I.J. Image, Graphics and Signal Processing, **2026, 1, 33-50**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2026.01.03

An Effective Semi-Supervised Feature Extraction Model with Reduced Architectural Complexity for Image Forgery Classification

Jisha K. R.*
APJ Abdul Kalam Technological University, Thiruvananthapuram, Department of Electronics and Communication Engineering, Rajagiri School of Engineering and Technology, Kakkanad, Kerala, India
E-mail: jishabaj@gmail.com
ORCID iD: https://orcid.org/0009-0002-8670-0691
*Corresponding Author

Sabna N.
Department of Electronics and Communication Engineering, Rajagiri School of Engineering and Technology, Kakkanad, Kerala, India
E-mail: sabnan@rajagiritech.edu.in
ORCID iD: https://orcid.org/0000-0002-8565-8412

Received: 05 June, 2025; Revised: 12 August, 2025; Accepted: 06 January, 2026; Published: 08 February, 2026

Abstract: A generalized deep learning approach tracking image forgeries of any category with reduced architectural complexity, without compromising the performance is presented in this paper. A convolutional encoder-decoder architecture-based image reconstruction model is framed to extract all the pertinent information from the images. Performance comparison of similar networks constructed with varying architectural complexity led to the selection of this design. The best reconstruction feature extractor showed faster convergence and improved accuracy, as observed from the training and validation performance curves. Dimensionally compressed information from the reconstruction model is utilized by dense layers and further classified. Experimenting with forgery datasets inclusive of different forgery types ensured the generalizability of the model. In comparison with the reconstruction models adopting transfer learning in the encoder side utilizing MobileNet, ResNet 50, and VGG 19, the proposed model exhibited competitive and consistently improved mean Precision and F1-score performance across multiple datasets, as validated through multi-seed experimentation. Additionally, with the reduced architecture, the proposed model performed on par than the state-of-the-art approaches against which it was compared.

Index Terms: Encoder-Decoder Architecture, Feature Extraction, Generalizability, Image Forgery Detection, Reduced Architectural Complexity

1. Introduction

Images are used in many different sectors and are essential to our everyday lives. They are the primary source of communication and information, and they also provide evidence for forensics, journalism, media, police investigations, and other fields. The validity of images has been compromised by the accessibility of open access and availability of image editing tools and software. This cleared the path for the growth of studies in localisation and image forgery detection [1]. Active and passive image forgery detection are the two main kinds of image forgery detection [2]. In order to verify and recover the original image, active techniques such as digital watermarks and digital signatures, need some possible information about the image beforehand. Regardless of any prior knowledge about the image, the passive or blind procedures verify the image authenticity. In the latter instance, forgeries are identified using the hints given by the different phases of image collection and storage.

Extensive research is being conducted in the field of Passive forgery detection. Earlier researches concentrated on handcrafted feature extraction algorithms [3, 4] which were tedious and complex. But, at present researches are centred on Deep Learning[5] which highlights automatic feature extraction capabilities. The development of this phase has madethe complex feature extraction process much easy without human intervention. There are several research works that use deep learning models to solve different areas of manipulation detection independently. There are several deep

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

learning models that use explicit feature extraction methods to handle and extract specific visual characteristics. These models will only be able to identify a certain kind of fake [6,7]. Since the individual conducting the analysis may not be aware of the type of fraud, a generalised model that can identify any kind of forgery can only be helpful. Furthermore, recent studies show that deep learning models enhance performance at the expense of architectural complexity because of their intricate designs and lengthy runtimes. Additionally, a large amount of data is needed for training. The concept of transfer learning has helped in reducing long training times[8]. However, altering the transfer learning model architecture and handling data that is distinct from the dataset with which the model is already trained, poses significant challenges. Furthermore, because of the large dimensions, it is challenging to categorise the characteristics that are taken from the fully linked layers of transfer learnt CNN models into binary classifications. Because of these factors, it is essential to have an effective generalised model that can identify image counterfeit without sacrificing performance or necessitating a complex architecture.

This research proposes an effective deep learning model for feature extraction in conjunction with a classification network for the identification of image forgeries. To extract the relevant features contributing to the classification of forgeries, at first, all the pertinent information of the images is extracted and those features are used by a classifier network, towards identifying and classifying the features relevant for forgery detection. An image reconstruction model framework with reduced architectural complexity is the best option for extracting all the relevant information. Data from recent studies demonstrate that autoencoders may effectively retrieve context information from an image[9]. Furthermore, the idea of selecting the encoder-decoder architecture for feature extraction was reinforced by their effectiveness in dimensionality reduction. For this reason, an encoder-decoder architecture-based image reconstruction, with less dense networks is meticulously developed by evaluating performance using appropriate ablation studies, leading to a comparison with two additional model frameworks. The best image reconstruction features are visualized using heatmaps and are utilized by the classifier network for classification. Most deep learning based works concentrated on the detection of copy move and splicing types of image forgeries. But in this study, a more generalized model is developed. To guarantee this, the proposed model is trained using 4 different datasets including image tampering such as copy move, splicing, cut paste, retouching and colorizing. In short, an effective simplified approach based on encoder-decoder architecture with reduced complexity in architecture with better performance, is proposed in this paper. The capability of the model is experimentally verified by comparing its architecture and performance with certain state-of-the-arts and with transfer learning models utilizing VGG19, ResNet 50 and MobileNet as the backbones of encoder.

The remaining sections of the paper are structured as follows: The relevant works in this topic are put together in Section 2, the proposed framework is outlined in Sections 3, the experiments are dealt in Section 4, and the results and their analysis are covered in Section 5. Finally, Section 6 provides an interpretation and summary of the findings with possibilities for the future work.

2. Related Works

A great deal of researches has been projected in image fraud detection and localization. This ranges from the traditional feature extraction algorithms to the latest deep learning approaches. In this section, various relevant efforts that use both conventional and deep learning-based strategies for identifying and localizing image forgeries are briefly outlined.

2.1 Conventional Forgery detection methods

The traditional methods included handcrafted feature extraction phase which is very complex and are limited to the detection of specific forgery types. A method of copy move forgery detection and classification with improved relevance vector machine is proposed by Rathore N. K et al. [10]. The method utilized biorthogonal wavelet transforms with Singular value decomposition-based feature extraction and classification is based on the existence of similar vectors. Image forgery detection involving noise analysis is carried out by Krawetz N [11] and Mahdian B et al. [12]. Usage of wavelet coefficients to address inconsistent noise in tampered images is outlined in the former. A method of noise level analysis involving segmentation of images with particular compression ratio and hence partitioning segments of homogeneous noise levels is performed in the latter case. Detection of varying noise levels may signify the tampering involved. Exploiting the features of the neighbouring pixels to approximate the camera filter array for the prediction of pixels are proposed by Ferrara P et al. [13] . A novel copy-move image forgery detection method using combined features of SIFT and LIOP followed by transition matching is proposed by Lin C et al. [14]. It is suggested to use picture segmentation for filtering in order to remove false matches. In order to identify duplicated locations, affine transformations between picture patches are finally obtained. The approaches that have been covered so far, either focus on any particular feature that is present in the image or use explicit feature extraction algorithms. However, this will only be beneficial in identifying a specific kind of forgery, and the process of explicitly extracting features is laborious and complex. This study, on the other hand, avoids human feature engineering by concentrating on a deep learning architecture that implicitly learns discriminative representations straight from data. Additionally, by using images of various forgery types for training, generalisation capability to forgery types is guaranteed.

2.2 Deep Learning Based Forgery Detection methods

The capability of Deep Learning models to automatically extract features from images was a turning point in the field of image forgery detection and localisation. With the advent of super computers and GPU, there was a visible shoot in deep learning-based researches in all applicable fields. A good deal of early studies using deep learning investigated several forms of forgeries, such as copy move, splicing, or their combinations. Two approaches for copy move forgery detection that use deep learning - one model by a custom architecture and another one model by transfer learning using VGG 16, are proposed by Rodriguez-Ortega Y.et al. [15]. In each case, the impact of the depth of the network is analysed in terms of performance metrics. A novel dual branch convolutional neural network for copy move forgery context is proposed to classify the images in to authentic and fake by Goel N et al.[16]. Different kernel sizes are employed in dual branches of convolutional networks and this helps in extracting multiscale features from the image. A fusion of deep learning models for copy move forgery detection was proposed by Krishnaraj N et al. [17]. Generative adversarial networks (GANs) and densely connected networks (DenseNet) are considered for fusion in a merger unit. Extreme learning classifier is used as classification layer and are optimised using artificial swarm algorithm. The combination of the complementary approaches in copy move and splicing detection techniques to boost the overall accuracy of fake image detection was proposed by Mohammed T. M et al.[18]. Copy move detection method is used as a prefiltering step and those images that are untampered are passed to a resampling deep learning-based detection architecture. The put forth method outperformed some of the state-of-the-art methods. A hybrid CNN-LSTM architecture to detect image forgery is proposed by Bappy J.H at al.[19]. The method focuses on the detection of boundary discrepancy between manipulated and non-manipulated regions. The proposed method can detect image manipulations including copy-move, removal, and splicing. Descriptive review of deep learning algorithms for cybersecurity applications is discussed by Dixit P et al.[20]. Guillaro F et al. present TruFor, a novel transformer-based fusion architecture[21]. The procedure generates both a pixel-level localisation map and an overall picture integrity score. The method also finds error prone areas in localisation predictions. A new network called ManTra-Net was developed by Wu. Y et al [22]. It is an integrated convolutional neural network made up of two sub-networks: one to identify local abnormalities between the features and another to extract features associated with manipulation traces. The proposed model was shown to be a robust, superior, and generalisable model in the application of forgery detection and localisation. A revolutionary deep neural architecture for image copy-move forgery detection is proposed by Wu Y et al [23]. It has a fusion module after a two-branch architecture. Both branches use visual artefacts to locate possible manipulation locations and visual similarities to locate copy-move regions. Most deep learning-based methods have significantly improved image forgery detection and localization by automatically learning hierarchical features. However, many existing models have architectural complexity and require large-scale labelled datasets for effective training. This work focuses on a deep learning architecture with reasonable layer depth, with reduced training images requirement without compromising performance towards forgery detection.

2.3 Transfer learning-based Forgery detection methods

Certain studies leverage the idea of transfer learning to lower the computing complexity of their methodology. An improved model using stacked autoencoders and Ensemble subspace discriminant classifier is proposed by Bibi S et al. [24]. The feature extraction is done by various CNN models like VGGNet and AlexNet and the best features are used for the classification. To reduce the long training times in the detection of Splicing datasets, a novel method was put forward by Qazi E.U.H et al. [25].The proposed architecture use ResNet50v2 as the base model, and the YOLO CNN weights for transfer learning. The proposed approach enabled to train the model with meaningful weights. A hybrid method that combines the advantages of residual network, UNET and dilated convolution is proposed by Ahmad Aminu A et al. [26]. The global image tampering evidences are captured and computational overhead are met by enlarging the receptive field size of convolutional kernels. Tracking Forgery and localization using color illumination, deep convolution neural network, and semantic segmentation are performed in the method recommended by Abhishek et al.[27]. VGG 16 CNN network is used as the transfer learning model for forgery detection in this approach. Further the images are subjected to semantic segmentation for localization. Multiple image splicing forgery detection using Mask R-CNN with MobileNet V1 as backbone model is proposed Kadam K.D et al.[28]. Analysis of the model with ResNet (51,101 and 151) is performed using MISD dataset and the proposed model proved the best in comparison. Fals-Unet, a convolutional neural network-based method is proposed to locate the forged regions by El Biach FZ et al. [29]. The model architecture of Fals-Unet architecture comprises of ResNet 50 based Encoder followed by decoder associating skip connections from the encoder. Localisation is efficiently done in terms of the metrics like F1-score, MCC, AUC, and Jaccard index. To compensate for limited training data, transfer learning-based methods make use of pretrained networks like VGG, ResNet, and Dense Net. Despite their effectiveness, these techniques inherit unnecessary complexity and are not optimised for forgery detection tasks. Furthermore, the sensitivity to subtle tampering artefacts may be limited due to the dependence on generic features learned from natural images. The suggested method, in contrast to conventional methods, is trained from scratch using task-specific objectives, allowing for greater flexibility to forgery detection without relying on pretrained models.

2.4 Forgery detection using encoder-decoder based methods

A novel deep learning system for identifying image forgeries based on double image compression is recommended by Ali S.S et al. [30]. The forged and unforged areas are enhanced in distinct ways upon recompression. Authentic region will be compressed twice - 1st in camera and 2nd during forgery. The suggested model is trained using the differences between the original and recompressed images. A two-level faster R-CNN network is proposed by Zhou P et al.[31]. In the first level features are extracted from the RGB stream of the input and the other leverages the noise inconsistency between real and fake images. These features are then fused to include the spatial co-occurrences of the two modalities. A new framework to improve forgery localization was put forth by Li H et al.[32]. In this approach two forensic approaches – statistical feature-based on detector and copy-move forgery detector are improved and integrated to obtain efficient localization of forgery. A novel architecture named HFSRNet capable of detecting multiple forgery types based on encoder decoder network is proposed by Chen H et al.[33]. LSTM with resampling features is adopted to capture traces for finding manipulations. Image manipulation detection by identifying the artefacts caused by tampering in image patches by utilizing LSTM and encoder-decoder architecture was proposed Bappy J.H et al.[34]. Resampling features of the images are also analysed to detect forgeries. A similar forgery detection technique utilizing skip connections in LSTM and encoder – decoder architecture is proposed by the Mazaheri G et al.[35]. The proposed network outperforms some of the existing techniques in localising the forgeries present in the images. A real time image forensic method was proposed by Yang B et al.[36]. The proposed methodology efficiently captures the artefacts caused due to manipulations in images by using multidomain learning. The manipulations can be automatically learned by multi domain learning convolutional network. A novel image forgery detection based on CNN framework considering images in full resolution is proposed by Marra F et al.[37]. The model includes gradient checkpointing along with training with limited memory resources. A new deep CNN based image forgery detection and localization incorporating multi-semantic CRF attention model is proposed by Rao y et al.[38]. Receptive fields attention maps are generated at different scales by four attention models with various semantics. In majority of these encoder-decoder techniques, deep feature extractors modified from segmentation or recognition networks, with significant layer depth are utilized. This contributes to unnecessary architectural complexity for binary forgery classification. The suggested method highlights a reduced-depth autoencoder, reusing reconstruction-trained encoder features for effective binary classification of image forgery detection.

A prevalent limitation shared by all these deep learning-based techniques is the intricacy of the models and prolonged training durations. There exist challenges in the usage of transfer learning models due to the large feature dimensions. In addition, most techniques targeted exclusively on identifying a particular kind of forgery. This led to the idea of developing a deep learning model that could ensure better performance with reasonable layer depth to extract all pertinent features from images, and evaluating the features contributing to forgery detection of all categories.

3. Proposed Framework

This research proposes an effective image forgery recognition approach that uses a classifier network in combination with reconstruction characteristics taken from an encoder-decoder architecture-based image reconstruction model. The classifier network, which has been trained to detect forgeries, receives the important characteristics that the feature extractor model has extracted from the image. Because of its significant dimensionality reduction capacity, the encoder-decoder architectural framework is used for feature extraction. The primary goal of the work is to design an architecture that maintains performance without penalising model complexity. The proposed work can be easily be executed resource constrained environments.

The input dimension of 256X256X3 is reduced to 64X64X256 using the feature extractor suggested in this study and the output dimension is 256X256X3. Since an image reconstruction model is framed, the encoder portion collects all the relevant features from the image for perfect reconstruction. The proposed framework is arrived from the ablation experiments conducted for image reconstruction. The experimental validation of the model's performance in contrast to models created by progressively increasing layers determines the model architecture. The feature extraction by the encoder that gave the best reconstruction of images and best loss convergence by the corresponding decoder are considered for further study. Three network architectures are constructed progressively by adding convolutional blocks one after the other, and their outputs are compared, to arrive at the suggested architecture. Framework 1 is a basic convolutional encoder that was developed with three convolutional blocks in the encoder side. Each convolutional block consists of 2D convolution[39] followed by Batch normalization [40] and ReLU [41] activation. Framework 2 was created by increasing the number of convolutional layers, normalisation, and activation function inside the block, to two, while maintaining recurrency. Lastly, Framework 3 was built by increasing the convolutional blocks, which have the same block structure as Framework 2, to five. Convolutional 2D transposition serves as the up-sampling layer of the decoder, which is followed by convolutional blocks with the same layers as the encoder side. The convolutional layers within the convolutional block use 3×3 kernels with stride 1 and same padding. The total parameter count of the 3 frameworks are 723,651, 1,685,315 and 27923523 respectively. All three variations are used, and the encoder's feature extraction that produced the best image reconstruction, accuracy, and loss convergence is considered for identifying forgeries. The details of the frameworks implemented and the details of the layers are detailed in Table 1.

Table 1. Details of the 3 frameworks implemented

Frameworks Implemented	Details of Convolutional blocks added in encoder	Details of Convolutional blocks added in decoder	Layers within each Convolutional block	Total number of layers
Framework 1	3 convolutional blocks and 2 max pooling layers	2 convolutional blocks and 2 transposed convolutional layers and 1 convolutional layer	One 2D Convolutional layer followed by Batch Normalization and ReLU	20 layers (11 layers in encoder and 9 layers in decoder)
Framework 2	3 convolutional blocks and 2 max pooling layers	2 convolutional blocks and 2 transposed convolutional layers and 1 convolutional layer	Two sets of 2D Convolutional layer followed by Batch Normalization and ReLU	35 layers (20 layers in encoder and 15 layers in decoder)
Framework 3	5 convolutional blocks and 4 max pooling layers	4 convolutional blocks and 4 transposed convolutional layers and 1 convolutional layer	Two sets of 2D Convolutional layer followed by Batch Normalization and ReLU	63 layers (34 layers in encoder and 29 layers in decoder)

The autoencoder model training is completed using the train set of image forgery dataset and the reconstructed images are verified using validation set. The predictions are validated using the test set. Following the implementation of the aforementioned models, the reconstructed images and loss convergences are observed and contrasted. The architectural framework with 3 convolutional blocks making 20 layers in the encoder side and 15 layers in the decoder was found to be the best architecture for feature extraction. The features of this model are used for classification of image forgeries. Since the encoder section of the framework 2 generated an optimal reconstruction of the image at the decoder side, the features that were obtained from it will encompass every significant feature. These features are then given to the classifier network. The classifier network consists of dense layers followed by sigmoid classifier owing to binary classification. The dimensionally reduced features are flattened and provided as input to the classifier network. The real and tampered images from the image forgery dataset are used to train the classifier network. The suggested model for detecting image forgeries is displayed in Fig 1.

3.1 Architectural Framework Details

The encoder section consists of 20 layers and decoder consists of 15 layers resulting in a 35-layer encoder decoder architecture. Maxpooling layers and convolutional blocks make up the encoder part. Batches of 2D convolution with appropriate filters, batch normalisation, and the ReLU activation function make up the convolutional block. The 3X3 kernel dimensions are used, and the number of filters is increased from 64 after each convolutional block. An algorithmic technique called batch normalisation speeds up and stabilises neural network learning. It involves using the statistical moments of the current batch to normalise the activation vectors. A one-degree equation known as the ReLU activation function creates zeros for all negative inputs and outputs all positive inputs. In summary, if the input is positive, it outputs the input as such; if it is negative, it outputs zeroes. After the convolutional block, max pooling layer[42] follows which chooses only the relevant information thereby, reducing the feature size and provides it to the next succeeding convolutional block. The feature information is supplied into the third convolutional block after the first two batches of convolutional blocks and max pooling layers The same series of operations take place in the third convolutional block and its output, which is the encoded output, is finally fed to the decoder section. Fig 2 illustrates the blocks of the encoder decoder architecture of the recommended Framework 2 architecture. After the initial convolution process of the model, the result is provided by

$$C_1(k) = \Sigma_{k=1}^{64} \sum_{i,j=1}^{256} \left(\sum_{d=1}^{3} x(d,i,j) * f_k(d,i,j) \right) + \emptyset_k \tag{1}$$

Where x denotes the image pixel, i and j denote the image resolution, d is the width indicating the channels, k is the filter index, f_k the convolution kernel of k^{th}filter, and $\emptyset_k$ is the bias of the k^{th} filter. The output of the convolution is subjected to batch normalisation with batch size 32. The output of batch normalisation is represented by

$$B_1 = \gamma_{C1} \left[\frac{x_{C1kl} - \frac{1}{n/\beta} \Sigma_{i=1}^{\beta} \Sigma_{j=1}^{n/\beta} x_{C1ijk}}{\sqrt{\Sigma_{i=1}^{\beta} \Sigma_{j=1}^{n/\beta} \sigma_{C1ijk}^2 + \epsilon}} \right]_{k=1}^{k=\beta}{}_{l=1}^{l=n/\beta} + \beta_{C1} \tag{2}$$

Where γ_c and β_c are learnable scale and shift parameter respectively. ε is a constant having the value 0.00001 and σ_c^2 represents the variance. The number of images is denoted by n and β is the batch size. x_{C1kl} denotes the first element in the image matrix after the first convolutional block indicated by c_1. The value of i indicates the row wise elements of the batches and the value of j indicates the column wise increments of the elements across the total batches. Following batch normalization is the activation layer. Here we use ReLU activation layer, followed by Max pooling operation.

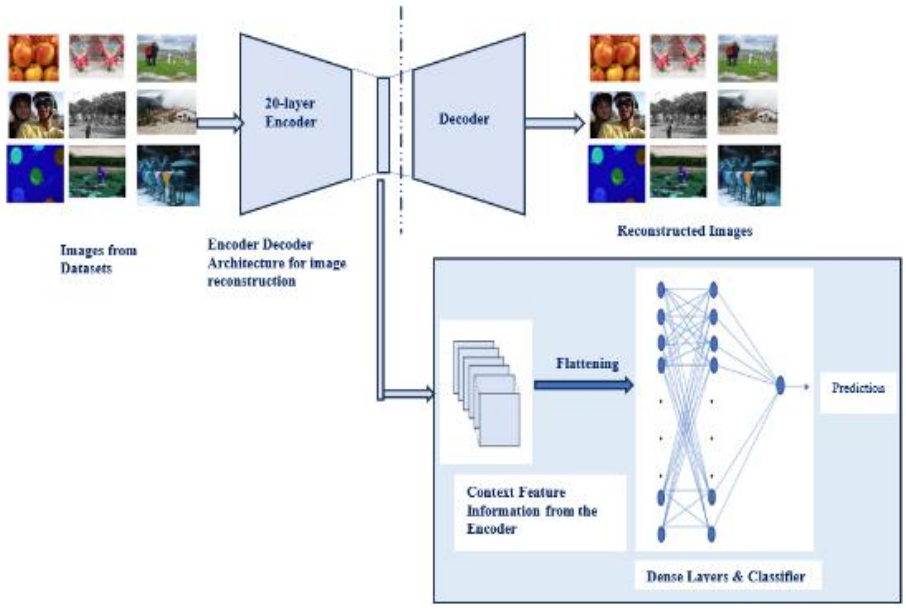

Fig. 1. Proposed Model

The output of the first convolutional block is represented by

$$CB_1 = \max[0, \gamma_{C2}\left[\frac{x_{C2kl} - \frac{1}{n/\beta}\sum_{i=1}^{\beta}\sum_{j=1}^{n/\beta} x_{C2ijk}}{\sqrt{\sum_{i=1}^{\beta}\sum_{j=1}^{n/\beta}\sigma_{C2ijk}^2 + \epsilon}}\right]_{l=1}^{l=n/\beta}{}_{k=1}^{k=\beta} + \beta_{C2}] \quad (3)$$

The above equation is the output of the first level of the convolution block and is the input to the first max pooling layer. The suffix c_2 is used since the convolutional block consists of two 2D convolutions. The model maintains the recurrency for the next convolutional block. After max pooling layer, the output of the second convolutional block is fed to the third convolutional block. The final outputs of Convolutional layer and Batch Normalisation in the third block, and the output of the final convolutional block in the encoder side is represented by the following equation.

$$CB_3 = \max[0, \gamma_{C6}\left[\frac{x_{C6k,l} - \frac{1}{n/\beta}\sum_{i=1}^{\beta}\sum_{j=1}^{n/\beta} x_{C6kij}}{\sqrt{\sum_{i=1}^{\beta}\sum_{j=1}^{n/\beta}\sigma_{C6ijk}^2 + \epsilon}}\right]_{l=1}^{l=n/\beta}{}_{k=1}^{k=\beta} + \beta_{C6}] \quad (4)$$

Decoder consists of convolution 2D transpose layers which performs up sampling, and convolutional 2D block followed by 4 convolutional blocks and a final 2d convolution layer. Transposed convolution is same as that of deconvolution. It carries out a regular convolution but reverts its spatial transformation. Finally, the images are reconstructed from the extracted feature information. The decoder is governed by the following equations. The first up sampling by the transposed convolution is given by

$$C_{T1}(k) = \Sigma_{k=1}^{128} \sum_{i,j=1}^{128} \sum_{d=1}^{256} (x_{CB3}^{T}(d.i.j) * f_k(d,i,j)) + \emptyset_k \quad (5)$$

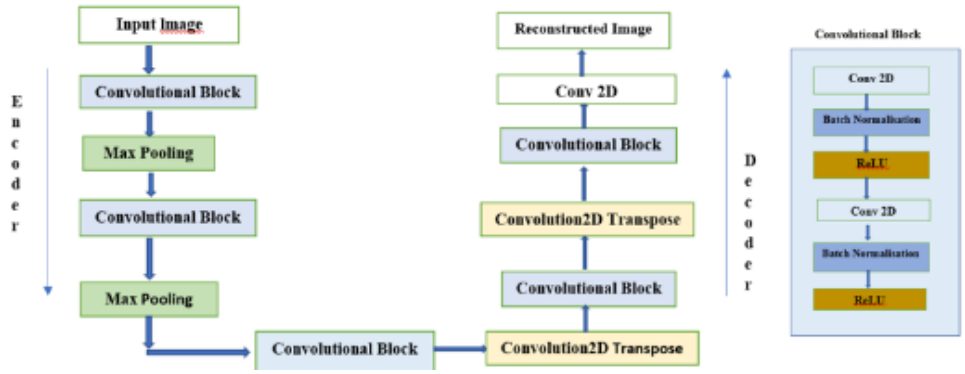

Fig. 2. Feature Extraction model for Framework 2

Where x_{CB3}^{T} corresponds to the elements of the spatially transformed output of convolutional block 3. The $d.i.j$ and k are the same notations as indicated in (1). The output of the transposed convolution is given to the convolution block 4 with 2 recurrent convolutions and output is represented by

$$CB_4 = \max[0, \gamma_{C8}\left[\frac{x_{C8k,l}-\frac{1}{n/\beta}\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}x_{C8kij}}{\sqrt{\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}\sigma_{C8ijk}^{2}+\epsilon}}\right]_{l=1}^{l=n/\beta}{}_{k=1}^{k=\beta} + \beta_{C8}] \quad (6)$$

x_{C8} corresponds to the convoluted output of the 8th convolutional layer. Similarly, the output of the final up sampling layer, Convolutional block and the final output is represented by the following equations

$$C_{T2}(k) = \Sigma_{k=1}^{64}\ \Sigma_{i,j=1}^{256}\ \Sigma_{d=1}^{128}(CB_4^{T}\ (d,i,j) * f_k\ (d,i,j)) +\ \emptyset_k \quad (7)$$

$$CB_5 = \max[0, \gamma_{C10}\left[\frac{xC_{9k,l}-\frac{1}{n/\beta}\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}x_{C9kij}}{\sqrt{\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}\sigma_{C9ijk}^{2}+\epsilon}}\right]_{l=1}^{l=n/\beta}{}_{k=1}^{k=\beta} + \beta_{C9}] \quad (8)$$

$$C_{11}(k) = \Sigma_{k=1}^{3}\ \Sigma_{i,j=1}^{256}\ \Sigma_{d=1}^{64}(C_7\ (d,i,j) * f_k\ (d,i,j)) +\ \emptyset_k \quad (9)$$

The output of the decoder will be reconstructed images of output shape 256x256x3. To form the complete model the features from the encoder block is vectorised and the resultant is given to dense network with fully connected layers which applies linear transformations, followed by the classifier. The output from the classifier indicates the image type - authentic or fake. The prediction using the test dataset can be carried out using the model evaluate function. The following equations represent the output layers mentioned before. $\bar{F}$ indicates the vector after flattening operation, Y' indicates the output of the fully connected layers after flattening, represented as function $D(\bar{F})$ and $\hat{Y}$ indicates the predicted output followed by sigmoid classifier represented by σ.

$$Y' = D\{\bar{F}[\ \max[0, \gamma_{C6}\left[\frac{x_{C6k,l}-\frac{1}{n/\beta}\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}x_{C6kij}}{\sqrt{\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}\sigma_{C6ijk}^{2}+\epsilon}}\right]_{l=1}^{l=n/\beta}{}_{k=1}^{k=\beta} + \beta_{C6}]\} \quad (10)$$

$$\hat{Y} = \sigma(D\{\bar{F}[\ \max[0, \gamma_{C6}\left[\frac{x_{C6k,l}-\frac{1}{n/\beta}\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}x_{C6kij}}{\sqrt{\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}\sigma_{C6ijk}^{2}+\epsilon}}\right]_{l=1}^{l=n/\beta}{}_{k=1}^{k=\beta} + \beta_{C6}]\}) \quad (11)$$

The fully connected layers have weights and bias as its parameters and for the back propagation change in error $(\frac{\partial E}{\partial W})$ with respect to the weights are to be calculated. Since error is not directly dependent on the weights, chain rule must be applied to find it. The discriminator loss function, L is based on the equation given by,

$$L = -\Sigma_{i=1}^{N} Y_i \log\{\sigma(D\{\bar{F}[\max[0, \gamma_{C6}\left[\frac{x_{C6k,l}-\frac{1}{n/\beta}\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}x_{C6kij}}{\sqrt{\Sigma_{i=1}^{\beta}\Sigma_{j=1}^{n/\beta}\sigma_{C6ijk}^{2}+\epsilon}}\right]_{l=1,k=1}^{l=n/\beta,k=\beta} + \beta_{C6}]\})\} \quad (12)$$

Where Y_i is the actual value and $\hat{Y}$ is the predicted value. The loss function is minimised to obtain the maximum accuracy of the model.

4. Experiments

The experiments were conducted as per the following stages. The architectures were implemented in Keras[43] with Tensorflow backend using Python 3 programming language and the simulations are run on Google Colab for smaller datasets. The training is done using four image forgery datasets. Simulations with larger datasets are done using Google Colab Pro with GPU Runtime and high RAM configuration.

In the first stage, as part of ablation studies, three models utilizing convolutional blocks in encoder-decoder architecture were implemented to extract the best image reconstruction features. All the models were run for 100 epochs using Coverage Dataset with Adam optimiser with default learning rate with batch size 32, and the loss function used was mean squared error[44]. The dataset was split in three parts (train, validate, and test samples) in an 80:10:10 ratio. Accuracy and loss plots were used to assess the training and validation outcomes, and reconstructed images for each case were also examined. After comparing the outcomes, the best reconstruction feature model was chosen for further study.

In the second stage, the features from the best image reconstruction architecture were combined with the classifier network. The best reconstruction feature is vectorised and is provided as input to fully connected layers, followed by the classifier. Experiments were done with different classifiers like Sigmoid [45], Softmax [46] and SVM [47].. The complete model is again trained for 25 epochs with Adam optimiser with learning rate set to 10^{-3} with the same batch size and binary cross entropy loss function. The results are predicted using the test data set. Performance metrics like Accuracy, Precision, F1 score and Recall were calculated and analysed for different datasets.

Comparing the performance of the suggested approach against some of the state-of-the-art and transfer learning models is the final stage. For transfer learning VGG19, ResNet50 and MobileNet is used as the base model in the encoder section. Comparison of the suggested model with some of the existing advanced methods based on the architecture and the detection performance are also carried out.

4.1 Statistical Validation

CNN models are used as the basic model on the encoder side of the transfer learning architecture, which is compared to the suggested method for forgery classification. The encoder component of the model design uses some of the best classification CNN models currently in use, such as VGG19, ResNet 50, and MobileNet, as the foundation model. The decoder is built utilising the transposed convolution layers and convolutional blocks in accordance with the pretrained layers of the current models, which are employed in the encoder. Dense layers and classifiers are integrated with this encoder, and is again trained using both authentic and fake images from the datasets. All experiments were repeated using three fixed random seeds (42, 123, and 2023) to address the stochastic nature of deep learning training. The models were trained independently using the same training, validation, and testing splits for every seed. Performance metrics were calculated on the test set, and the final results are provided as mean ± standard deviation across the seeds and the overall outcomes are contrasted with the suggested method. A paired t-test was performed on the per-seed performance scores to see whether the observed performance differences between the suggested model and baseline architectures (VGG19, ResNet50, and MobileNet) are statistically significant. The metric values acquired under identical random seeds were considered paired observations for each dataset. A two-tailed test with a significance level of $\alpha = 0.05$ is conducted, since all models were evaluated on the same dataset splits under identical random seeds, ensuring a fair and controlled comparison. Statistical testing was undertaken separately for each dataset to keep out cross-dataset bias and capture dataset-specific performance variability.

4.2 Datasets

The datasets used for training and evaluation are Coverage[48], CG-1050[49], Casia1 and MICC F 220. The descriptions of datasets are specifics are listed in Table 2. The images are resized to 256 X 256 and then divided by 255 to normalise them, so they fall between 0 and 1. The image sets are considered in mini batches of size 32 for the training and validation phases.

Table 2. Datasets

Dataset	Total Number of Images	Image Size	Types of forgery involved
Coverage	200	400 x 486	Copy move
MICC F220	220	800 x 600	Copy move
CG-1050	1050	4608 x 3456	Copy move, cut paste, retouching, colorizing
CASIA-1	1721	348 x 256	Splicing

4.3 Performance evaluation metrics

The capability and the efficacy of the selected model in classifying image manipulation is scrutinized using different performance parameters such as Accuracy, Precision, Recall and F1 score. The positive prediction values denote Precision. Out of the total actual predictions of the model, how many of the predictions are correctly made, gives the precision. Sensitivity or the true positive rate of the predictions carried out is Recall. The number of right predictions to the total correct values in the problem gives Recall. F1 score is the metrics F_β with β value equal to 1. The value of β is normally taken as 1 in the analysis of application in which both false positive and false negative scores have equal impacts.

5. Results and Discussions

The result analysis is done with respect to the 3 experimental stages.

5.1 Ablation Study

For the initial experimental phase, three architectural frameworks with different architectural complexities are created by successively adding convolutional blocks. Table 1 already contains a thorough description of the blocks that were added. The images from the dataset Coverage split in to train,validate and test sets in the ratio 80:10:10 are considered for the ablation study. All the 3 architectures are trained using the images from the train set, validated using validate set and tested using the test set. All experiments were carried out using strictly disjoint training, validation, and test splits to prevent data leaking, making sure that no test images were used for parameter tuning or model training. The visualisation of the reconstructed images, accuracy, and loss convergence are used to determine the best model. The reconstructed images of the 3 frameworks are shown in Fig.3 (a), (b) and (c) respectively. It shows the reconstructed images of all the frameworks implemented, using the test data of the Coverage dataset. All the models are trained for 100 epochs. The training and validation accuracies and losses are also compared. The training and validation accuracies and losses of all the 3 models are represented graphically in Fig.4(a), (b) and (c). The graph shows that the losses converge fast to minimum for Framework 2 in comparison with the other 2 frameworks. It is inferred that the second framework converges to the minimum loss point around 85 epochs. It converges to loss of 0.05 in 15 epochs, whereas 1 and 3 frameworks converge around 25 and 30 epochs respectively. Model 2 demonstrates faster and has more stable convergence compared to other architectures, as observed from training loss, and accuracy curves. The accuracy improves with number of epochs as well. All the results shows that Framework 2 offers good reconstruction with better loss convergence and accuracy.

5.1.1 Heatmap and Grad-CAM Visualizations of reconstruction

The efficiency of Framework 2 in extracting all pertinent information of the image is visually understood by the heatmap and Grad-CAM visualizations. The features extracted by the model to reconstruct the image at various stages of convolutions in the model is obtained and displayed in Fig.5. Considering the average values of gradients flowing from the first layer till the last convolutional layers, activation maps of features are also generated. The heatmaps and Grad-CAM of the features at various stages of reconstruction model along with the original image is represented in Fig 5[a-c]. A heatmap is a type of data visualization where values from a matrix or 2D grid are represented using colour gradients.

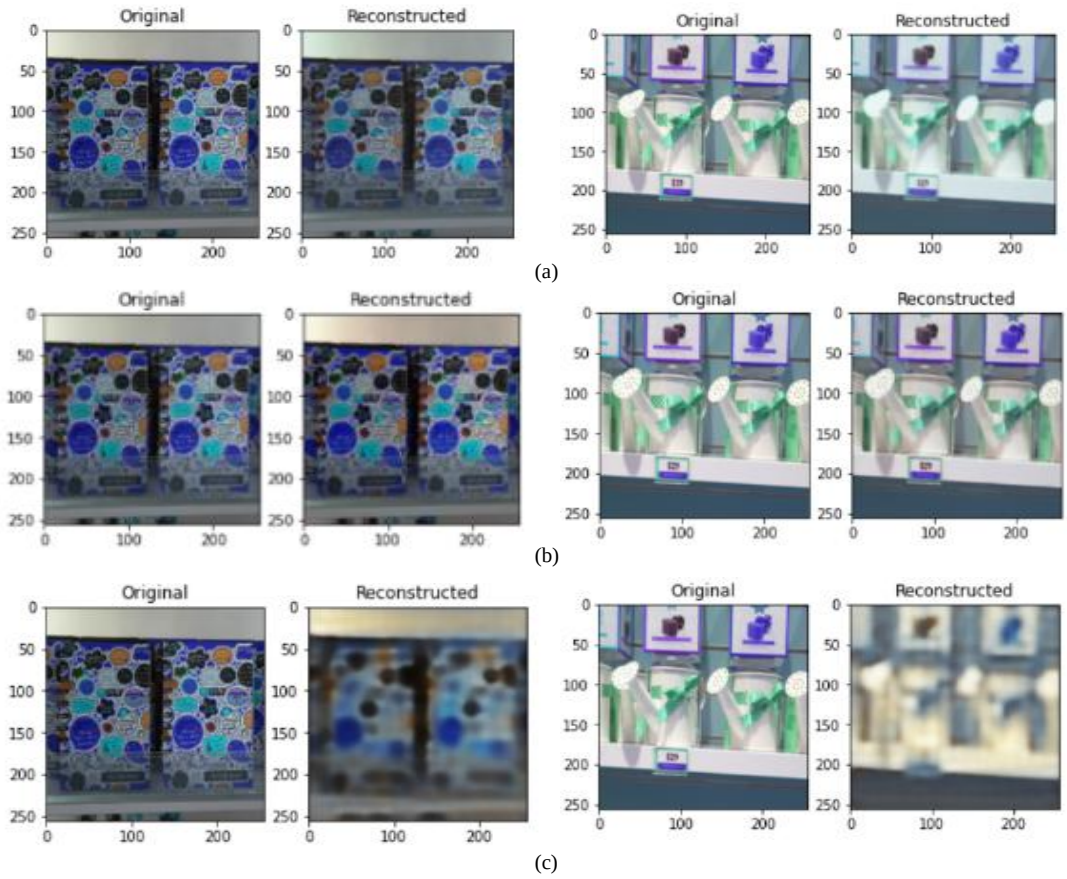

Fig. 3. Original and Reconstructed images of (a) Framework 1 (b) Framework 2 (c) Framework 3

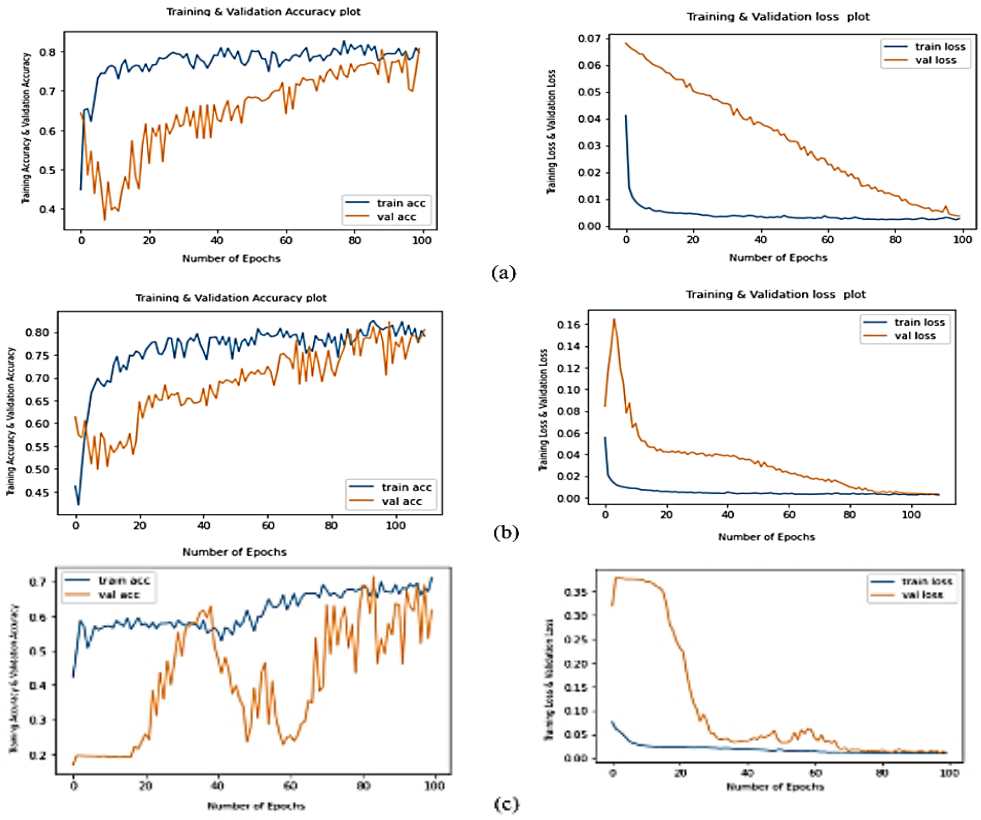

Fig. 4. Accuracy and Loss plots of (a) Framework 1 (b) Framework 2 (c) Framework 3

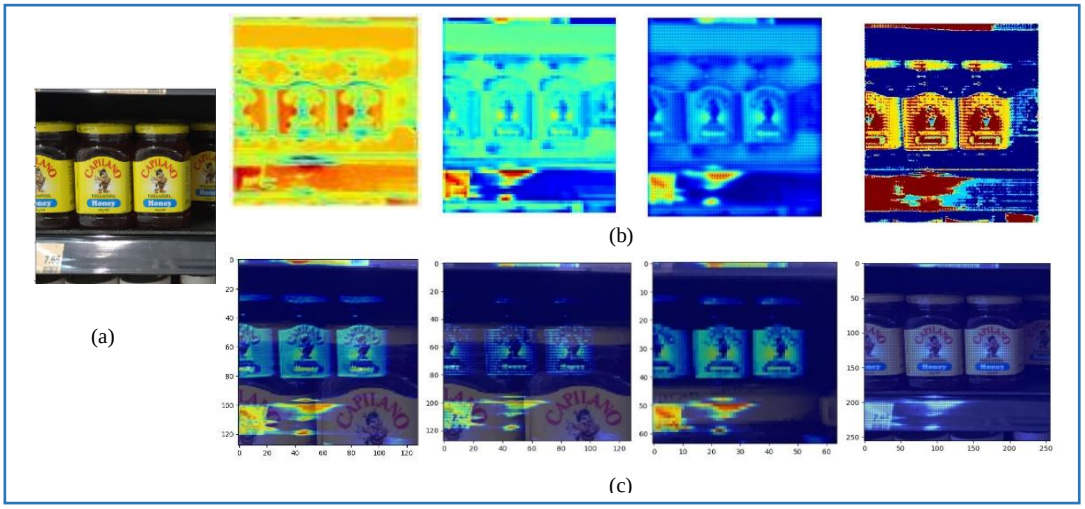

Fig. 5. Qualitative visualization of recommended reconstruction model behaviour at various stages of convolutions (a) Original Images (b) Heatmaps representing feature activation responses extracted from intermediate convolutional layers of the autoencoder, highlighting regions that strongly influence the reconstruction process (c) Grad-CAM Visualizations of spatial feature importance with respect to the reconstruction output, providing interpretability of the learned reconstruction features.

Heatmaps speed up comprehension by encoding numerical values as colours, thereby facilitating data interpretation. Hence, they are considered as an effective tool for improving interpretability, which promotes more insightful thinking and improved decision-making. Heatmaps generated at the convolutional layers 2, 5, 7 and 10 respectively are displayed in Fig 5(b). Grad-CAM is a useful tool for improving the transparency and interpretability of deep learning models, strengthening their reliability and effectiveness in a variety of applications. It enables viewers to figure out which aspects of an image are most significant for a model's forecast. The Grad-CAM visualizations of intermediate convolutional layers in Framework 2 is shown in fig 5(c). The heatmaps and Grad-CAM visualisations that are displayed are obtained from the intermediate layers of the reconstruction network and are meant to offer qualitative understanding of the reconstruction behaviour. It highlights spatial locations that play a significant role in the reconstruction process and higher responses in forged images due to structural or texture inconsistencies. As the suggested framework concentrates on reconstruction and image-level classification, these visualisations are not optimised for explicit forgery localisation.

5.2 Performance of the Proposed framework in classifying Image forgeries

The output from the 20 layers encoder section which gave best reconstruction is integrated with the fully connected dense layers and the classifier, to form the complete model. The resulting model is trained for 25 epochs using different datasets with Adam optimizer, learning rate 10^{-3} and categorical cross entropy loss function. The results are predicted using the test data set. The model is trained using the datasets Coverage, MICC F 220, CG 1050 and Casia 1. The dataset was split in to train, validate, and test samples in 80:10:10 ratio. The reconstructed images by the model using datasets and their corresponding accuracy and loss plots of training and validation are shown in the Fig.6 (a), (b), (c) and (d). Experiments were done with different classifiers like Sigmoid, Softmax and SVM. Results show that different datasets behave differently to different classifiers. The performance metrics for the different classifiers are obtained and tabulated in Table 3. Results show that for Coverage dataset, SVM classifier works better than sigmoid and softmax. For MICCF 220 Sigmoid works better than the others and for CG1050 Sigmoid and Softmax classifiers perform well than SVM. Result analysis show that the datasets Coverage and Casia 1 have better performance with SVM classifier and CG 1050 and MICC F 220 have better performance with Sigmoid classifier.

5.3 Comparison with Transfer Learning–Based CNN Encoders

In this experimental phase the performance of the recommended framework is compared with transfer learning model using VGG19, ResNet50 and MobileNet architectures as the base models in encoder section. The encoder section is implemented using the pretrained weights of VGG19, ResNet 50 and MobileNet architecture. The resulting models are trained for 100 epochs using and the reconstructed images are compared with that of the reconstruction model. Reconstructed images of the CNN encoder-decoder model using VGG19, ResNet 50, MobileNet and the proposed framework is shown in Fig.7(a-d). Fig.7(a) and 7(b) shows the reconstructed images of the recommended reconstruction model and the existing CNN models using datasets Coverage and CG1050. Fig. 7(c) and 7(d) shows the reconstructed images of the proposed framework and the existing CNN models using datasets MICCF 220 and Casia 1. The figure shows the reconstructed image along with the original image. From the visuals, it is evident that the recommended framework reconstructs images better than the transfer learning models. The features extracted from the existing CNN models are utilized for classification and output layers consisting of fully connected layers and Sigmoid classifier are added to each transfer learning model encoder, and their corresponding prediction of test images are evaluated. The

performance of the transfer learning model is compared with the performance of the proposed approach in terms of the metrics such as Accuracy, Precision, Recall and F1 score. Three random seeds were used to repeat each experiment, and the results are presented as mean ± standard deviation and the values are tabulated in Table 4.

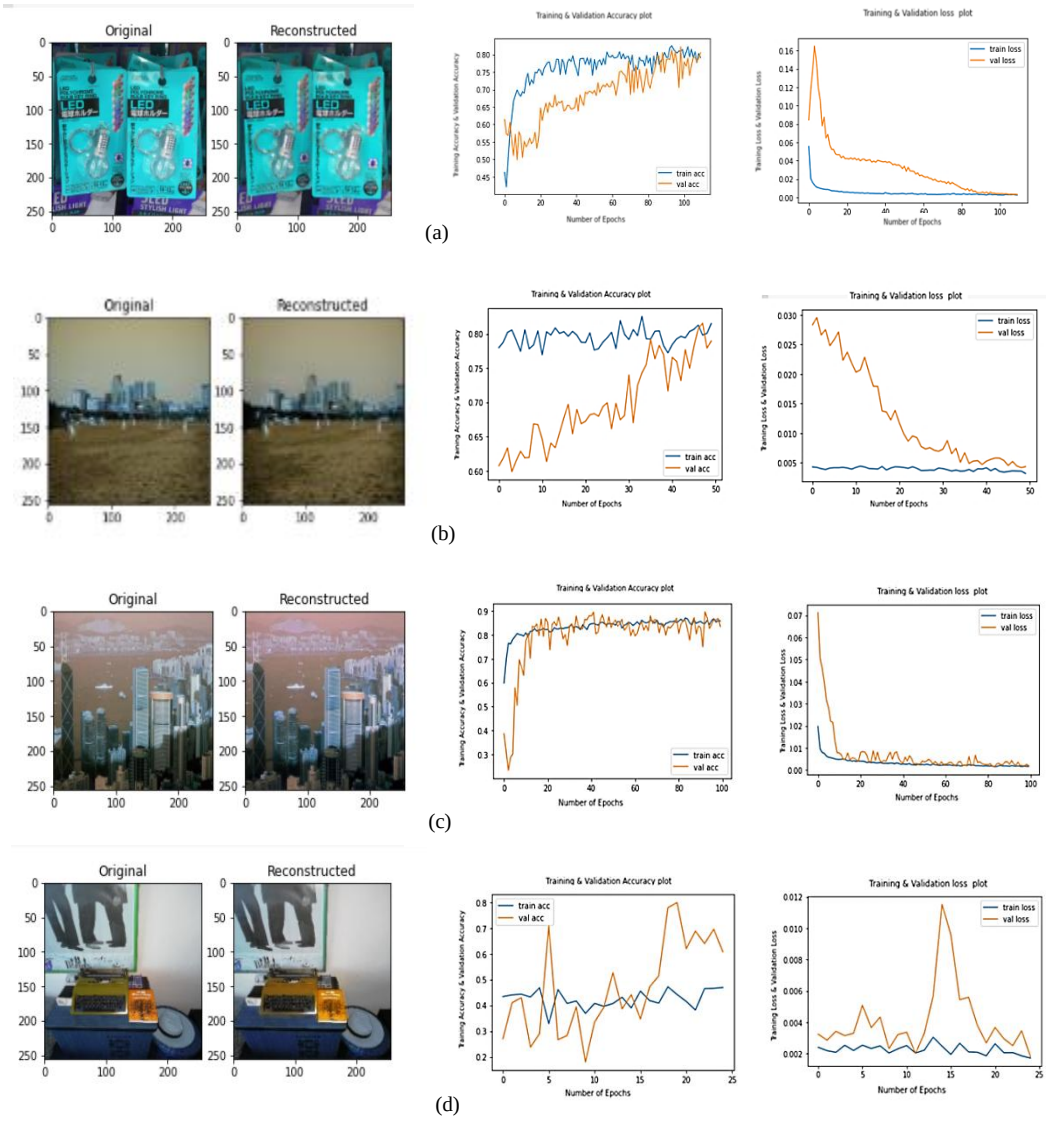

Fig. 6. Original Input image & Reconstructed Image and Accuracy and Loss plots of best reconstruction Model using datasets (a) Coverage (b) MICC F220 (c) CG-1050 dataset (d) Casia 1 Dataset

Table 3. Performance metrics of the best reconstruction model added with different classifiers

Datasets	Sigmoid				Softmax				SVM			
	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1
Coverage	0.2	0.28	0.23	0.2	0.05	0.06	0.05	0.05	0.5	0.58	0.5	0.52
MICC F220	0.91	0.93	0.91	0.91	0.77	0.88	0.77	0.79	0.77	0.81	0.77	0.78
CG-1050	0.88	1	0.88	0.94	0.88	1	0.88	0.94	0.49	0.6	0.49	0.41
Casia-1	0.43	0.89	0.43	0.58	0.48	0.87	0.48	0.59	0.48	0.87	0.48	0.59

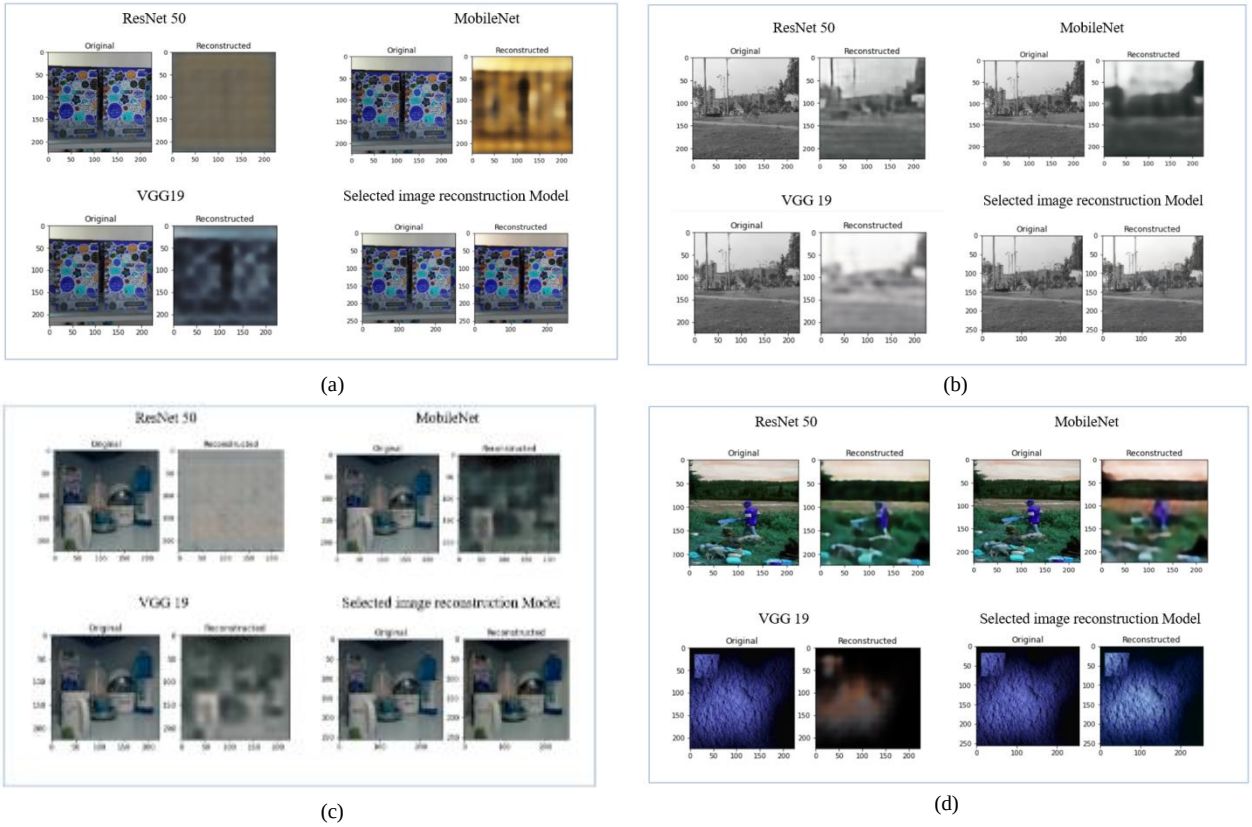

Fig. 7. Original and Reconstructed images of CNN models as encoder and Proposed framework using Datasets (a) Coverage (b) CG1050 (c) MICCF 220 (d) Casia I

Table 4. Classification performance (mean ± standard deviation) over three random seeds

Models	Datasets	Accuracy	Precision	Recall	F1_score
VGG 19	Coverage	0.2667 ±0.0850	0.0769 ±0.1088	0.1000 ±0.1414	0.0870 ±0.1230
	MICC F220	0.7424 ±0.1714	0.6905 ±0.1347	0.6970 ±0.4285	0.6379 ±0.3423
	CG-1050	0.8939 ±0.0107	0.8932 ±0.0097	1.0000 ±0.000	0.9436 ±0.0054
	CASIA-1	0.4143 ±0.0191	0.4153 ±0.0604	0.4014 ±0.2510	0.3726 ±0.1916
ResNet 50	Coverage	0.2000 ±0.2121	0.2273 ±0.1928	0.4000 ±0.4243	0.2857 ±0.2694
	MICC F220	0.8902 ±0.0054	0.8898 ±0.0048	1.0000 ±0.000	0.9417 ±0.0027
	CG-1050	0.9091 ±0.000	0.8462 ±0.000	1.0000 ±0.000	0.9167 ±0.000
	CASIA-1	0.3545 ±0.0926	0.3791 ±0.0942	0.4767 ±0.2940	0.4097 ±0.1706
MobileNet	Coverage	0.0833 ±0.0236	0.1111 ±0.0786	0.1333 ±0.0943	0.1212 ±0.0857
	MICC F220	0.9091 ±0.0000	0.8462 ±0.0000	1.0000 ±0.0000	0.9167 ±0.0000
	CG-1050	0.8409 ±0.0335	0.8839 ±0.0053	0.9444 ±0.0396	0.9128 ±0.0197
	CASIA-1	0.4759 ±0.0491	0.5026 ±0.0285	0.6918 ±0.2181	0.5733 ±0.0910
Proposed Model	Coverage	0.08 ±0.0471	0.1375 ±0.0659	0.1667 ±0.0943	0.1504 ±0.0781
	MICC F220	0.8182 ±0.643	0.8154 ±0.0218	0.8182 ±0.1286	0.8135 ±0.0730
	CG-1050	0.7803 ±0.126	0.8808 ±0.0120	0.8675 ±0.1515	0.8679 ±0.0873
	CASIA-1	0.4393 ±0.0649	0.4333 ±0.1096	0.5914 ±0.3374	0.4778 ±0.2157

Table 5. Dataset-wise paired two-tailed t-test p-values (α = 0.05) comparing the proposed method against baseline models.

Dataset	Metrics	VGG19	ResNet 50	MobileNet
Coverage	Accuracy	0.0927	0.4227	1.0000
	Precision	0.6504	0.4227	0.8203
	F1 score	0.6600	0.4227	0.6648
MICCF220	Accuracy	0.5490	0.1835	0.1835
	Precision	0.2937	0.1835	0.1835
	F1 score	0.5110	0.1835	0.1835
CG 1050	Accuracy	0.3197	0.3582	0.4945
	Precision	0.1747	0.5258	0.5763
	F1 score	0.3346	0.3659	0.4811
Casia 1	Accuracy	0.6670	0.5210	0.4971
	Precision	0.8893	0.7428	0.4005
	F1 score	0.7361	0.0925	0.5701

Statistical significance testing was conducted for accuracy, precision, and F1-score, which are the most relevant metrics for forgery classification performance evaluation. The dataset-wise p-values from a paired two-tailed t-test ($\alpha = 0.05$) comparing the suggested approach with transfer learning-based baselines (VGG19, ResNet50, and MobileNet) for accuracy, precision, and F1-score metrics are shown in Table 5. Although the proposed model shows competitive mean performance across all datasets, the observed differences are not statistically significant at the selected significance level. This result can be attributed to the minimal number of independent experimental runs and restricted dataset sizes, which lower the statistical power of hypothesis testing. However, the analysis shows evidence of consistent and stable performance of the suggested model across various datasets and performance criteria.

5.4 Comparison with State-of-the-art techniques

The comparison of the proposed model with some of the existing state-of-the-art techniques based on the architecture and performance parameters are conducted. Table 6 specifies the depth and the details of the layers of the models used in works [7, 35, 38] against the proposed model. The number of images used for training the models are also included. The information from the table makes it evident that the suggested model has reduced architectural complexity and less training data requirement unlike the other models. With this reduced architectural complexity and less training data, the model is capable of producing best results which is detailed by the performance comparison with the above-mentioned state-of-the-art works and with some conventional methods as indicated in Table 7 and 8. The suggested framework and typical state-of-the-art (SOTA) methods documented in the literature using Coverage and Casia 1 datasets are contrasted in Table 7 and 8. It projects the extent to which the proposed model performs, with the conventional methods of feature extraction using wavelet coefficients [11], noise inconsistencies [12], artifacts from the Camera Filter array [13], and deep learning-based methods [31, 33, 35] using the Coverage and Casia 1 dataset. Tables 7 and 8 have missing entries since a number of previous studies only include a portion of assessment measures (such as F1-score or precision). Rather than experimental omission in the current work, these missing values are a reflection of reporting methods in the original investigations. The performance of the proposed method is presented as mean ± standard deviation over several experimental runs, whereas the values for current methods are directly taken from their respective papers. Since different methods report different evaluation metrics and are trained under different experimental setups, comparisons are not strictly equivalent. Table 7 shows the comparison of F1 scores and precision experimented using Casia 1 dataset with the models indicated by the reference number. A comparison of the proposed methodology against the values mentioned in [33] is summarized in the Table 8 and the best values are highlighted. The stated values are acquired with varying dataset sizes and training setups. Instead of being a straight head-to-head comparison, the table functions as a qualitative standard. The suggested model achieves competitive F1-score and precision even though it was trained on a much smaller dataset (2,553 images) than numerous deep architectures developed on tens of thousands of samples as indicated in Table 6. This demonstrates its computing effectiveness and capacity for generalisation in situations with limited data. Table 8 shows that the suggested approach possesses dataset-dependent performance. It out performs a number of current methods with a competitive F1-score (0.4778 ± 0.2157) on the CASIA-1 dataset. On the other hand, for Coverage dataset, HFSRNet outperforms the suggested model with a higher F1-score of 0.624. This discrepancy can be explained by the fact that HFSRNet has a deeper architecture and a much bigger training set, whereas the suggested technique uses a lightweight architecture and a small number of samples. This indicates high sensitivity of the proposed method to training initialization and data sampling on the Coverage dataset, which contains limited and homogeneous copy–move forgeries. In contrast, the proposed method demonstrates more stable and competitive performance on the CASIA-1 dataset (0.4778 ± 0.2157), suggesting better generalization to diverse tampering patterns.

Table 6. Comparison of Architectural Complexity of the Proposed model with the State-of-the art models

Models	Total Number of layers	Layer Details	Number of images used for Training the model
[7]	68	20 Convolutional layers, 5 Maxpooling, 13 Batch normalisation, 15 ReLU activations, 2 dropout, 5 Deconvolution, 6 Cropping layers and 2 softmax activation layers	5123
[33]	42	1 transformation layer, 2 stacked LSTM layers, 17 Convolutional layers, 14 ReLU activations,4 Maxpooling and 4 Deconvolutions	65000
[34]	44	2 stacked LSTM layers, 15 Convolutional layers, 9 Maxpooling,12 ReLU activations, 3 Batch Normalization and 3 Up sampling layers.	65000
[35]	41	2 stacked LSTM layers, 18 Convolutional layers, 5 Maxpooling, 7 ReLU activations, 7 Batch Normalization and 2 Up sampling layers.	12617
[38]	84	75 Convolutional layers, 4 Attention layers, 4 Atrous Convolutional layers, 1 Atrous Spatial Pyramid Pooling layer	14000
Proposed	**35**	11 Convolutional layers, 10 Batch normalization and ReLU activation layers, 2 Maxpooling and 2 Transposed Convolutional layers	**2553**

Table 7. Comparison of F1 scores and Precision obtained for the selected model using Casia 1 dataset, with the state-of-the-art techniques.

References	F1 score	Precision
H.Chen et al [33]	0.467†	/
Salloum et al. [7]	0.54†	/
P.Zhou et al. [27]	0.408†	/
Rao et al. [38]	/	**0.61**†
Proposed	**0.4778 ±0.2157**	0.4333 ±0.1096

Note: best values are in bold. '/' conveys that the respective metrics are not included in the literature and the best scores are highlighted. Results marked with † are directly reported from the corresponding original publications and were not re-evaluated in this study.

Table 8. Comparison of F1 scores obtained for the proposed approach using standard dataset like Coverage and Casia 1 with the state-of-the-art techniques.

State-of-art-techniques	Coverage	Casia 1
ELA [11]	0.222†	0.214†
NOI1 [12]	0.269†	0.263†
CFA1 [13]	0.190†	0.207†
RGB-N [31]	0.437†	0.408†
HFSRNet [33]	**0.624**†	0.467†
LSTM-EnDec [34]	/	0.391†
LSTM-EnDec-Skip [35]	/	0.432†
Proposed	0.1504 ±0.0781	**0.4778 ±0.2157**

Note: best values are in bold. '/' conveys that the respective metrics are not included in the literature and the best scores are highlighted. Results marked with † are directly reported from the corresponding original publications and were not re-evaluated in this study.

6. Conclusion

In this paper a straightforward and effectual approach for image forgery detection by utilizing image reconstruction features with classifier network is proposed. The image reconstruction feature extraction model is selected by experimental evaluation of performance with the models one step ahead and behind in terms of architectural complexity. The comparison was based on the image reconstruction, training and validation accuracies, and losses. Moreover, the feature extraction is interpreted using heatmaps and Grad-CAM. The model performance is evaluated using the datasets like Coverage, MICC F 220, CG 1050 and Casia 1. Datasets utilized for training include image forgeries of major varieties which makes the model a generalized forgery detection model. The selected image reconstruction model showed showed noticeably faster convergence and accuracy from the training and validation curves. The context information from the selected model is utilized for forgery classification and is experimented with different classifiers like Softmax, Sigmoid and SVM. Experiments were also carried out with all the 4 datasets considering Accuracy, Precision, Recall and F1 score as the performance metrics. Study revealed that each dataset has varying responses to different classifiers. Finally, the efficacy of the selected model is weighed up with the existing CNN models and some of the state-of-the-art techniques. VGG19, ResNet 50 and MobileNet was used as the encoder in the transfer learning model and the entire study is experimented using different datasets. Results show that the proposed framework demonstrates competitive and stable mean Precision and F1-score performance across datasets in comparison with the experiments done with the existing CNN models as image reconstruction feature extractors. The simplicity and efficiency of the proposed model architecture are illustrated by comparing its architecture and performance measures with some of the current state-of-the-art techniques. Since the encoder-decoder architecture is used, the feature dimensions are reduced and the findings demonstrate that the model guarantees good results without penalizing architectural complexity. Additionally, the model assures good performance even with a limited quantity of training data. The suggested approach concentrates on binary image-level classification, but does not specifically address pixel-level forgery localisation, which is still an area for future research. As demonstrated by the Coverage dataset, performance is sensitive to very small dataset size, underscoring the necessity of enough data diversity. Training on several heterogeneous datasets (Coverage, CG1050, MICC-F220, and CASIA 1) provides implicit exposure to variations in resolution and compression, even though explicit robustness experiments under controlled post-processing were not carried out. Nevertheless, systematic evaluation under aggressive post-processing will be taken into consideration in future studies.

Acknowledgment

This work is being done as a part of the Research program under APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala. The authors acknowledge the support and provision of all facilities essential for the completion of this work, from APJ Abdul Kalam Technological University, Thiruvananthapuram, and Rajagiri School of Engineering and Technology, Kakkanad, Kerala.

References

[1] S. Walia and K. Kumar, "Digital image forgery detection: a systematic scrutiny," Sep. 03, 2019, *Taylor and Francis Ltd.* doi: 10.1080/00450618.2018.1424241.

[2] S. Walia and K. Kumar, "An eagle-eye view of recent digital image forgery detection methods," in *Communications in Computer and Information Science*, Springer Verlag, 2018, pp. 469–487. doi: 10.1007/978-981-10-8660-1_36.

[3] G. Kumar and P. K. Bhatia, "A detailed review of feature extraction in image processing systems," in *International Conference on Advanced Computing and Communication Technologies, ACCT*, Institute of Electrical and Electronics Engineers Inc., 2014, pp. 5–12. doi: 10.1109/ACCT.2014.74.

[4] K. B. Meena and V. Tyagi, "A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms," *Multimed. Tools Appl.*, vol. 79, no. 11–12, pp. 8197–8212, Mar. 2020, doi: 10.1007/s11042-019-08343-0.

[5] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[6] N. B. A. Warif *et al.*, "Copy-move forgery detection: Survey, challenges and future directions," Nov. 01, 2016, *Academic Press*. doi: 10.1016/j.jnca.2016.09.008.

[7] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image Splicing Localization Using A Multi-Task Fully Convolutional Network (MFCN)," Sep. 2017, doi: 10.1016/j.jvcir.2018.01.010.

[8] A. H. Khalil, A. Z. Ghalwash, H. A. G. Elsayed, G. I. Salama, and H. A. Ghalwash, "Enhancing Digital Image Forgery Detection Using Transfer Learning," *IEEE Access*, vol. 11, pp. 91583–91594, 2023, doi: 10.1109/ACCESS.2023.3307357.

[9] Y. Ji, H. Zhang, Z. Zhang, and M. Liu, "CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances," *Inf. Sci. (N Y).*, vol. 546, pp. 835–857, Feb. 2021, doi: 10.1016/j.ins.2020.09.003.

[10] N. K. Rathore, N. K. Jain, P. K. Shukla, U. S. Rawat, and R. Dubey, "Image Forgery Detection Using Singular Value Decomposition with Some Attacks," *National Academy Science Letters*, vol. 44, no. 4, pp. 331–338, Aug. 2021, doi: 10.1007/s40009-020-00998-w.

[11] N. Krawetz, "A Picture's Worth... Digital Image Analysis and Forensics," 2007. [Online]. Available: www.hackerfactor.com

[12] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image Vis. Comput.*, vol. 27, no. 10, pp. 1497–1503, Sep. 2009, doi: 10.1016/j.imavis.2009.02.001.

[13] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012, doi: 10.1109/TIFS.2012.2202227.

[14] C. Lin *et al.*, "Copy-move forgery detection using combined features and transitive matching."

[15] Y. Rodriguez-Ortega, D. M. Ballesteros, and D. Renza, "Copy-move forgery detection (Cmfd) using deep learning for image and video forensics," *J. Imaging*, vol. 7, no. 3, Mar. 2021, doi: 10.3390/jimaging7030059.

[16] N. Goel, S. Kaur, and R. Bala, "Dual branch convolutional neural network for copy move forgery detection," *IET Image Process.*, vol. 15, no. 3, pp. 656–665, Feb. 2021, doi: 10.1049/ipr2.12051.

[17] N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar, "Design of Automated Deep Learning-Based Fusion Model for Copy-Move Image Forgery Detection," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/8501738.

[18] T. M. Mohammed *et al.*, "Boosting image forgery detection using resampling features and Copy-move analysis," in *IS and T International Symposium on Electronic Imaging Science and Technology*, Society for Imaging Science and Technology, 2018. doi: 10.2352/ISSN.2470-1173.2018.07.MWSF-118.

[19] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath, "Exploiting Spatial Structure for Localizing Manipulated Image Regions."

[20] P. Dixit and S. Silakari, "Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review," Feb. 01, 2021, *Elsevier Ireland Ltd*. doi: 10.1016/j.cosrev.2020.100317.

[21] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva, "TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization." [Online]. Available: https://grip-unina.github.io/TruFor/.

[22] Y. Wu, W. Abdalmageed, and P. Natarajan, "ManTra-Net: Manipulation Tracing Network For Detection And Localization of Image Forgeries With Anomalous Features."

[23] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy-move image forgery with source/target localization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11210 LNCS, pp. 170–186, 2018, doi: 10.1007/978-3-030-01231-1_11/FIGURES/7.

[24] 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*, vol. 11, 2021, doi: 10.22967/HCIS.2021.11.032.

[25] E. U. H. Qazi, T. Zia, and A. Almorjan, "Deep Learning-Based Digital Image Forgery Detection System," *Applied Sciences (Switzerland)*, vol. 12, no. 6, Mar. 2022, doi: 10.3390/app12062851.

[26] A. Ahmad Aminu, N. Nnanna Agwu, and A. Steve, "Detection and Localization of Image Tampering using Deep Residual UNET with Stacked Dilated Convolution," *IJCSNS International Journal of Computer Science and Network Security*, vol. 21, no. 9, 2021, doi: 10.22937/IJCSNS.2021.21.9.28.

[27] Abhishek and N. Jindal, "Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation," *Multimed. Tools Appl.*, vol. 80, no. 3, pp. 3571–3599, Jan. 2021, doi: 10.1007/s11042-020-09816-3.

[28] K. D. Kadam, S. Ahirrao, K. Kotecha, and S. Sahu, "Detection and Localization of Multiple Image Splicing Using MobileNet V1," *IEEE Access*, vol. 9, pp. 162499–162519, 2021, doi: 10.1109/ACCESS.2021.3130342.

[29] F. Z. El Biach, I. Iala, H. Laanaya, and K. Minaoui, "Encoder-decoder based convolutional neural networks for image forgery detection," *Multimed. Tools Appl.*, vol. 81, no. 16, pp. 22611–22628, Jul. 2022, doi: 10.1007/s11042-020-10158-3.

[30] S. S. Ali, I. I. Ganapathi, N. S. Vu, S. D. Ali, N. Saxena, and N. Werghi, "Image Forgery Detection Using Deeplearning by Recompressing Images," *Electronics (Switzerland)*, vol. 11, no. 3, Feb. 2022, doi: 10.3390/electronics11030403.

[31] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning Rich Features for Image Manipulation Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Dec. 2018, pp. 1053–1061. doi: 10.1109/CVPR.2018.00116.
[32] H. Li, W. Luo, X. Qiu, and J. Huang, "Image Forgery Localization via Integrating Tampering Possibility Maps," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1240–1252, May 2017, doi: 10.1109/TIFS.2017.2656823.
[33] H. Chen, C. Chang, Z. Shi, and Y. Lyu, "Hybrid features and semantic reinforcement network for image forgery detection," in *Multimedia Systems*, Springer Science and Business Media Deutschland GmbH, Apr. 2022, pp. 363–374. doi: 10.1007/s00530-021-00801-w.
[34] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and Encoder-Decoder Architecture for Detection of Image Forgeries," Mar. 2019, doi: 10.1109/TIP.2019.2895466.
[35] G. Mazaheri, N. Chowdhury Mithun, J. H. Bappy, and A. K. Roy-Chowdhury, "A Skip Connection Architecture for Localization of Image Manipulations."
[36] B. Yang, Z. Li, and T. Zhang, "A real-time image forensics scheme based on multi-domain learning," in *Journal of Real-Time Image Processing*, Springer, Feb. 2020, pp. 29–40. doi: 10.1007/s11554-019-00893-8.
[37] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.06751
[38] Y. Rao, J. Ni, and H. Xie, "Multi-semantic CRF-based attention model for image forgery detection and localization," *Signal Processing*, vol. 183, Jun. 2021, doi: 10.1016/j.sigpro.2021.108051.
[39] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.07285
[40] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding Batch Normalization."
[41] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning Activation Functions to Improve Deep Neural Networks," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6830
[42] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review."
[43] N. Ketkar, "Introduction to Keras," in *Deep Learning with Python*, Apress, 2017, pp. 97–111. doi: 10.1007/978-1-4842-2766-4_7.
[44] K. Janocha and W. M. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," Feb. 2017, [Online]. Available: http://arxiv.org/abs/1702.05659
[45] S. Ucliimurat, Y. Hamamotot, and S. Tomitas, "On the Effect of the Nonlinearity of the Sigmoid Function in Artificial Neural Network Classifiers."
[46] K. Duan, S. S. Keerthi, W. Chu, S. Krishnaj Shevade, and A. N. Poo, "Multi-category Classification by Soft-Max Combination of Binary Classifiers."
[47] J. Weston and C. Watkins, "Multi-class Support Vector Machines," 1998.
[48] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "COVERAGE — A novel database for copy-move forgery detection," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 2016, pp. 161–165. doi: 10.1109/ICIP.2016.7532339.
[49] M. Castro, D. M. Ballesteros, and D. Renza, "A dataset of 1050-tampered color and grayscale images (CG-1050)," *Data Brief*, vol. 28, Feb. 2020, doi: 10.1016/j.dib.2019.104864.

Authors' Profiles

Jisha K. R. was born in Kerala, India on April 17, 1982. She received B.Tech. degree in Electronics and Instrumentation Engineering from College of Engineering, Kidangoor under Cochin University of Science and Technology, Kochi, India in 2004 and M.E degree in Applied Electronics from Kumaraguru College of Technology under Anna University, Chennai, India in 2012. Her major field of study was electronics. She is currently pursuing Ph.D. degree from APJ Abdul Kalam Technological University, Thiruvananthapuram, India at Rajagiri School of Engineering and Technology, Kakkanad, Kerala. Her area of research is Deep Learning based Image processing.

Sabna N. was born in Ernakulam, India, in 1984. She received B.Tech. degree in Electronics and Communication Engineering from M.G. University, Kottayam, India in the year 2006 and MTech. degree (Digital Electronics) as well as Ph.D. degree from the Cochin University of Science and Technology, Kochi, India in 2011 and 2018 respectively. She has been working as Assistant Professor in Rajagiri School of Engineering and Technology, Kakkanad, Kerala, since January 2017. Her research interests include Signal and Image Processing, Underwater Acoustics, Underwater Acoustic Communication, etc.

How to cite this paper: Jisha K. R., Sabna N., "An Effective Semi-Supervised Feature Extraction Model with Reduced Architectural Complexity for Image Forgery Classification", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.18, No.1, pp. 33-50, 2026. DOI:10.5815/ijigsp.2026.01.03