 = ijk = -1$. Quaternions are non-commutative for multiplication (see Table 1), and other basics of quaternions can be found in [48].

A quaternion can be created from two complex numbers with the help of the Clay-Dickson theorem [49]. All the three quaternion units are orthogonal to each other.

Let us assume $m, n \varepsilon C, m = a + bi, n = c + di, a, b, c, d \varepsilon R$, so

$$q = m + nj, \quad q \varepsilon Q, j.j = -1 \tag{2}$$

Hence,

$$q = a + bi + cj + dk \tag{3}$$

TABLE 1. Multiplication of quaternions.

$\downarrow$ **X** $\rightarrow$	**1**	**i**	**j**	**k**
1	1	i	j	k
i	i	−1	k	−j
j	j	−k	−1	i
k	k	j	−i	−1

The transformation given in Equation (3) is used to create the quaternion from complex numbers. The quaternions are also called Hypercomplex numbers [50]. More details on how to construct a quaternion can be found in [49], [50].

Assume μ_1, μ_2 are the two axes of the unit quaternion, and they are mutually perpendicular after that q can be disintegrated into two different complex coordinates in the direction of μ_1 and μ_2.

$$q = m' + n'\mu_2, \quad \mu_2^2 = -1 \tag{4}$$

where $m' = a' + b'\mu_1$, $n = c' + d'\mu_1$, $a', b', c', d' \varepsilon R$ then, $q = a' + b'\mu_1 + c'\mu_2 + d'\mu_3$. Here, $\mu_3 = \mu_1\mu_2$ and μ_3 is perpendicular to μ_1 and μ_2. For the problem being addressed in this article, the coordinates of image (b, c, d) will be transformed into the coordinates (a', b', c', d') under the three axes i.e., μ_1, μ_2 and μ_3. An RGB image can be denoted using a quaternion matrix, as shown in Equation (5).

$$f_q\ (m,n) = f_r\ (m, n)i + f_g\ (m, n)j + f_b\ (m, n)k \tag{5}$$

where $f_r\ (m, n), f_g\ (m, n\ and\ f_b\ (m, n)$ are red, green and blue color components of the image [51].

2) DISCRETE COSINE TRANSFORM ON QUATERNIONS

In literature, most of the techniques based on the Discrete Cosine Transform separate color channels of an image. For example, the YCbCr color space of the image is obtained, and solely Y component is chosen for the detection procedure [52], [53]. This does not help exploit the correlation between all the color channels. On the other hand, QDCT can handle all three channels simultaneously, and color images can be controlled in an integrated manner. Discrete Cosine Transform assists in separating the image into spectral sub-groups of divergent importance, and it uses cosine functions of different wavenumbers as basis functions and works on real-valued spectral coefficients and signals. The cosine functions can bear high energy compaction. DCT components are considerably more focused on origin as compared to other frequency-domain transforms. In the proposed approach, forward quaternion discrete cosine transform (FQDCT) has been used. FQDCT can be categorized into two types: left-handed and right-handed since quaternion multiplication follows a non-commutative rule. So, the forward discrete cosine transform of signal $f_q\ (m, n)$ is given in Equation (6), as shown at the bottom of the next page.

For $0 \leq m < M, 0 \leq n < N$.

3) OBTAIN MARKOV FEATURES OF EACH BLOCK

The steps to calculate the Markov features are provided in [28], and it has been observed that the Markov chain features in DWT and DCT domain have performed considerably well. In our method, the primary stage of extracting the features differs from [28]. The original color images are split into sliding blocks of size 8 × 8. The block obtained after segmentation is still a colored sub-image. The quaternion is made from the R, G, and B components of the sub-image. Further, DCT is applied to obtained quaternion matrix. The coefficients of the transformation are assembled in a matrix, and the square root of the real and the imaginary part is computed. Thus, the final matrix is obtained by arranging the blocks according to the block location. To compute Markov features from the QDCT coefficients, round the coefficients and the absolute values for further processing. Next, compute vertical, horizontal, major diagonal, and minor diagonal distances (Q_V, Q_H, Q_D, and Q_{-D}) within blocks by applying Equation (7) to Equation (10).

$$Q_V\ (U, V) = Q\ (U, V) - Q\ (U, V + 1) \tag{7}$$

$$Q_H\ (U, V) = Q\ (U, V) - Q\ (U + 1, V) \tag{8}$$

$$Q_D\ (U, V) = Q\ (U, V) - Q\ (U + 1, V + 1) \tag{9}$$

$$Q_{-D}\ (U, V) = Q\ (U + 1, V) - Q\ (U, V + 1) \tag{10}$$

where $Q\ (U, V)$ is the matrix containing the rounded off QDCT coefficients. Similarly compute the vertical, horizontal, major diagonal and anti-diagonal differences between the blocks, i.e. inter-block distances (R_V, R_H, R_D and R_{-D}) using Equation (11) to Equation (14).

$$R_V\ (U, V) = Q\ (U, V) - Q\ (U, V + 8) \tag{11}$$

$$R_H\ (U, V) = Q\ (U, V) - Q\ (U + 8, V) \tag{12}$$

$$R_D\ (U, V) = Q\ (U, V) - Q\ (U + 8, V + 8) \tag{13}$$

$$R_{-D}\ (U, V) = Q\ (U + 8, V) - Q\ (U, V + 8) \tag{14}$$

Since difference values obtained in Equation (7) to Equation (14) are integers and contain a broad range so, it is desirable to look for some methods such as rounding-off and thresholding. For this, a threshold T is given, which is a positive integer. If the value of an entity after rounding-off in a different array obtained in Equation (7) to Equation (14) is more than T or less than -T, then the value is substituted with T or -T correspondingly using Equation (15).

$$A_{new} = \begin{cases} -T, & A_{old} < -T \\ T, & A_{old} > T \\ A_{old}, & otherwise. \end{cases} \tag{15}$$

Finally, calculate the above-obtained inter-block and intra-block matrices' transitional probabilities using Equation (16) to Equation (23), as shown at the bottom of the next page. where $i, j \varepsilon \{-\mathrm{T}, -\mathrm{T}+1, -\mathrm{T}+2 \ldots, 0, \ldots \mathrm{T}-2, \mathrm{T}-1, \mathrm{T}\}$, m and n signify the number of rows and columns respectively of the authentic image. $\delta(.) = 1$, only if it satisfies all the arguments otherwise $\delta(.) = 0$. The feature vector's obtained final count is given by $(2\mathrm{T}+1) \times (2\mathrm{T}+1) \times 8$. In our experiments, T is taken as 4. Hence, a 648-dimensional feature vector is obtained. The experiments were also run with thresholds 2, 3, 4, and 5. At thresholds 2 and 3, the accuracy is decreasing. On the other hand, when the threshold is greater than 4, there is no significant change in the accuracy. Moreover, the computational expense increases because the feature vector size is given by $(2T + 1) \times (2T + 1) \times 8$ which is equal to $4T^2 + 4T + 9$ which follows the quadratic growth with the increase in T. And computational complexity is directly proportional to feature size, hence T is directly proportional to computational expense. So, $T = 4$ is chosen for further experimental work.

B. STREAM II: EXTRACTING OFF-THE-SHELF FEATURES 'RESFEATS' FROM RESNET-18

The term ResFeats is given by Mahmood *et al.* [54], where they combined various low-level and high-level features extracted from multiple residual blocks of the network for improved accuracy for underwater image classification. Similarly, we have ResNet-18 architecture as a rich textural feature extractor by inputting local binary pattern codes to the model. The steps for extracting ResFeats are discussed below, and the pseudocode for this Stream is given in Algorithm 2:

1) RGB TO YCbCr

In Stream I, R, G, and B color channels of the image are taken into account, so for Stream II, we want to utilize achromatic characteristics of the image because we have already used chromatic components in Stream I. For that, the luminance component of the YCbCr color space is taken for further processing. YCbCr color space characterizes the color as intensity and exploits the characteristics of a human eye.

Algorithm 2 Extract_deep_features

```
Input: $I_{RGB}$: RGB image
Output: Vector of 512-dimensional deep features, $D_f$
Preprocessing;
$I_{RGB} \rightarrow I_{YCbCr}$
$I_{YCbCr} \rightarrow$ Y_channel
Apply Scale Orientation invariant Local Binary patterns;
SO_LBP = LBP(Y_channel)
Feature_maps = Multi_Dimensional_Scaling(SO_LBP)
Load the pre-trained Resnet18 model;
Net = Load(resnet18)
$D_f$ = Extract_features(Net,
preprocessed_images, layer, 'pool5')
Return $D_f$
```

The advantage of YCbCr color space is that it can separate luminance from chrominance more efficiently than RGB color space. Luminance in the image is light intensity, or the amount of light ranges from black to white. The point of a luminance channel is to capture all of the available (visible) wavelengths at the same time and enable you to concentrate

$$Q(U,V) = \frac{2 \cdot c_U \cdot c_V}{\sqrt{MN}} \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left[f_q(m,n) \cdot \cos\left(\frac{\pi\,(2m+1)\,U}{2M}\right) \cdot \cos\left(\frac{\pi\,(2n+1)\,V}{2N}\right) \right] \tag{6}$$

$$PM1_H(i,j) = \frac{\sum_{U=1}^{m-2} \sum_{V=1}^{n} \delta(Q_H(U,V) = i, Q_H(U+1,V) = j)}{\sum_{U=1}^{m} \sum_{V=1}^{n} \delta(Q_H(U,V) = i)} \tag{16}$$

$$PM1_V(i,j) = \frac{\sum_{U=1}^{m-1} \sum_{V=1}^{n-1} \delta(Q_H(U,V) = i, Q_H(U,V+1) = j)}{\sum_{U=1}^{m-1} \sum_{V=1}^{n-1} \delta(Q_H(U,V) = i)} \tag{17}$$

$$PM2_H(i,j) = \frac{\sum_{U=1}^{m-1} \sum_{V=1}^{n-1} \delta(Q_V(U,V) = i, Q_V(U+1,V) = j)}{\sum_{U=1}^{m-1} \sum_{V=1}^{n-1} \delta(Q_V(U,V) = i)} \tag{18}$$

$$PM2_V(i,j) = \frac{\sum_{U=1}^{m-1} \sum_{V=1}^{n-1} \delta(Q_V(U,V) = i, Q_V(U,V+1) = j)}{\sum_{U=1}^{m-1} \sum_{V=1}^{n-1} \delta(Q_V(U,V) = i)} \tag{19}$$

$$PM3_D(i,j) = \frac{\sum_{U=1}^{m-16} \sum_{V=1}^{n-16} \delta(R_D(U,V) = i, R_D(U+8,V+8) = j)}{\sum_{U=1}^{m-16} \sum_{V=1}^{n-16} \delta(R_D(U,V) = i)} \tag{20}$$

$$PM3_{-D}(i,j) = \frac{\sum_{U=1}^{m-16} \sum_{V=1}^{n-16} \delta(R_{-D}(U+8,V) = i, R_{-D}(U,V+8) = j)}{\sum_{U=1}^{m-16} \sum_{V=1}^{n-16} \delta\,(R_{-D}(U,V) = i)} \tag{21}$$

$$PM3_H(i,j) = \frac{\sum_{U=1}^{m-8} \sum_{V=1}^{n-8} \delta(R_V(U,V) = i, R_V(U+8,V) = j)}{\sum_{U=1}^{m-8} \sum_{V=1}^{n-8} \delta(R_V(U,V) = i)} \tag{22}$$

$$PM3_V(i,j) = \frac{\sum_{U=1}^{m-8} \sum_{V=1}^{n-8} \delta(R_H(U,V) = i, R_H(U,V+8) = j)}{\sum_{U=1}^{m-8} \sum_{V=1}^{n-8} \delta(R_H(U,V) = i)} \tag{23}$$

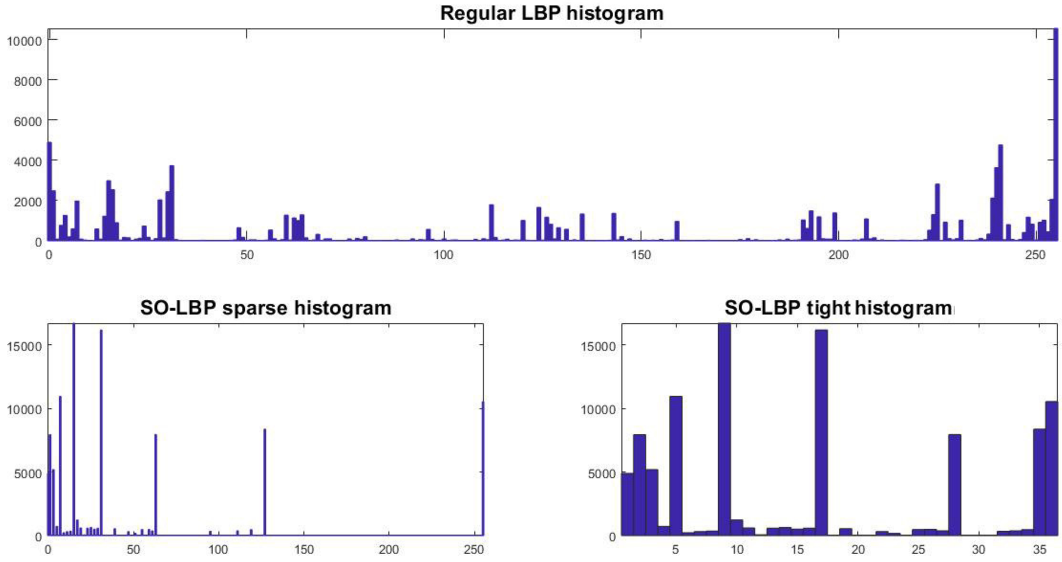

FIGURE 2. **Histograms of regular LBP codes and SO-LBP codes.**

your efforts of noise reduction (via stacking), sharpness, and detail into one layer. YCbCr representation can be obtained from RGB colorspace, as shown in Equation (24).

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.279 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{24}$$

2) OBTAIN SCALE AND ORIENTATION INVARIANT LOCAL BINARY PATTERN FEATURE MAPS FROM THE Y COMPONENT OF THE IMAGE

Local Binary Patterns are used in various image processing applications for texture-based classifications and have been demonstrated to be highly discriminative. The only drawback of LBP is that it is susceptible to affine transformations. To make up for image rotations, a broadly utilized pivot invariant encoding of LBP portrayal is used, which necessitates either an implicit or express arrangement of patterns that is done at the encoding level. Various variants of LBP have been introduced earlier that have diminished the feature representations' discriminative intensity by expanding the capacity to characterize the core microstructures at numerous scales. Rotation-invariant substitute of LBP is also unfit to make up for image scaling

Consequently, we have utilized scale and orientation resistant LBP [55] on the above acquired ELA images. This is accomplished by processing scale-invariant features and pivot-invariant features independently and afterward consolidating both feature representations to improve the discriminative intensity of the LBP. Orientation invariance is achieved by adjusting the features at the extraction level utilizing a strong global estimator. Scale-adjusted features are determined concerning the image's assessed size based on a conveyance of scale standardized Laplacian reactions in a scale-space representation. Final SO-LBP is obtained by combining the above two features in a multi-scale representation. The occurrences of the SO-LBP codes in the image are collected into a histogram. Figure 2 shows the histogram of regular LBP codes, sparse histogram of SOLBP codes, and tight histogram of SO-LBP codes of a forged image.

The obtained SO-LBP codes cannot be directly given to the deep neural network architecture. So to feed them to the architecture, we have to convert the SO-LBP into feature maps. SO-LBP features are converted into LBP maps using multi-dimensional scaling to change the pattern values into points in a metric space. Convolutional operations can average together the transformed points, yet their distances are approximately equal to the original code-to-code distances. Distance reflects the core resemblance of the image intensity arrangements used to create every LBP code sequence. A complete disparity grid represents the distances among all the potential code values. For a given disparity grid, MDS looks to map the codes to a low-dimensional measurement space. Further, to account for the distinctions in spatial areas of pixel code patterns instead of Hamming distance, we utilize the Earth 'Mover's Distance (EMD). EMD is characterized to mirror the slightest exertion needed to change over one dispersion into another. It is being used at this point as a proportion of the contrast among two LBP codes. EMD approximation is performed rather than computing the true EMD between code strings. The obtained SO-LBP maps are further fed to the pre-trained ResNet-18 model to get the deep textural ResFeats.

3) OBTAIN 'RESFEATS' FROM RESNET-18

Residual Networks [43] are analogous to networks with convolution, pooling, activation and fully-associated layers arranged over one another. The only distinction is that it has an identity connection between the layers. Through identity connections, the network straightforwardly tries to find out the output functions with no further support. A pre-trained deep neural network named ResNet-18 is used, which is an 18-layer deep model. There are other wider residual networks available such as ResNet-34, ResNet-50, etc. These networks are overly deep and do not converge fast. The obtained SO-LBP feature maps are resized to 224 × 224 × 3 to input to the deep neural model. The pre-trained model is loaded with 'ImageNet' weights. The in-depth features are extracted after the last convolutional batch, i.e., from the 'pool5' layer called ResFeats. The extracted features are 512-dimensional vectors, D_f.

C. CLASSIFICATION OF FUSED FEATURES USING A SHALLOW NEURAL NETWORK

After obtaining 648-D handcrafted features and 512-D ResFeats, both are merged to form an 1160-D feature vector of images, i.e., F_f using Algorithm 3. To avoid the outlier issue, the dataset of feature vectors is normalized using the z-score strategy. The normalized 1160-D feature vectors are categorized using a shallow neural network (SNN) having two feed-forward layers with a sigmoid function on the hidden layer and a softmax function for the output layer. The structure of the network is shown in Figure 3. The network is trained using the scaled conjugate gradient method for weight and bias value updations.

```
Algorithm 3 Fused_features
    Input: $H_f$, $D_f$
    Output: Fused features, $F_f$
    Concatenate $H_f$ and $D_f$
    $F_f \rightarrow [H_f, D_f]$
    Return $F_f$
```

Table 2 shows the parameters used for training the network. Min_grad is the minimum performance gradient before training is stopped. When the performance gradient becomes too tiny, continued training is unlikely to produce significant improvements. Max_fail parameter determines the maximum number of validation checks before training is stopped. Lambda is used for regulating the indefiniteness of the Hessian. Sigma is another training function parameter that determines the weight change for the second derivative approximation during training

IV. EXPERIMENTAL SETUP AND RESULTS

The experiments are performed on a 10th Gen Core i5 Processor with 8 GB of RAM and 4 GB of NVIDIA 1650 Ti Graphics. The handcrafted feature extraction code is written in Python, whereas ResFeats are extracted using MATLAB 2018b platform. The features are fused using MATLAB code and the classification is also performed on the MATLAB platform using Neural Network toolbox. The SNN is trained using the Scaled Conjugate Gradient method. For handcrafted feature extraction, various python modules and libraries were used, namely, cmath, geopandas, Descartes, PySAL(Python Spatial Analysis Library), pyquaternion, and SciPy library. Two accessible standard datasets for image tampering detection have been used for the assessment of the projected approach. The datasets considered in this work are CASIA TIDE v1 [56] and CASIA TIDE v2 [56]. These datasets are delivered by the Chinese Academy of Sciences. The particulars of the datasets are mentioned in Table 3. Both the datasets contain mixed kinds of forgeries (copy-move and splicing).

TABLE 2. Parameters used for training the network for classification.

Parameter	Value
Epochs	1000
Min_grad	1e-6
Maximum time to train (in seconds)	inf
Max_fail	6
Lambda	5.0e-7
Sigma	5.0e-5

TABLE 3. Details of the datasets used for evaluation.

Dataset	original images	modified images	Image resolution	Image format
CASIA v1.0	800	925	384 × 256	JPEG
CASIA v2.0	7491	5123	240×160 to 900×600	JPEG, TIFF, BMP

For assessing the proposed approach's performance, cross-entropy is calculated for each output-target element using Equation (25). For graphical representation, the ROC curve, Error Histogram (EH), and performance plot are shown.

$$CE = -t. \times \log(y); \tag{25}$$

where t is the target value, and y is the predicted value. The total cross-entropy performance is the mean of the individual values. Minimizing cross-entropy results in good classification, so lower cross-entropy values are good, whereas zero means no error. Percentage error (%E) is also calculated that indicates the fraction of samples that are misclassified. A value of zero means no misclassification, and a value of 100 specifies maximum misclassifications.

In the first experiment, the results are achieved using only handcrafted Markov-based features. Classification of the 648-D feature vector is performed using the shallow neural network discussed in previous sections. For this purpose, the 1160-D feature vector in Figure 3 is replaced by a 648-D engineered feature vector while keeping all the processing steps the same. Figure 4 shows the ROC curve obtained on CASIA v1 and CASIA v2 for handcrafted features. The area under the curve for CASIA v2 is larger than CASIA v1.

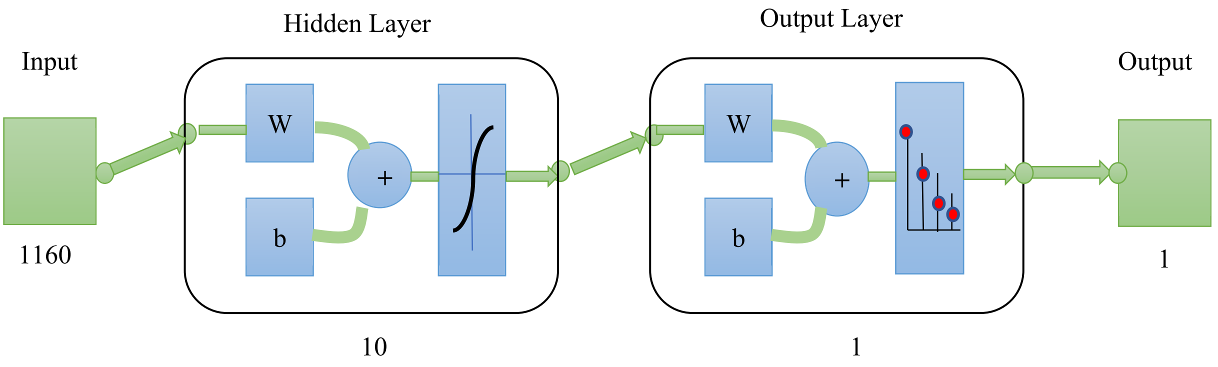

FIGURE 3. Structure of the shallow neural network for classification.

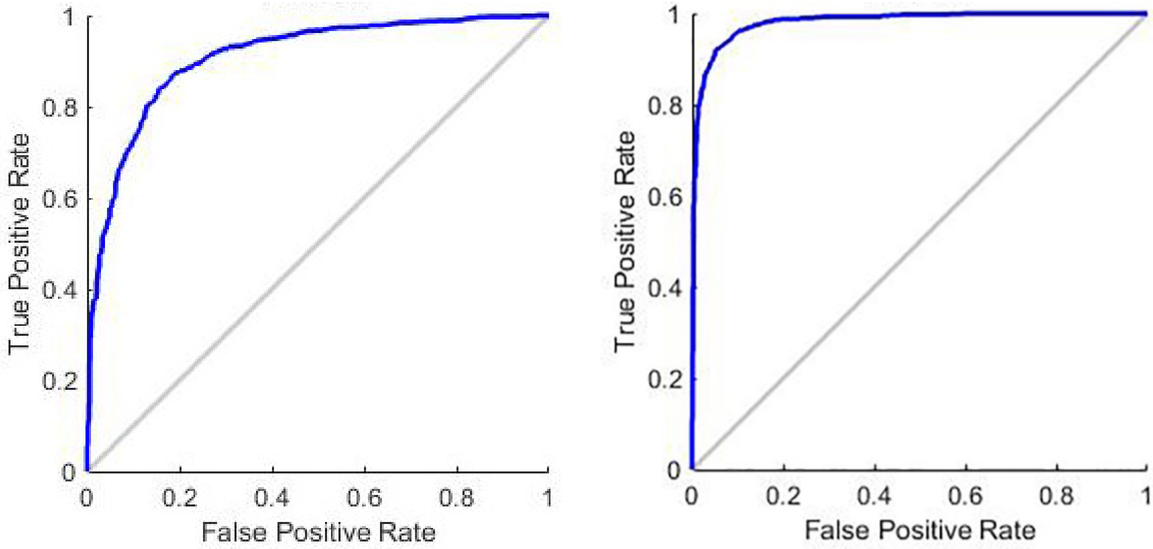

FIGURE 4. ROC curve for handcrafted features for CASIA v1 and CASIA v2, respectively.

The handcrafted features performed well for low-resolution images with the use of transitional probabilities in the frequency domain. This difference in results of these datasets is because the two databases are different, and accordingly, have different qualities of authentic and forged images such as resolution and image file format types. Also, these outcomes exhibit that the manually engineered image features have a considerable difference in detection accuracy conditional to the dataset's attributes. Hence, this issue can decrease the trustworthiness of the detection system that utilizes just engineered features.

In the second experiment, the performance of ResFeats is tested on both datasets, and the ROC curve for both datasets is shown in Figure 5. For execution, a pre-trained ResNet-18 model that was effectively trained on the ImageNet database is used. The model is loaded with ImageNet weights to initialize the parameters of the deep neural network. Features are extricated from the 'pool5' layer. For classification, our SNN model parameters are well-initialized, and the subsequent training process has shown rapid convergence. The network takes 5 seconds and 28 epochs to converge. The deep high-level features can detect the images having scaled and rotated forgeries. The experiments were also conducted by using Cb and Cr channels in the preprocessing step. The detection accuracy for the blue chrominance channel (Cb) was 72.17% and 79.8% for CASIA v1 and CASIA v2, respectively. Similarly, the detection accuracy has been checked for the red chrominance channel (Cr), and the values obtained for CASIA v1 and CASIA v2 were 60.34% and 72.5%, respectively.

In the final set of experiments, we tested the performance of the proposed combined features to detect forgeries. The ROC curve shown in Figure 6 depicts the performance of combined features classified using SNN. It is pretty clear from the graphs that the performance has improved as the curve is more towards the top left corner than the curves in Figure 4 and Figure 5. The combination of the features has resulted in improving the detection accuracy as both the features have complemented each other to overcome the weakness of the other.

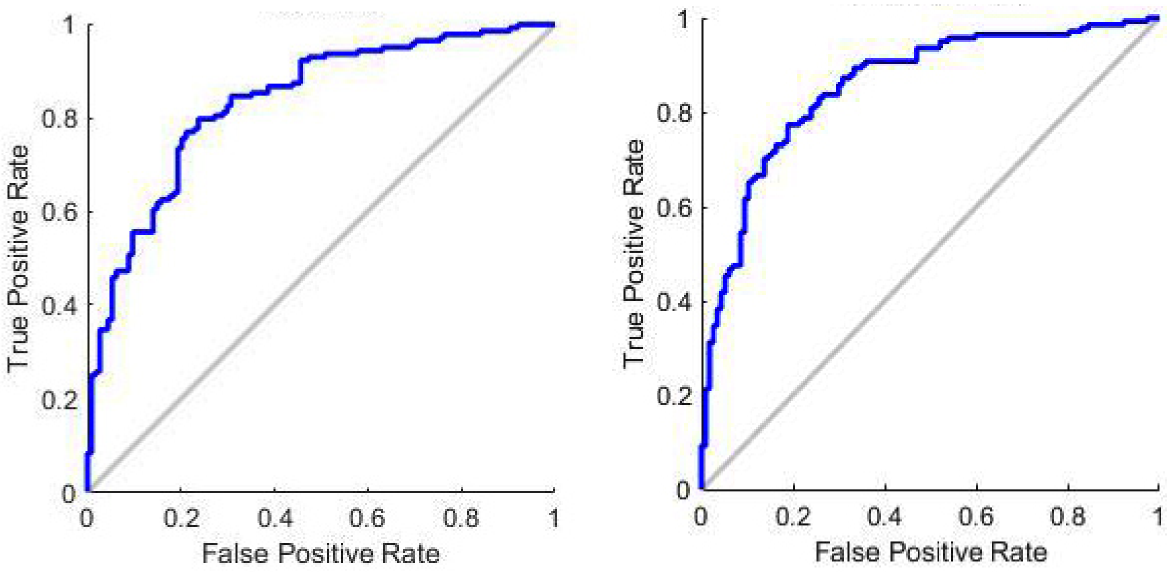

FIGURE 5. ROC curve for ResFeats for CASIA v1 and CASIA v2, respectively.

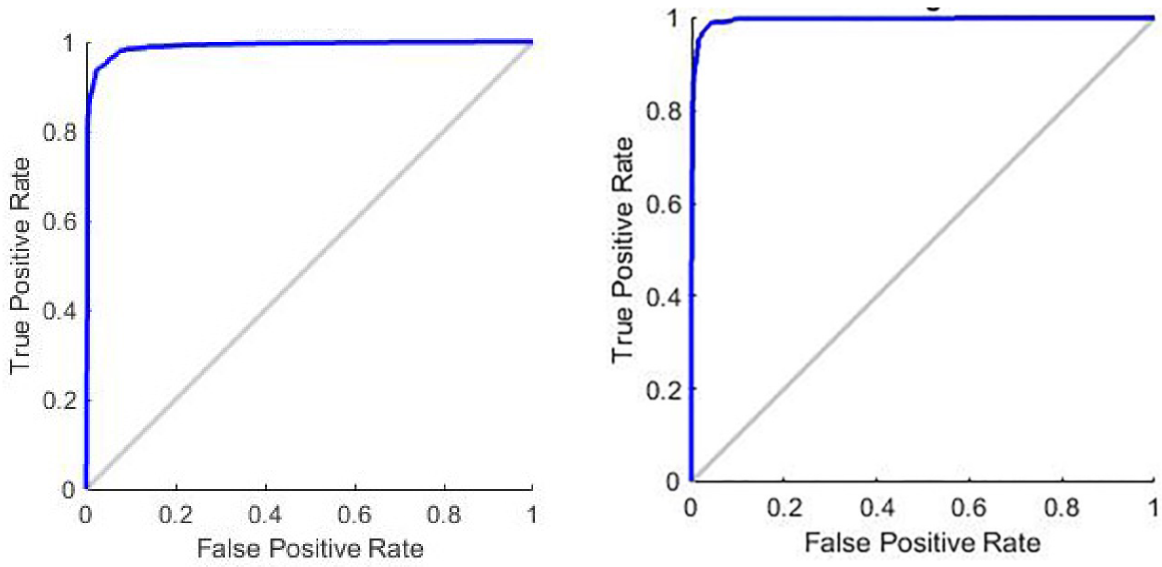

FIGURE 6. ROC curve for proposed approach on CASIA v1 and CASIA v2, respectively.

TABLE 4. Cross entropy and percentage error for CASIA v1 and CASIA v2 datasets.

Datasets		CE	%E
CASIA v1	Training	4.19634e-1	4.39045e-0
	Validation	1.37362e-0	6.00917e-0
	Testing	1.35698e-0	8.09174e-0
CASIA v2	Training	5.38857e-1	1.55068e-0
	Validation	1.08575e-0	2.89296e-0
	Testing	1.07824e-0	3.504580e-0

Table 4 shows the cross-entropy error and percentage error for CASIA v1 and CASIA v2 to classify combined features using SNN. The cross-entropy error and percentage error for the training, validation, and testing phase have been calculated. Further, results are also shown in the form of an error histogram and performance plot to analyze the proposed approach's results. Error histogram is used to visualize errors between target values and predicted values after training the SNN. It can be seen from Figure 7 that most of the errors range between -0.05089 to 0.04901. Around 700 instances from the test dataset and 701 instances from the validation dataset each have an error value of -0.05089. Similarly, 850 instances from the test dataset and 853 instances from the validation dataset have an error value of 0.04901. It can be seen in Figure 8 that the errors are more prominent as compared to the case of the CASIA v2 in Figure 7.

The method performed better in terms of accuracy as contrasted to state-of-art methods. Table 5 shows the comparison of the accuracy of handcrafted features alone, ResFeats and combined features, and other state-of-art methods based

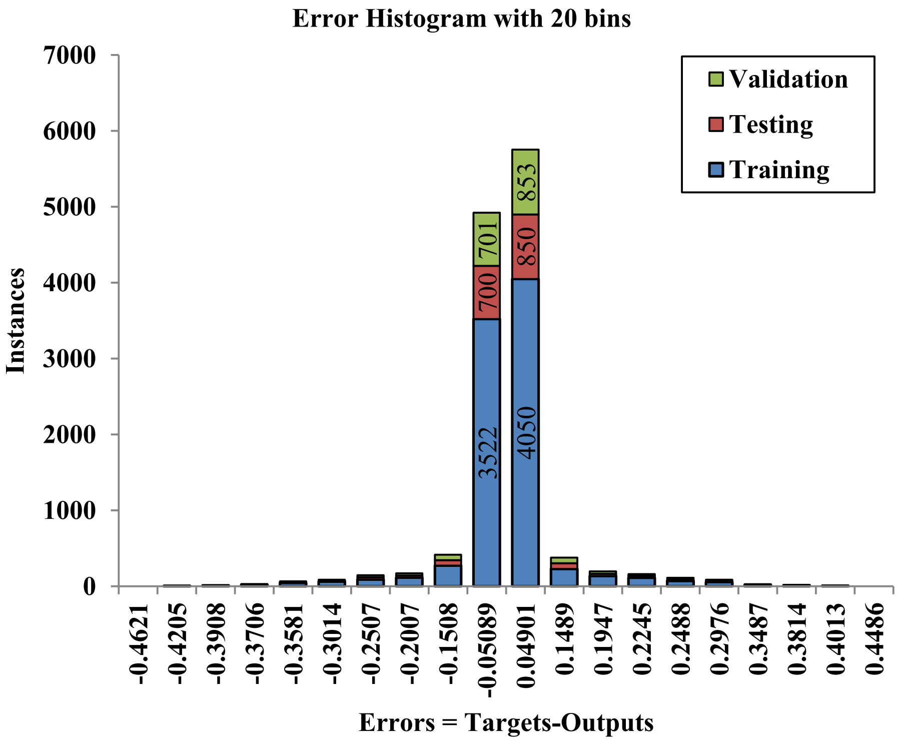

FIGURE 7. Error Histogram for proposed approach for CASIA v2.

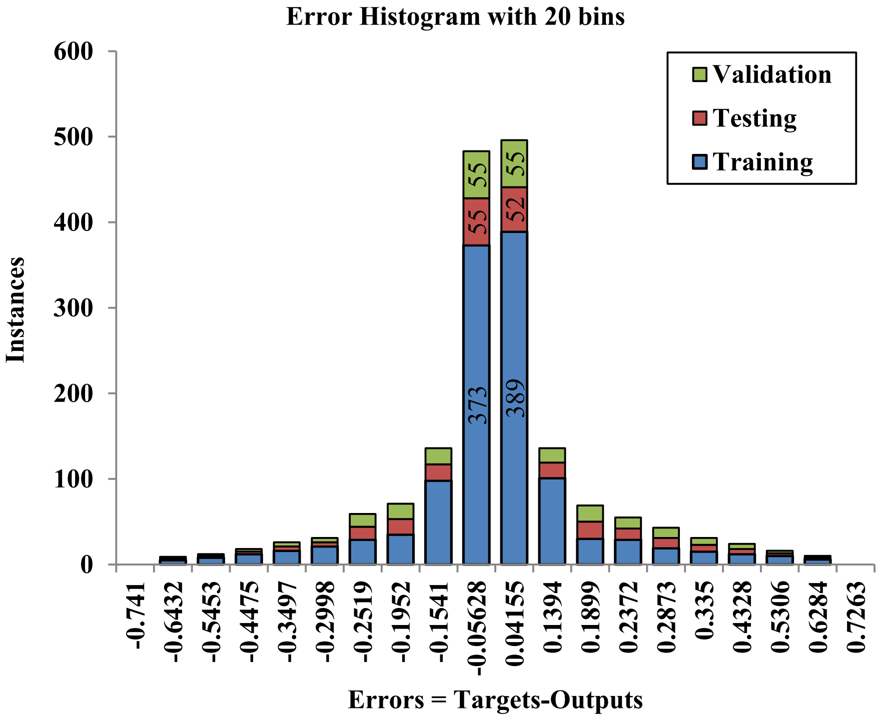

FIGURE 8. Error Histogram for proposed approach on CASIA v1 dataset.

on feature fusion for digital image manipulation detection. It is also observed that results for CASIA v1 are inferior to CASIA v2. The major reason for that is the difference between the two datasets. CASIA v1 is a much smaller dataset as compared to the CASIA v2 dataset, moreover, CASIA v1 contains only JPEG compressed images whereas

TABLE 5. Comparison of accuracy (%) on CASIA v1 and CASIA v2 datasets.

Method		CASIA v1	CASIA v2	Feature_set	Dimensionality
Handcrafted features based	Multi-texture[23]	94.13	97.03	LBP, LPQ, BSIF, BGP	19680
	Texture features[24]	97.4	98.6	HOSF, HOG, and LBP	-
	Entropy Filter and LPQ[25]	95.41	98.33	Multi-scale LPQ	-
	DWT and LBP[26]	-	94.09	Texture features	1024
	Expansion and Maximization[27]	97.86	97.33	Markov based features	-
	Proposed Stream I	94.56	97.21	Markov based features	648
Deep learning-based	Modified AlexNet[44]	96.85	97.44	-	-
	SRM-CNN[45]	98.04	97.83	CNN-based descriptor	
	Stacked Auto-Encoder (SAE)[46]	-	91.09	3 Level 2D Daubechies Wavelet	450
	Proposed Stream II	83.34	87.58	Deep textural features	512
Proposed approach	Fusion of handcrafted and deep features	**97.94**	**99.31**	'ResFeats' and Markov based features	1160

CASIA v2 contains JPEG, TIFF, and BMP images. TIFF is a lossless file format and BMP images are uncompressed. A lot of information is lost in the process of JPEG compression which affects the detection results. It is witnessed that the proposed method either outdone or yielded competitive results.

V. CONCLUSION

Shrewd attackers craft forgeries in the digital images so that state-of-art forensics tools cannot track anomaly characteristics accurately. On the other hand, deep learning methods are well known to provide the wisdom to formulate high-level features suitable for classification problems. Moreover, carefully designed handcrafted features extracted from images also perform very well with comparatively good accuracies. However, after much experimentation by researchers, attackers hold an edge over state-of-art deep learning methods and manually engineered features in case of forgery detection. Therefore, this paper proposes a novel feature fusion-based approach that exploits RGB color space and luminance channels to trap forgeries in digital images. Our experiments demonstrated that the manually engineered Markov features based on transitional probabilities along four directions in Quarternion Discrete Cosine Transform are appropriate for identifying forged images using RGB color characteristics of low-resolution images. On the other hand, high-level deep image features based on luminance components and textural characteristics have performed up to the mark for false-negative cases of manually engineered features. Moreover, LBP-based pre-processing of raw images has immensely improved the classification of scaled and rotated images in benchmarked datasets. Consequently, by joining the two categories of image features, the detection accuracy is fundamentally upgraded compared to using a single method and other contemporary methods. The proposed fusion-based approach used 1160-D features and has state-of-art accuracy of 99.3% and 97.94% on CASIA v1 and CASIA v2 datasets, respectively. The authors are also investigating the reliability of the approach by implementing the proposed approach on other datasets as well. The proposed approach is promising for offline forensic analysis of digital images. However, for real-time analysis high dimensionality of fused features is the primary bottleneck. The authors will investigate novel approaches to decrease the feature complexity of the proposed fusion-based approach in the future.

CONFLICT OF INTEREST

All the authors declared that they have no conflict of interest in this work

REFERENCES

[1] Z. Zhang and Y. Zhou, "Study of image splicing detection," *Adv. Intell. Comput. Theor. Appl. Asp. Theor. Methodol.*, vol. 5226, pp. 1103–1110, Sep. 2008.

[2] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Res. Workshop*, 2003, pp. 55–61.

[3] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 845–850, Feb. 2005.

[4] H. Farid, "Image forgery detection," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[5] A. Haouzia and R. Noumeir, "Methods for image authentication: A survey," *Multimedia Tools Appl.*, vol. 39, no. 1, pp. 1–46, Aug. 2008.

[6] S. Walia and K. Kumar, "Digital image forgery detection: A systematic scrutiny," *Austral. J. Forensic Sci.*, vol. 51, no. 5, pp. 488–526, Sep. 2019.

[7] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[8] N. B. A. Warif, A. W. A. Wahab, M. Y. I. Idris, R. Salleh, and F. Othman, "SIFT-symmetry: A robust detection method for copy-move forgery with reflection attack," *J. Vis. Commun. Image Represent.*, vol. 46, pp. 219–232, Jul. 2017.

[9] F. Yang, J. Li, W. Lu, and J. Weng, "Copy-move forgery detection based on hybrid features," *Eng. Appl. Artif. Intell.*, vol. 59, pp. 73–83, Mar. 2017.

[10] B. Soni, P. K. Das, and D. M. Thounaojam, "CMFD: A detailed review of block based and key feature based techniques in image copy-move forgery detection," *IET Image Process.*, vol. 12, no. 2, pp. 167–178, Feb. 2018.

[11] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. IEEE Multimedia Expo Int. Conf.*, Jul. 2007, pp. 1750–1753.

[12] S. Bayram, H. Taha Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1053–1056.

[13] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, "Passive detection of image forgery using DCT and local binary pattern," *Signal, Image Video Process.*, vol. 11, no. 1, pp. 81–88, Jan. 2017.

[14] T. Mahmood, T. Nawaz, R. Ashraf, M. Shah, Z. Khan, A. Irtaza, and Z. Mehmood, "A survey on block based copy move image forgery detection techniques," in *Proc. Int. Conf. Emerg. Technol. (ICET)*, Dec. 2015, pp. 1–6.

[15] M. Ali and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Process. Image Commun.*, vol. 39, pp. 46–74, Nov. 2015.

[16] I. Amerini, R. Becarelli, R. Caldelli, and A. Del Mastio, "Splicing forgeries localization through the use of first digit features," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 143–148.

[17] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5302–5306.

[18] G. Muzaffer and G. Ulutas, "A new deep learning-based method to detection of copy-move forgery in digital images," in *Proc. Sci. Meeting Elect.-Electron. Biomed. Eng. Comput. Sci.*, Apr. 2019, pp. 1–4.

[19] R. K. Remya and M. Wilscy, "Pretrained convolutional neural networks as feature extractor for image splicing detection," in *Proc. Int. Conf. Circuits Syst. Digit. Enterprise Technol. (ICCSDET)*, Dec. 2018, p. 4.

[20] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, Dec. 2016.

[21] E.-S.-M. El-Alfy and M. A. Qureshi, "Combining spatial and DCT based Markov features for enhanced blind detection of image splicing," *Pattern Anal. Appl.*, vol. 18, no. 3, pp. 713–723, Aug. 2015.

[22] X. Zhao, S. Wang, S. Li, and J. Li, "Passive image-splicing detection by a 2-D noncausal Markov model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 185–199, Feb. 2015.

[23] D. S. Vidyadharan and S. M. Thampi, "Digital image forgery detection using compact multi-texture representation," *J. Intell. Fuzzy Syst.*, vol. 32, no. 4, pp. 3177–3188, Mar. 2017.

[24] A. R. Abrahim, M. S. M. Rahim, and G. B. Sulong, "Splicing image forgery identification based on artificial neural network approach and texture features," *Cluster Comput.*, vol. 22, no. S1, pp. 647–660, Jan. 2019.

[25] S. Agarwal and S. Chand, "Image forgery detection using multi scale entropy filter and local phase quantization," *Int. J. Image, Graph. Signal Process.*, vol. 10, pp. 78–85, Feb. 2015.

[26] M. Kaur and S. Gupta, "A passive blind approach for image splicing detection based on DWT and LBP histograms," in *Proc. Int. Symp. Secur. Comput. Commun.*, 2016, pp. 318–327.

[27] J. G. Han, T. H. Park, Y. H. Moon, and I. K. Eom, "Efficient Markov feature extraction method for image splicing detection using maximization and threshold expansion," *J. Electron. Imag.*, vol. 25, no. 2, Apr. 2016, Art. no. 023031.

[28] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognit.*, vol. 45, no. 12, pp. 4292–4299, 2012.

[29] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2285–2294.

[30] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[31] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond frame-level CNN: Saliency-aware 3-D CNN with LSTM for video action recognition," *IEEE Signal Process. Lett.*, vol. 24, no. 4, pp. 510–514, Apr. 2017.

[32] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2017.

[33] R. He, X. Wu, Z. Sun, and T. Tan, "Wasserstein CNN: Learning invariant features for NIR-VIS face recognition," 2017, *arXiv:1708.02412*. [Online]. Available: http://arxiv.org/abs/1708.02412

[34] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[35] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Computer Vision* (Lecture Notes in Computer Science), vol. 10111. Cham, Switzerland: Springer, 2017, pp. 213–228.

[36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, Nov. 1998.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] A. Fred Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*. [Online]. Available: http://arxiv.org/abs/1803.08375

[41] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 730–734.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

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

[44] S. Samir, E. Emary, K. El-Sayed, and H. Onsi, "Optimization of a pre-trained AlexNet model for detecting and localizing image forgeries," *Information*, vol. 11, no. 5, pp. 1–19, 2020.

[45] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. 8th IEEE Int. Work. Inf. Forensics Secur. WIFS*, Dec. 2016, pp. 1–6.

[46] Y. Zhang, J. Goh, L. Lei, and V. Thing, "Image region forgery detection: A deep learning approach," in *Proc. Singap. Cyber-Secur. Conf.*, vol. 14, 2016, pp. 1–11.

[47] D. T. Nguyen, T. D. Pham, N. R. Baek, and K. R. Park, "Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors," *Sensors*, vol. 18, no. 699, pp. 1–29, 2018.

[48] J. Barbic, "Quaternions and rotations," Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. 520, 2011, pp. 1–12.

[49] C. E. Moxey, S. J. Sangwine, and T. A. Ell, "Hypercomplex correlation techniques for vector images," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1941–1953, Jul. 2003.

[50] I. Kantor, A. Solodovnikov, and A. Shenitzer, *Hypercomplex Numbers: An Elementary Introduction to Algebras*. Berlin, Germany: Springer, 1989.

[51] W. K. Wong, C. K. Loo, W. S. Lim, and P. N. Tan, "Quaternion based thermal condition monitoring system," *Natural Comput. Inf. Commun. Technol.*, vol. 2, pp. 352–362, Apr. 2010.

[52] M. Kaur and S. Walia, "Forgery detection using noise estimation and HOG feature extraction," *Int. J. Multimedia Ubiquitous Eng.*, vol. 11, no. 4, pp. 37–48, Apr. 2016.

[53] M. Hussain, S. Q. Saleh, H. Aboalsamh, G. Muhammad, and G. Bebis, "Comparison between WLD and LBP descriptors for non-intrusive image forgery detection," in *Proc. IEEE Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Jun. 2014, pp. 197–204.

[54] A. Mahmood, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "ResFeats: Residual network based features for underwater image classification," *Image Vis. Comput.*, vol. 93, Jan. 2020, Art. no. 103811.

[55] S. Hegenbart and A. Uhl, "A scale-and orientation-adaptive extension of local binary patterns for texture classification," *Pattern Recognit.*, vol. 48, no. 8, pp. 2633–2644, Aug. 2015.

[56] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, Jul. 2013, pp. 422–426.

SAVITA WALIA received the B.Tech. degree in computer science and engineering (CSE) from Punjab Technical University, Jalandhar, India, in 2012, and the M.E. degree in information technology from the University Institute of Engineering and Technology (UIET), Panjab University, Chandigarh, India, where she is currently pursuing the Ph.D. degree with the Faculty of Engineering and Technology. She has published more than ten research papers in refereed journals and conferences. Her research interests include image processing, digital image forensics, and information security.

KRISHAN KUMAR received the B.Tech. degree in computer science and engineering from the National Institute of Technology at Hamirpur, Hamirpur, in 1995, the master's degree in software systems from the Birla Institute of Technology and Sciences at Pilani, Pilani, in 2001, and the Ph.D. degree from the Indian Institute of Technology at Roorkee, Roorkee, in February 2008. He is currently a Professor with the Department of Information Technology, University Institute of Engineering and Technology, Panjab University, Chandigarh. He has more than 22 years of teaching, research, and administrative experience. He has published two national and two international books in the field of computer science and network security. He has also published more than 150 articles in national/international peer reviewed/indexed/impact factor journals and IEEE, ACM, and Springer proceedings. His publications are well cited by eminent researchers in the field. His general research interests include the areas of network security and computer networks. His specific research interests include intrusion detection, protection from Internet attacks, Web performance, network architecture/protocols, and network measurement/modeling.

MUNISH KUMAR received the master's degree in computer science and engineering and the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2008 and 2015, respectively. He started his career as an Assistant Professor in computer science with the Jaito Centre, Punjabi University, Patiala. He is currently working as an Assistant Professor with the Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India. He has guided five Ph.D. research scholars. He has published five international patents. He has published more than 100 research articles in the reputed international journals and conference proceedings. He has more than 1000 citations for his articles at Google Scholar. His research interests include character recognition, handwriting recognition, computer vision, machine learning, and pattern recognition. He is a Professional Member of IEEE.

XIAO-ZHI GAO received the B.Sc. and M.Sc. degrees from the Harbin Institute of Technology, China, in 1993 and 1996, respectively, and the D.Sc. (Tech.) degree from the Helsinki University of Technology (now Aalto University), Finland, in 1999. He has been working as a Professor with the University of Eastern Finland, Finland, since 2018. He is a Guest Professor at the Harbin Institute of Technology, Beijing Normal University, and Shanghai Maritime University, China. He has published more than 400 technical papers in refereed journals and international conferences. His current Google Scholar H-index is 34. His research interests include nature-inspired computing methods with their applications in optimization, data mining, machine learning, control, signal processing, and industrial electronics.