Extraction of image resampling using correlation aware convolution neural networks for image tampering detection

Manjunatha Shivanandappa[1], Malini M. Patil[2]
[1]Department of Information Science and Engineering, Global Academy of Technology, Bengaluru, India
[2]Department of Information Science and Engineering, J S S Academy of Technical Education, Bengaluru, India

Article Info

Article history:

Received Apr 24, 2021
Revised Jan 17, 2022
Accepted Jan 31, 2022

Keywords:

Compression
Convolution neural network
Deep learning
Image tampering detection
Image transformation
Resampling feature extraction

ABSTRACT

Detecting hybrid tampering attacks in an image is extremely difficult; especially when copy-clone tampered segments exhibit identical illumination and contrast level about genuine objects. The existing method fails to detect tampering when the image undergoes hybrid transformation such as scaling, rotation, compression, and also fails to detect under small-smooth tampering. The existing resampling feature extraction using the Deep learning techniques fails to obtain a good correlation among neighboring pixels in both horizontal and vertical directions. This work presents correlation aware convolution neural network (CA-CNN) for extracting resampling features for detecting hybrid tampering attacks. Here the image is resized for detecting tampering under a small-smooth region. The CA-CNN is composed of a three-layer horizontal, vertical, and correlated layer. The correlated layer is used for obtaining correlated resampling feature among horizontal sequence and vertical sequence. Then feature is aggregated and the descriptor is built. An experiment is conducted to evaluate the performance of the CA-CNN model over existing tampering detection methodologies considering the various datasets. From the result achieved it can be seen the CA-CNN is efficient considering various distortions and post-processing attacks such joint photographic expert group (JPEG) compression, and scaling. This model achieves much better accuracies, recall, precision, false positive rate (FPR), and F-measure compared existing methodologies.

Corresponding Author:

Manjunatha Shivanandappa
Department of Information Science and Engineering, Global Academy of Technology
Bengaluru, 560 060, India
Email: manju.dvg2020@gmail.com

1. INTRODUCTION

With the growth of technology and availability of image editing software adopting artificial intelligence technique makes tampering detection challenging as both tampered image looks very similar to the original image. Image can tamper through different means such as content preserving and content change [1]. The primary tampering attacks such as splicing, copy-clone, and object removal, are used for changing the semantic representation of an image. On contrary, the secondary tampering attacks such as compression, blurring, contrast enhancement are not a big concern as they do not change the meaning/structure of an image. Thus, this work focuses on detecting the primary tampering attacks and also improve the accuracy of localization of tampered regions at the pixel level.

The state-of-art tampering detection methodologies have majorly focused on detecting to identify whether an image has been tampered with or not [2], [3]. In [4], [5] the tampering region is localized at a pixel level. In [6], [7] focused on localizing tampering at the patch level and added noise into frequency domain [8], [9] of joint photographic expert group (JPEG) compressed image for improving resampling detection performance. In recent times, the number of deep learning-based tampering detection [10]–[12] such as convolutional neural networks (CNN) [13]–[15], long-short term memory (LSTM) and stacked auto-encoders (SAE) [16] have been presented. In media crime scene investigation, the majority of state-of-art tampering detection methodologies have focused on detecting certain types of tampering only such as splicing [17] and copy-clone [18], [19]. As a result, these methodologies cannot be used for detecting hybrid tampering detection. This paper aimed at detecting hybrid tampering attacks and segmenting tampering regions by employing an improved convolution neural network [4].

Segmentation of tampered regions is a challenging task. Recently, CNN-based semantic segmentation methodologies [20], [21] have attained wide attention. In [21], used fully connected CNN for analyzing region shape and object content by extracting feature sets at different levels in a hierarchical manner. The CNN-based framework works very well in the area of object detection [19] and segmentation [20], [21] in learning and a better understanding of the content of different segments. Unlike object segmentation, tampered segments could be copied objects from different regions of an image or could be removed objects. A good, tampered image will have good similarities among authenticated and fake images [14]. Although convolution neural network produces spatial maps for different segments of multimedia content, they achieve very poor performance in generalizing different artifacts induced by different tampering methodologies. As a result, tampering region segmentation using a standard convolution neural network may not produce a good result. In [4] carried out a comparative analysis of various existing tampering region segmentation methodologies [20], [21] and showed they do not perform well for object removal and copy-move tampering [22]–[25]. Image forgeries create certain artifacts such as compression, and resampling, which can be better learned using resampling features [6], [26]. Due to interpolation resampling introduces periodic correlation between the pixels. The CNN-based tampering detection methodologies shows good translational invariance to produce spatial maps across different segment of multimedia content, and certain artifacts are well-learned using resampling feature sets [27]; which can be utilized to locate tampered segments [28], [29]. From extensive, it can be seen resampling feature detection of hybrid attacks within copy-clones attacks is a challenging task. The existing tampering detection method [3], [18], [30] provides a poor result when a tampered image is noisy and also failed to detect the tampering segment under the small-smooth region.

The major challenges of tampering detection: detection of multiple copy-clone tampering within an image and distinguish source and tampered region is challenging. Detecting tampering under a small and smooth region is very difficult [31], [32]. Extracting resampling feature correlation among horizontal and vertical directions using the standard CNN model is challenging. How to extract resampling feature when the image is extremely noisy. It is extremely very difficult to detect tampering when a different type of tampering operation such as scaling, rotation, compression is performed within a copy-clone attack [33]–[35].

The research hypothesis is that the state-of-art tampering detection methodologies [36] using deep learning techniques are effective in detecting various types of tampering attacks. However, existing models predominantly achieves poor results when hybrid attacks are introduced into an image; for example, when a copy-clone attack is transformed by rotation, scaling, and compression. This is because existing model fails to learn correlation among neighboring pixel. This working hypothesis is that effective learning of neighboring pixels and correlating relationship [37] in obtaining effective resampling feature extraction for detecting a hybrid attack.

For overcoming research issues this paper presents an improved CNN architecture namely the correlation aware convolution neural network (CA-CNN) model for extracting resampling features. To detect tampering segments under small and smooth segments, the image is resized [37], [38]. Then, even under a noisy environment, the resampling feature can be extracted using CA-CNN architecture with good correlation. Finally, these features are trained considering different images, and a descriptor is constructed for detecting image tampering.

The contribution of research work is described: this paper presented a correlation-aware convolution neural network for detecting tampering in the image. The CA-CNN model can exploit resampling feature correlation among horizontal and vertical directions by introducing a correlation layer. The CA-CNN can detect multiple tampering within an image considering a noisy environment with different kinds of tampering operations such as scaling, rotation, and compression. The model achieves better tampering detection performance when trained with the CA-CNN model considering diverse tampering datasets such as coverage, media integration and communication center (MICC), and copy move forgery detection (CoMoFoD); no prior methodology has considered performance evaluation considering all these datasets together. The CA-CNN-based tampering detection method achieves better recall, precision, and F1-score performance than existing tampering detection methodologies.

2. RESAMPLING FEATURE-BASED TAMPERING DETECTION USING CORRELATION AWARE CONVOLUTION NEURAL NETWORK

This section presents image tampering methodologies using resampling features and convolution neural networks. First, present preprocessing and resampling feature extraction for performing tampering detection. Second, the extracted features are trained using the CA-CNN model shown in Figure 1 for detecting whether an image is tampered with or not and segment the tampered region. The step-by-step process of proposed resampling feature-based tampering detection using CA-CNN is shown in algorithm 1.

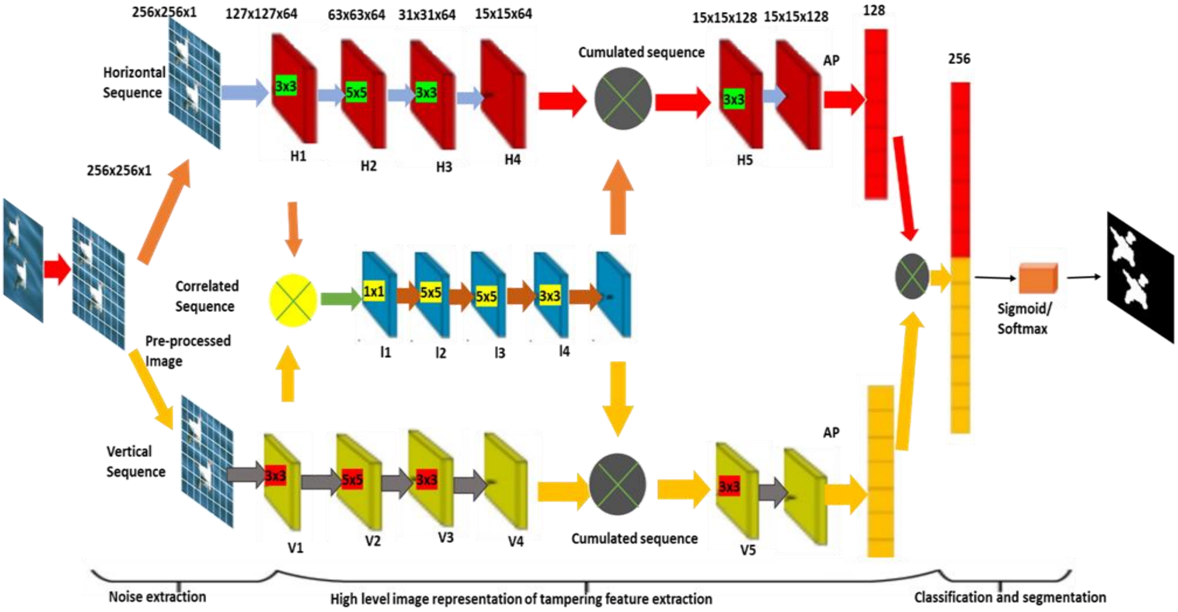

Figure 1. The architecture of CA-CNN for image tampering detection

Algorithm 1. Resampling feature-based tampering detection using CA-CNN

Step 1. Start.

Step 2. Load image.

Step 3. Segment the image into non-overlapping patches of 64 (i.e., 8*8).

Step 4. Extract resampling features with different dimensions using a noise-invariant layer.

Step 5. Extract high-level features in each patch in both horizontal and vertical directions.

Step 6. Extract common features among horizontal and vertical are cumulated and aggregated.

Step 7. Aggregated features are fed into SoftMax layer to perform classification image is tampered with or not.

Step 8. Segment the tampered region.

Step 9. Stop.

2.1. Preprocessing and resampling feature detection and extraction

In general, the images are tampered with using the following operations such as object removal, splicing, and copy-move. This tampering affects the statistical feature alongside the edges of the forged segments. In [29], the resampling detection method is presented using affine transformation and the Laplacian operator for extracting the resampling features for respective patches. This work uses a similar methodology for the extraction of resampling features in a given image. First, the image is segmented into a non-overlapping patch size of 64 (i.e., 8*8). When considering an image with the size of 512*512, then each patch dimension size will be 64*64. Further, for producing magnitude of linear projected error for different patches Laplacian operator is used [13]. For accumulating errors concerning the different angles of projection this work uses affine transformation because there exist periodic correlations among resampling signals. At last, fast fourier transform (FFT) is applied for identifying the resampling features periodic characteristic of the signals. Generally, the resample feature sets have the capability of identifying different resampling nature such as rotation, up or down-sampling, and JPEG thresholding.

For bringing good tradeoffs between increasing accuracy and reducing computation complexity here the image is resized to 512*512 which may induce certain artifacts such as up- or down-sampling, and image quality variations. In [13] showed that the resampling feature can be utilized for classifying the aforementioned artifacts. Further, resampling feature sets are used for classifying patches. However, in this work, it is used for localizing at a pixel level. For obtaining a higher number of features it is important to bring good tradeoffs in choosing the patch size. This is because resampling signal can be easily established in larger patch size as it will have a higher number of repeated features; however, identifying small, tampered segments will be difficult for localizing it. The existing resampling-based tampering detection methodologies extracted resampling features considering a block size of 8*8. However, in this work patch size is set to 32*32 for obtaining more useful information. The main factor of using the resampling feature within the patches is to establish the nature of local artifacts because of different tampering.

The outcome of CNN mainly depends on the organization of the patches. It can either be ordered in vertical or horizontal directions; however, it fails to obtain relevant local feature information. This is because, if we are arranging the patches in a vertical direction, then the patch sets of different neighbors horizontally will be disconnected by a complete column of patches. Thus, takes a lot of time and CNN fails to bring a good correlation among these patches. Similarly, if we traverse through horizontal direction over the rows will result in the same problem [19], [20]. Thus, in this work for establishing a good correlation among both directions here, we introduce an additional layer namely the correlation layer.

2.2. Correlation aware-CNN based tampering detection methodology

In this work we used deep learning methodology for detecting resampling features; here the tampering detection is considered as a pattern classification problem. The architecture of correlation aware-CNN (CA-CNN) architecture for tampering detection is shown in Figure. 1. The CA-CNN tampering detection methodology is composed of three layers. In layer one, the resampling features with different dimensions are captured using a noise-invariant layer; here the variance of the neighboring pixel among vertical and horizontal directions is captured. Second, in both horizontal and vertical directions, the tampered segment high-level features are extracted. Here for capturing association among vertical and horizontal directions, vertical and horizontal features are correlated and aggregated. Lastly, the aggregated features are given as an input for the SoftMax/sigmoid layer. The SoftMax layer is efficient in solving multiple tampering classification problems and sigmoid can be used for solving a binary tampering classification problem. More detail of CA-CNN architecture for tampering detection is discussed below.

2.3. Noise elimination

Resampling feature detection is challenging which generally relies on or is affected by the content of an image. However, some well-noted recent work has shown that the resampling feature can be obtained from the redundant feature of the spatial domain and doesn’t depend on the content of an image. In the work, the residual among particular pixels and its respective estimates obtained through interpolating its adjacent pixels, the noise is modeled. For modeling it, in this work a new convolution layer is introduced; this layer is the first layer and is known as the noise invariant layer. From Figure 1 it can be seen two high-pass filters are selected as convolution kernels for reducing training overhead. These filters are used for capturing neighboring pixels' variance in both horizontal and vertical streams. For example, an image with the size of a pixel of 256*256 is initially convolved with 3*1 and 1*3 filters considering padding and stride of 1. The aforementioned mention filter setting will aid in learning noisy features using correlation among local pixels. Thus, the noise-invariant layers will provide a noise map of forecasting residuals of 256*256*1.

2.4. Bidirectional sequence feature extraction

In this section, the resampling high-level feature is extracted from noise obtained through the noise-invariant layer. The existing method predominantly focused on extracting correlation features in one particular direction; thus, resampling feature detection performance is degraded. For addressing in this work the resampling feature is extracted through horizontal and as well as vertically also. These features are extracted independently and feature weight obtained through different directions is not shared. From Figure 1, we can see both vertical and horizontal sequences have five identical clusters. Each cluster is composed of four layers such as batch normalization, convolution, pooling, and activation layers. The last cluster is composed of a supplementary resampling feature obtained through a correlated sequence. Lastly, the feature extracted through different sequences is aggregated.

2.5. Correlation sequence feature extraction

This section aimed at modeling better decision making (i.e., linear fusion making) for extracting resampling behavior by merging bidirectional features. Thus, this paper presents an efficient correlation feature extraction method of a correlated sequence composed of four distinct clusters. The first cluster is composed of batch normalization, convolution, and an activation layer. The other clusters are composed of an added pooling layer; the feature obtained from the first cluster of both the sequence are aggregated and are represented through 1*1 convolution Kernels with stride 1 for obtaining linear feature fusion. The other three clusters are used for extracting high-level feature representations of cumulated features. Lastly, the outcome (i.e., feature map) obtained through the cumulated sequence is interpolated back towards vertical and horizontal sequences. The correlation feature learning method extract better feature without affecting feature extraction performance of both the sequence.

2.6. Classification

Here we present a fully connected layer using the sigmoid/SoftMax function that takes the final feature extracted from the previous layer as input to it. Using the proposed classifier, the probability that a certain feature fits the respective category is obtained, and the most ideal group is the outcome of the CNN classifier. The above-stated statement is functionally represented through (1) and (2):

$$P(z=1|y)=\frac{1}{1+f^{-a}} \tag{1}$$

$$P(z=k|y)=\frac{f^{a_k}}{\sum_{l=0}^{L} f^{a_k}} \tag{2}$$

where (1) represents the sigmoid function applied for binary tampering detection classification problems and represents the outcome of neurons of a fully connected layer. $P(z=1|y)$ represents the probabilities that y will put forth into the successful cluster. As shown in (2) defines the SoftMax function for performing multiple tampering detection, where a_k represents the outcome of respective k^{th} the neuron of a fully connected layer. $P(z=k|y)$ represents the probabilities that y will fall into a k^{th} cluster.

2.7. Convolution layer

The convolution layer is used for extracting features as described (3):

$$G_k^{(o)}=\sum_{l=0}^{L} G_l^{(o-1)} * \alpha_{lk}^{(o)}+c_k^{(o)} \tag{3}$$

where $*$defined two-dimensional convolution function, $\alpha_{lk}^{(o)}$ represent the l channel of respective k^{th} convolution kernel within o^{th} layer, $G_l^{(o-1)}$ defines the j^{th} feature map extracted within the $(o-1)^{th}$ layer, $G_k^{(o)}$ represents the k^{th} feature map constructed within o^{th} layer, and $c_k^{(o)}$ defines the k^{th} bias term of o^{th} layers. Here we used three convolution layers with the size of $(1*1, 3*3,$ and $5*5)$ with stride size is fixed to 1.

2.8. CNN batch normalization

In process of training, the feature maps computed using the convolution layer must be normalized for optimizing data distribution variations in the middle layer. For doing, a batch normalization layer is introduced between the activation and convolution layers. The process of carrying out batch optimization is mathematically represented using (4) to (7). The mean among entire data within the batch is computed using (4):

$$\beta=\frac{1}{n}\sum_{j=0}^{n} y_j \tag{4}$$

where β depicts mean, n represents feature size considered within the batch, y_j defines the j^{th} data within the batch. Similarly, the variance among the entire feature within the batch is computed using (5):

$$\gamma^2=\frac{1}{n}\sum_{j=0}^{n}\left(y_j-\beta\right)^2 \tag{5}$$

where γ^2 depicts the variance. Then every feature is normalized for generating a new set of features $\hat{y}_j$ with variance and mean set to 1 and 0, respectively. The $\hat{y}_j$ is computed as (6):

$$\hat{y}_j=\frac{y_j-\beta}{\sqrt{\gamma^2+\delta}} \tag{6}$$

where δ is greater than 0 defining a small floating-point digit. This is done for eliminating dividing by zero errors. The optimized feature is defined using (7):

$$z_j = \varphi \hat{y}_j + \omega \tag{7}$$

where φ and ω are feature learned by the CNN, z_j represent the j^{th} outcome of batch normalization layer. The optimized feature obtained uses a non-linear activation function for better feature representation; that is, significant changes are the previous layer due to trivial changes in the forward layer are eliminated by introducing the batch normalization layer.

2.9. CNN activation

Here the activation layer is composed of a non-linear function. To improve the tampering detection accuracies, the resampling feature sets extracted using convolution layer is transformed into different space. Generally, rectified linear unit (ReLU), sigmoid, and TanH are used in the activation layer. Generally, TanH is mostly preferred over Sigmoid in most applications because TanH's average output is zero. ReLU is much faster than TanH, but its training accuracies are poor when the learning rate is kept larger. On the other side, the TanH can increase constantly concerning features; thus, attain very effective outcomes concerning features with major variance. As a result, in this work TanH function is used in the activation layer.

2.10. Pooling layer

In this layer, the feature maps are down-sampled for reducing their element size. Further, it signifies hierarchical patter by cumulating the observed window of successive convolution layers. Here we use averaging pooling (AP) and max-pooling (MP) function. In, AP the feature maps are down-sampled to 1 through averaging pooling, and for reducing the model parameter the AP replaces the fully connected layer. An important thing to be noted here is that the AP is used just for the last pooling layer of both vertical and horizontal sequences. The MP for every input feature provides outcome with maximum value and except the fifth layer of both horizontal and vertical sequence, it is applied to all the pooling layers. The kernel size of Max pooling is set to 3*3 with stride set to 2 for capturing the pattern of the adjacent pixel concerning each pixel. The proposed tampering detection using the CA-CNN framework achieves a much better detection and segmentation outcome than the traditional CNN-based tampering detection methodology which is experimentally shown below.

3. RESULTS AND DISCUSSION

Here experiment is carried for evaluating the performance of tampering detection performance using the proposed CA-CNN method and existing CNN-based tampering detection methodologies considering different datasets. Here performance is evaluated using MICC-600, Coverage, and CoMoFoD dataset. The aforementioned dataset is widely used in most recent tampering detection methods for validating performance.

The CA-CNN model is using Python, C++, and MATLAB libraries. The performance of CA-CNN and the existing tampering detection method are evaluated in terms of the following metrics such as true positive rate (TPR) (i.e., recall), F1 score, and false positive rate (FPR). To verify the performance of the proposed CA-CNN-based image forensics, the experimental results are compared to existing tampering detection methodologies [1], [8], [9], [26], [30] to perform the forgeries, including copying and translations, scaling, rotation, and compression.

3.1. Performance evaluation on MICC dataset

Here experiment is conducted using the MICC-F600 dataset. The dataset is composed of 440 original images and 160 tampered images. The tampering segmentation outcome achieved using the proposed CA-CNN and existing tampering detection model is shown in Figure 2. The Figure 2(a) shows the original image, Figure 2(b) shows respective ground truth of tampered region, segmentation outcome achieved using existing and CA-CNN tampering model is shown in Figures 2(c) and 2(d), respectively. Further, the accuracy performance of the proposed CA-CNN-based tampering detection method over the existing tampering detection method is carried is shown in Table 1. From Figure 2 it can be seen the proposed CA-CNN model achieves better tampering region segmentation outcomes when compared with existing models. From the result achieved it can be seen the proposed CA-CNN-based tampering detection method achieves a much superior outcome than the existing tampering detection method in terms of Recall/TPR, FPR, and F1-Score for the MICC-F600 dataset. Thus, the proposed CA-CNN-based tampering detection method is robust in detecting forged segments considering rotation and scaling.

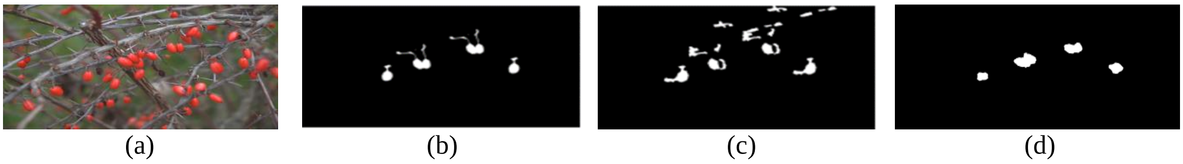

Figure 2. Comparative analysis of proposed CA-CNN-based tampering detection method over existing tampering detection methodology: (a) original image, (b) ground truth, (c) existing tampering region segmentation method [8], and (d) tampering region segmentation method using CA-CNN

Table 1. Comparative analysis of proposed CA-CNN-based tampering detection method over existing tampering detection method for MICC-F600 dataset

	Recall/TPR	FPR	F1-Score	FP
Raju *et al.* 2018 [22]	89.14	-	92.6	-
Li *et al.* 2019 [30]	97.5	5.68	91.5	91.8
CA-CNN	99.1	1.4	98.6	96.5

3.2. Performance evaluation on coverage dataset

Here experiment is carried out using a coverage dataset. The dataset is very challenging, which contains 100 copy-move tampered images and the corresponding original images with similar but genuine objects. The tampering segmentation outcome achieved using the proposed CA-CNN and the existing tampering detection model is shown in Figures 3 and 4. The Figure 3(a) shows the original image, Figure 3(b) shows respective ground truth of tampered region, segmentation outcome achieved using base, Base-Ada-Aten, Ar-Net, and CA-CNN tampering model is shown in Figures 3(c), 3(d), 3(e), and 3(f), respectively.

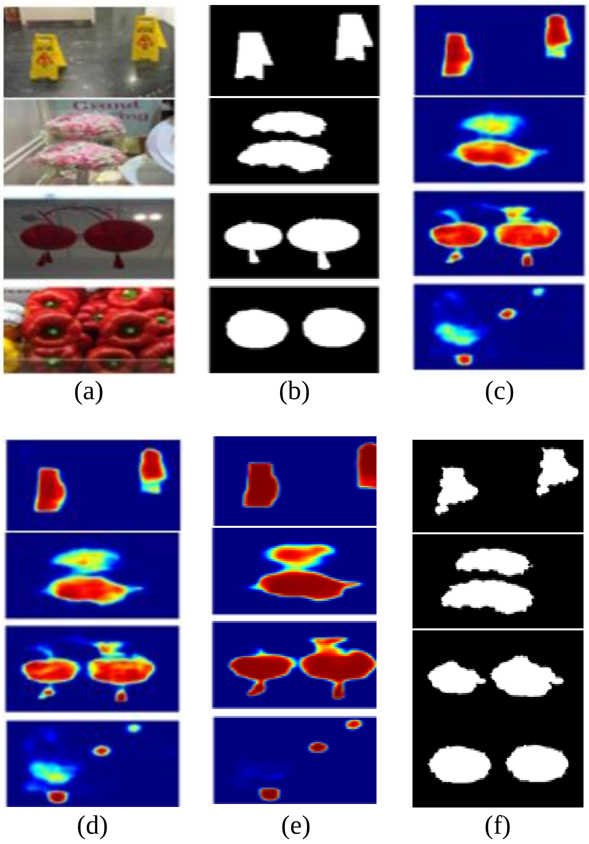

Figure 3. Tampering region segmentation outcome using coverage dataset: (a) original image, (b) ground truth image, (c) base, (d) base-ada-aten, (e) Ar-net [8], and (f) CA-CNN

Further, the accuracy performance of the proposed CA-CNN-based tampering detection method over the existing tampering detection method is shown in Table 2. From Figure 3 we can see CA-CNN achieves better tampering segmentation outcomes for all images except image 3. The Figure 4(a) shows the original image, Figure 4(b) shows respective ground truth of tampered region, segmentation outcome achieved using BusterNet, STRDNet (source/target region distinguishment network), and CA-CNN tampering model is shown in Figures 4(c), 4(d), and 4(e), respectively. Similarly, in Figure 4 we can see CA-CNN achieves very good tampering segmentation outcomes for image 1 and archives not that good tampering segmentation outcomes for image 2. On the overall result achieved it can be seen the proposed CA-CNN model achieves better tampering region segmentation outcomes when compared with existing models. From the result achieved it can be seen the proposed CA-CNN-based tampering detection method achieves a much superior outcome than the existing tampering detection method in terms of Accuracy and F1-Score for Coverage dataset.

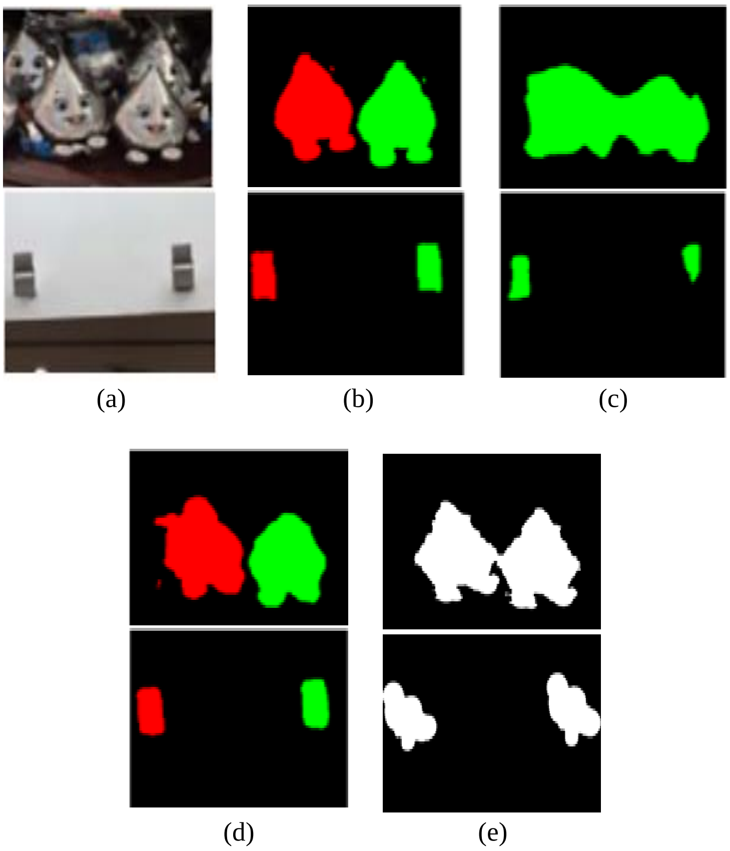

Figure 4. Tampering region segmentation outcome using coverage dataset: (a) original image, (b) ground truth image, (c) BusterNet [1], (d) STRDNet [26], and (e) CA-CNN

Table 2. Comparative analysis of proposed CA-CNN based tampering detection method over existing tampering detection method for Coverage dataset

Model	Accuracy	F1
Base [9]	0.8581	-
Base-Ada-Atten [9]	0.8542	-
AR-Net [8]	0.8488	-
BusterNet [1]	-	0.464
STRDNet [26]	-	0.677
CA-CNN	0.8563	0.7456

3.3. Performance evaluation on CoMoFoD dataset

Here experiment is carried out using the CoMoFoD dataset. The dataset contains 200 base tampered images. To hide the traces of manipulation, each base image will undergo 25 post-processing methods, with a total of 5k tampered images. The tampering segmentation outcome achieved using the proposed CA-CNN and existing tampering detection model is shown in Figure 5. The Figure 5(a) shows the original image, Figure 5(b) shows respective ground truth of tampered region, segmentation outcome achieved using BusterNet, STRDNet, and CA-CNN tampering model is shown in Figures 5(c), 5(d), and 5(e), respectively. Further, the accuracy performance of the proposed CA-CNN-based tampering detection method over the existing tampering detection method is carried is shown in Table 3. From Figure 5 we can see CA-CNN achieves better tampering segmentation outcomes for all images except image 1. On overall result achieved the proposed CA-CNN model achieves better tampering region segmentation outcome when compared with existing models. From the result achieved it can be seen the proposed CA-CNN-based tampering detection method achieves a much superior outcome than the existing tampering detection method in terms of Recall/TPR, precision, and F1-Score for the CoMoFoD dataset.

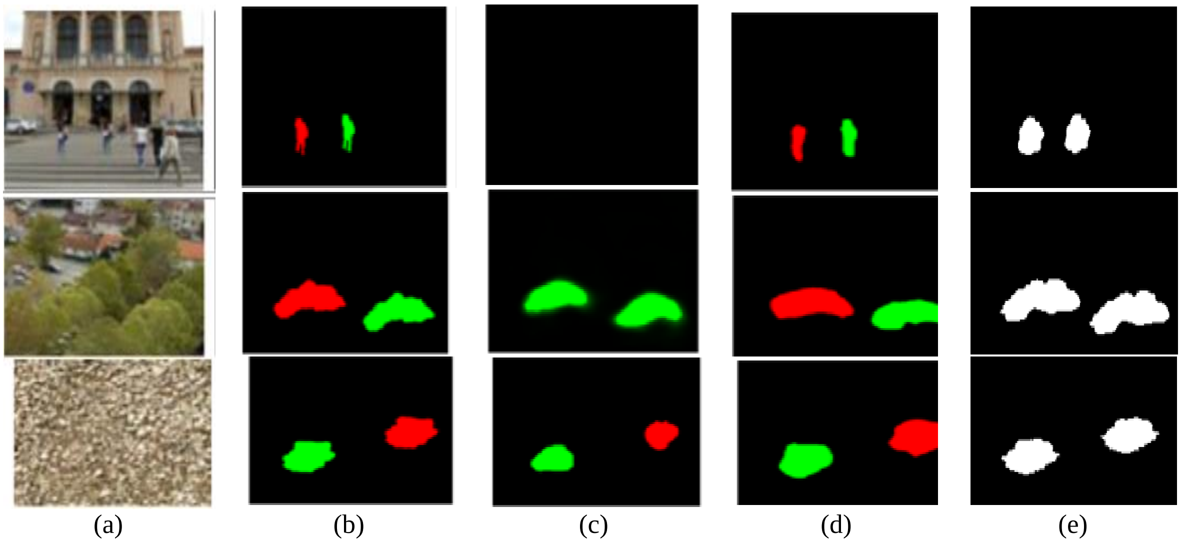

Figure 5. Tampering region segmentation outcome using CoMoFoD dataset: (a) original image, (b) ground truth image, (c) BusterNet [1], (d) STRDNet [26], and (e) CA-CNN

Table 3. Comparative analysis of proposed CA-CNN based tampering detection method over existing tampering detection method for CoMoFoD dataset

Model	Recall	Precision	F1
Base [26]	0.3811	0.4768	0.4236
Base-Ada-Atten [26]	0.4075	0.4661	0.4349
AR-Net [8]	0.4655	0.5421	0.5009
BusterNet [39]	-	-	0.493
STRDNet [40]	-	-	0.511
CA-CNN	0.89	0.7654	0.856

4. CONCLUSION

This paper presented robust tampering detection using the correlation-aware-CNN model. The CA-CNN-based tampering detection methodologies can effectively classify forged and non-forged segments and can semantically segment the forged region. The CA-CNN model can retain spatial features by using resampling features among different patches and establish correlation among tampered and non-tampered patches by employing correlated layers. Then, these resampling features are aggregated for eliminating spatial dependencies, and a descriptor is built for the whole image. An experiment is conducted on standard MICC-F600, D0, Coverage, and CoMoFoD datasets which includes different copy-clone, scaling, rotation, and compression. From the results attained it can be seen the CA-CNN-based tampering detection model achieves a much superior True positive rate, F1 score, False Positive rate, F-measure, and accuracies when compared with the existing tampering detection model. Future work would consider evaluating the accuracies of the proposed tampering model at pixel level and carry out comparative analysis over existing tampering detection methodologies. Further, evaluate the model considering a more diverse dataset. Along with, would consider improving tampering and segmentation performance.

REFERENCES

[1] J. Fu *et al.*, "Dual attention network for scene segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 3141–3149, doi: 10.1109/CVPR.2019.00326.

[2] 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.

[3] M. E. Azol, N. H. Ramli, Y. S. L. Lee, and S. A. Abuzar, "A coarse-to-fine copy-move image forgery detection method based on discrete cosine transform," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 14, no. 2, pp. 843–851, May 2019, doi: 10.11591/ijeecs.v14.i2.pp843-851.

[4] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath, "Exploiting spatial structure for localizing manipulated image regions," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 4980–4989, doi: 10.1109/ICCV.2017.532.

[5] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based CNN features," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jul. 2017, pp. 1855–1864, doi: 10.1109/CVPRW.2017.232.

[6] J. Bunk *et al.*, "Detection and localization of image forgeries using resampling features and deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jul. 2017, pp. 1881–1889, doi: 10.1109/CVPRW.2017.235.

[7] Y. Liu, Q. Guan, X. Zhao, and Y. Cao, "Image forgery localization based on multi-scale convolutional neural networks," *arXiv preprint arXiv:1706.07842*, Jun. 2017, doi: 10.1109/TGRS.2018.2848473.

[8] Y. Zhu, C. Chen, G. Yan, Y. Guo, and Y. Dong, "AR-Net: adaptive attention and residual refinement network for copy-move forgery detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6714–6723, Oct. 2020, doi: 10.1109/TII.2020.2982705.

[9] J.-L. Zhong and C.-M. Pun, "An end-to-end dense-inceptionnet for image copy-move forgery Detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2134–2146, 2020, doi: 10.1109/TIFS.2019.2957693.

[10] L. Jiao and J. Zhao, "A survey on the new generation of deep learning in image processing," *IEEE Access*, vol. 7, pp. 172231–172263, 2019, doi: 10.1109/ACCESS.2019.2956508.

[11] Z. J. Barad and M. M. Goswami, "Image forgery detection using deep learning: a survey," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2020, pp. 571–576, doi: 10.1109/ICACCS48705.2020.9074408.

[12] A. Kuznetsov, "Digital image forgery detection using deep learning approach," *Journal of Physics: Conference Series*, vol. 1368, no. 3, Nov. 2019, doi: 10.1088/1742-6596/1368/3/032028.

[13] B. Bayar and M. C. Stamm, "A deep learning approach to Universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2016, pp. 5–10, doi: 10.1145/2909827.2930786.

[14] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2016, pp. 1–6, doi: 10.1109/WIFS.2016.7823911.

[15] R. Huang, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen, "A method for identifying origin of digital images using a convolution neural network," *arXiv preprint arXiv: 1911.00655*, Nov. 2019.

[16] Y. Zhang, J. Goh, L. L. Win, and V. Thing, "Image region forgery detection: a deep learning approach.," *Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016*, 2016. .

[17] S. T. M. and K. B. Ramesh, "Novel framework for optimized digital forensic for mitigating complex image attacks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5198–5207, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5198-5207.

[18] A. Pourkashani, A. Shahbahrami, and A. Akoushideh, "Copy-move forgery detection using convolutional neural network and K-mean clustering," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2604–2612, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2604-2612.

[19] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection," *arXiv preprint arXiv: 1909.06751*, Sep. 2019.

[20] Y. Liang, Y. Fang, S. Luo, and B. Chen, "Image resampling detection based on convolutional neural network," in *2019 15th International Conference on Computational Intelligence and Security (CIS)*, Dec. 2019, pp. 257–261, doi: 10.1109/CIS.2019.00061.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, Nov. 2015.

[22] P. M. Raju and M. S. Nair, "Copy-move forgery detection using binary discriminant features," *Journal of King Saud University - Computer and Information Sciences*, Nov. 2018, doi: 10.1016/j.jksuci.2018.11.004.

[23] H.-Y. Huang and A.-J. Ciou, "Copy-move forgery detection for image forensics using the superpixel segmentation and the Helmert transformation," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, Dec. 2019, doi: 10.1186/s13640-019-0469-9.

[24] X.-Y. Wang, C. Wang, L. Wang, L.-X. Jiao, H.-Y. Yang, and P.-P. Niu, "A fast and high accurate image copy-move forgery detection approach," *Multidimensional Systems and Signal Processing*, vol. 31, no. 3, pp. 857–883, Jul. 2020, doi: 10.1007/s11045-019-00688-x.

[25] A. K. Jaiswal and R. Srivastava, "A technique for image splicing detection using hybrid feature set," *Multimedia Tools and Applications*, vol. 79, no. 17–18, pp. 11837–11860, May 2020, doi: 10.1007/s11042-019-08480-6.

[26] B. Chen, W. Tan, G. Coatrieux, Y. Zheng, and Y.-Q. Shi, "A serial image copy-move forgery localization scheme with source/target distinguishment," *IEEE Transactions on Multimedia*, vol. 23, pp. 3506–3517, 2021, doi: 10.1109/TMM.2020.3026868.

[27] A. Flenner, L. Peterson, J. Bunk, T. M. Mohammed, L. Nataraj, and B. S. Manjunath, "Resampling forgery detection using deep learning and a-contrario analysis," *arXiv:1802.03154*, Mar. 2018.

[28] 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," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286–3300, Jul. 2019, doi: 10.1109/TIP.2019.2895466.

[29] G. Cao, A. Zhou, X. Huang, G. Song, L. Yang, and Y. Zhu, "Resampling detection of recompressed images via dual-stream convolutional neural network," *arXiv preprint arXiv:1901.04637*, Jan. 2019.

[30] Y. Li and J. Zhou, "Fast and effective image copy-move forgery detection via hierarchical feature point matching," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1307–1322, May 2019, doi: 10.1109/TIFS.2018.2876837.
[31] H. Chen, X. Yang, and Y. Lyu, "Copy-move forgery detection based on keypoint clustering and similar neighborhood search algorithm," *IEEE Access*, vol. 8, pp. 36863–36875, 2020, doi: 10.1109/ACCESS.2020.2974804.
[32] B. Diallo, T. Urruty, P. Bourdon, and C. Fernandez-Maloigne, "Robust forgery detection for compressed images using CNN supervision," *Forensic Science International: Reports*, vol. 2, Dec. 2020, doi: 10.1016/j.fsir.2020.100112.
[33] X. Wang, H. Wang, S. Niu, and J. Zhang, "Detection and localization of image forgeries using improved mask regional convolutional neural network," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 4581–4593, 2019, doi: 10.3934/mbe.2019229.
[34] M. A. Elaskily *et al.*, "A novel deep learning framework for copy-moveforgery detection in images," *Multimedia Tools and Applications*, vol. 79, no. 27–28, pp. 19167–19192, Jul. 2020, doi: 10.1007/s11042-020-08751-7.
[35] H. Yao, M. Xu, T. Qiao, Y. Wu, and N. Zheng, "Image forgery detection and localization via a reliability fusion map," *Sensors*, vol. 20, no. 22, Nov. 2020, doi: 10.3390/s20226668.
[36] T. M. Mohammed *et al.*, "Boosting image forgery detection using resampling features and copy-move analysis," *arXiv:1802.03154*, Feb. 2018.
[37] B. Bayar and M. C. Stamm, "On the robustness of constrained convolutional neural networks to JPEG post-compression for image resampling detection," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2152–2156, doi: 10.1109/ICASSP.2017.7952537.
[38] D.-Y. Huang, T.-W. Lin, W.-C. Hu, and C.-H. Chou, "Boosting scheme for detecting region duplication forgery in digital images," in *Advances in Intelligent Systems and Computing*, 2014, pp. 125–133.
[39] Jian Li, Xiaolong Li, Bin Yang, and Xingming Sun, "Segmentationbased image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015, doi: 10.1109/TIFS.2014.2381872.
[40] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 6818–6828, doi: 10.1109/ICCV.2019.00692.

BIOGRAPHIES OF AUTHORS

Manjunatha Shivanandappa is a Research Scholar at JSSATE Research Centre, Dept. of CSE, JSSATE, affiliated to VTU Belagavi. He completed M. Tech Computer Science and Engineering from NMAMIT Nitte Mangalore, affiliated to VTU Belagavi, Karnataka. Now he is working as an Associate Professor Department of ISE, Global Academy of Technology, Bengaluru. He can be contacted at email: manjunaths@gat.ac.in.

Malini M. Patil is presently working as an Associate Professor in the Department of Information Science and Engineering at J.S.S. Academy of Technical Education, Bangalore, Karnataka, India. She received her Ph.D. Degree from Bharathiar University in the year 2015. Her research interests are big data analytics, bioinformatics, cloud computing, image processing. She has published more than 20 research papers in many reputed international journals. She is a member of IEEE, IEI, ISTE, and CSI. She has attended and presented papers in many international conferences in India and Abroad. She can be contacted at email: drmalinimpatil@gmail.com.