Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777618

COPY-MOVE FORGERY DETECTION USING DEEP CONVOLUTIONAL NEURAL-NETWORK FEATURES WITH MACHINE LEARNING-BASED CLASSIFIER

Kaleemur rehman, Saiful Islam
ZHCET, AMU Aligarh UP 202001,India
ZHCET, AMU Aligarh UP 202001,India

Abstract: An image forgery detection method detects and locates forged components from manipulated images. Identifying whether an image is forged or non-forged requires a sufficient number of features to detect manipulation or tampering. The patch descriptor extracts efficient and highly effective in-depth features from images using a pre-trained convolutional neural network (CNN). An eventual discriminative feature for SVM classification is attained through a feature fusion technique. We compare our outcome with existing state-of-the-art techniques using publicly available benchmark images from CASIA v2.0. The experiment result demonstrations that the proposed approach using a pre-trained CNN model-based features with Support Vector Machine (SVM) classifier has achieved 98.91% accuracy. It is clearly shows from that the proposed model is both effective and adaptable.

1. Introduction

With the rapid advances and availabilities of powerful image processing software, digital images are becoming increasingly undependable. When digital images are used in a sensitive matter, such as evidence in a court of law, one major problem arises the authenticity of the images. The means for establishing image authenticity and discovery of tampering are treated by image forensics. There are two main categories for digital image authentication: active detection techniques and passive detection techniques. The active methods use a digital signature (Yuan et al., 2016) or watermarking (Xia et al., 2016) embedded in the image before the analysis stage.

Forgery in the digital world is now common in images and other media like video, audio and painting, as well as antique images and paintings. Image forgery is the most commonly utilized, as any person can easily do it using image editing tools. By this, the image will be so well manipulated that the human eye cannot differentiate between the original and tampered image. In this, a part of the original image is copied and pasted somewhere else in the same image to hide something or alter something essential. It is tough to detect because of its similar characteristics (Garg & Saini, 2017). The copy-move digital image forgery means taking some parts of an image and then imposing it on a different segment of the same image to cover certain unwanted parts or elements (D. G. Lowe, 2004). This type of forgery is renowned as its detection is very complicated. This kind of forgery is challenging to detect as features of the image, such as colour, noise, etc., are the same, as well as the source and also the receive side of the image forged are also same (Bayram et al., 2005; Y. Li & Wang, 2012; Mahdian & Saic, 2007). A digital image tampering example can be seen in figure 1. It shows four missiles appearing to take-off from a desert launch

pad as published on the Iranian revolutionary guard website in the right image. Figure 1 shows the left image of three missiles launching. According to analysts, there were three missiles launched. An image tampered with appears closely replicated in its marked regions (Soni et al., 2018).

Figure 1: Showing Original image and tampered image respectively.

Using automatic feature extraction in forgery detection is essential because it can help detect forgery more efficiently. A broad range of forgery research fields can benefit from machine learning (ML) as well as deep learning (DL) algorithms. Color illumination, semantic segmentation and deep convolution neural networks (CNN), are the primary methods Deep Learning (DL) uses to identify and localize splicing image forgeries. According to recent research, image forgery has been remarkably effective with CNNs (Asghar et al., 2017; Huo & Zhu, 2019; Wu et al., 2018a). As a result, we propose a CNN that can handle and detect copy-move forgeries end-to-end in this paper. This CNN includes five main layers: input, CL, FCL, classification layers, and output layers.

In contrast to other deep learning techniques, patch extraction-based CNNs do not require a lot of hardware resources. SVM-based classifiers are used in the classification work. There are three contributions as follows:

1. This paper proposes a technique to detect copy-move forgery (CMFD) feature automatically by using the Convolutional Neural Network (CNN) model.
2. Binary classification (authentic/forged) is performed with the help of the SVM classifier.
3. Comparison with existing state-of-the-art techniques can be used to evaluate the proposed model.

1.1 Paper Organization

This paper is organized as follows. Literature reviews are designed according to the latest research and presented in an in-depth survey in tabular form for better understanding and interpretation in Part II. The proposed methodology is explained in the part III. Part IV provides detailed information about the dataset characteristic used in this paper and the analysis of the results with a graphical representation of the proposed model. In the last the conclusion is presented in part V.

2. Literature Review

It is commonly used to identify positive loop closures by using floating-point features like Scale-Invariant Feature Transforms (SIFT) (G. Lowe, 2004), Oriented FAST and Rotated

BRIEF (ORB) (Rublee et al., 2011), and Speeded Up Robust Features (SURF)) (Bay et al., 2006), or binary features like Binary Robust Independent Elementary Features (BRIEF) (Calonder et al., 2010). To further improve the performance, it can be used to extract local features based on pixel-scale level as well as the larger spatial context.

Haar wavelet coefficients are a sequential arrangement of square-shaped functions utilized within the Fourier analysis for identifying regions related to cyber forensics in video archives (Gouri & Balan, 2017). The author (Dixit et al., 2017) presents a new scheme for detecting copy-move forgeries using stationary wavelet transforms (SWT). The SWT is shift-invariant, unlike most wavelet transforms (e.g. discrete wavelet transform (DWT)), and it helps locate the similarity, i.e. matches, between blocks of an image and the dissimilarity, i.e. noise, between blocks caused by blurring. SVD is used to extract features from images to represent the blocks. In addition, blur invariance is achieved through colour-based segmentation in this work.

The author (Yang et al., 2018) integrates the block-based and keypoint-based methods for detection of copy-move forgeries into a new multi-granularity super-pixels matching technique. The research uses CNN to identify copy-move forgeries and image splicing (Rao & Ni, 2016). The CNN network had been pre-trained on labelled images to extract features from patches. After the features have been disengaged, the SVM classification model is trained.

The research work (Wu et al., 2018b) uses CNN and a deconvolutional network for copy-move forgeries detection (CMFD). Each block of the test image is split up into features, which then extracted using CNN. A self-correlation analysis is then conducted between these blocks. After localizing the matched points between blocks, a deconvolutional model recreates the forgery mask. The CMFD can more robustly detect copy-move forgeries after post-processing procedures, such as affine transformations and JPEG compressions. Table 1 shows the various researchers with their observations and findings.

Table 1: Comparison of various CMFD methods with their performance:

Author & Year	Method	Observations	Accuracy/ Precision
(J. Li et al., 2014)	Segmentation based method	High time complexity, but effective in detecting most CMF attacks	average precision 86%
(Sridevi et al., 2012)	parallel block matching algorithm	With the reduction in execution time, it is effective in a variety of CMF attacks	Not available
(Amerini et al., 2011)	Scale Invariant Features Transform (SIFT) based detection method/ g2NN	Effective for the detection of multiple clones in an image.	TPR of 93%
(Bianchi & Piva, 2012)	DCT coefficients	Image with double JPEG compression is not good	94.5%
(Cozzolino et al., 2015)	Algorithms for Patch Match and Fast Nearest Neighbor Search	Geometric distortions of different types are robustly handled with a reduced time complexity.	Not available

(Bappy et al., 2019)	Long-Short Term Memory (LSTM) cells with encoder-decoder networks for segmenting images	Classification the manipulated or non-manipulated regions effectively.	Average precision 93.4%
(Wu et al., 2018a)	DNN and BusterNet architecture,	It provides localization of sources and targets and is robust against various attacks	CoMoFoD-80.49 % and CASIA dataset76%.
(Huang & Ciou, 2019)	discrete cosine transform (DCT) and transfer vectors	locate forged region of the image with high precision, and less computational complexity	Average Precision 93%
(Armas Vega et al., 2021)	Superpixel segmentation, feature extraction with SIFT, and Helmert transformation	Detects symmetric, recurring targets with poor accuracy but is robust to JPEG compression	Average precision 98%
(Park & Choeh, 2018)	key points extraction with SIFT Algorithm	It can rotate, scale, add white Gaussian noise, etc.	80%

Detecting median filtering from a compressed and small image is very difficult. The author (Chen et al., 2015) suggested a CNN-based technique to extract median filtering residuals from an image. The first layer in CNN is a filter that decreases interference caused by textures and edges. Models can investigate median filter traces after interference has been removed. 15352 images were used to test the approach, comprising five datasets.

Based on recent developments in computer vision, the author (Tyagi & Yadav, 2023) proposed ForensicNet, a convolutional neural network (CNN). The author uses CNNs to mix information in the spatial dimension using depth-wise convolutions, inverted bottlenecks, and separate down-sampling layers. It improves accuracy and reduces network parameters/FLOPs by inverting bottlenecks. There are separate layers for downsampling for the network converges. A normalization layer can also stabilize training when the spatial resolution changes. A depth-wise convolution has the same number of groups and channels as a grouped convolution. ForensicNet shows a large gap of improvement differentiated with the existing experiment methods.

3. Methodology

The proposed model is designed based on the CNN architecture with SVM classification. In-depth features are extracted from images using CNN architecture. In SVM classifier, the final discriminative features are obtained by feature fusion. In the proposed model, CNN architecture consists of 10 layers, such as 8 convolutions and 2 max pooling layers. The input patches of the images, the fully connected layer and softmax layer, are used in the training phase, and the same input patches, SVM, are used in the testing phases. This pipeline has three layers: convolution, batch normalization, as well as rectified linear unit (ReLU) with a fully

connected layer. The feature representation is then fed into an SVM classifier that predicts whether the features correspond to an original or tampered image. The following sections describe the CNN network architecture and the method of training CNNs and SVMs.

3.1 Convolutional neural network

The convolutional layer (CL) is utilized as a feature extractor that learns the representation of features. CNN uses this image as input. The pooling layer reduces the resultant map of the convolution layer and prevents it from overfitting. A fully connected layer is composed of a stack of convolutional layer and pooling layers that are used to extract an abstract feature representation. Pooling is decoding multiple images into one unit as part of an aggregation. In the flattened layer, the feature map is flattened into the feature vector and then passed to the FCL after training from a pre-processed model with fine-tuning. Pattern recognition is performed with the fully connected layer, and the probabilistic conversion of the feature vector is performed with the Softmax activation function.

There is a process of convolving the dimension of the input image $w \times l \times c$ from a hidden layer h^{n-1} with k changed kernels dimension $s \times s \times c$ where w and l are the width and height of the (RGB) input image, respectively, c is the total feature maps in the hidden layer h^{n-1} and s is the filter size. The hidden layer $h(n)$ of an input image contains k feature maps. There is an overlying distance called stride between the $s \times s$ local regions of an image, known as receptive fields. The input and output of convolution layers are arrays called feature maps. The $C_n(X)$ is the feature map for X input data in the convolution layer n, with the kernel and bias well-defined by W^n and B^n respectively. The formula for computing the convolutional layer is as follows [9]:

$$C_n(X) = pooling\ (f^n(F^{n-1}(X) * W^n + B^n))$$

Nonlinear mappings often follow convolutional layers in activation layers. Activation layers apply a nonlinear function to every pixel in an image. The proposed work used ReLU, i.e., $f^n(x) = max(0, x)$. Author (Krizhevsky et al., 2017) demonstrated that CNNs trained by neural activations with ReLU were numerous times faster than CNNs trained by other activation functions in practice. The $pooling(.)$ operation decreases the dimensions of data with the help of mean and max operation.

The architecture implemented in this study is a CNN comprising nine convolutional and two max-pooling layers, as presented in Figure 2. The input size of the network is a $128x128x3$ patch, where 3 represents the RGB color channels. The first two convolutions have a $5x5$ kernel size and output 3 with 30 kernels, respectively. A pooling operation follows these layers with a $2x2$ filter. The subsequent eight layers have 16 kernels, with a $3x3$ kernel size for the convolutions and a $2x2$ filter for the max pooling. These seven convolutions have stride one, whereas the pooling operation has stride two.

Additionally, every convolutional layer uses the ReLU activation function. It has to be mentioned that local response normalization is applied to every feature map before the pooling operation to improve generalization. In particular, the essential value in each neighbourhood is standardized by the values of its three neighbouring channels.

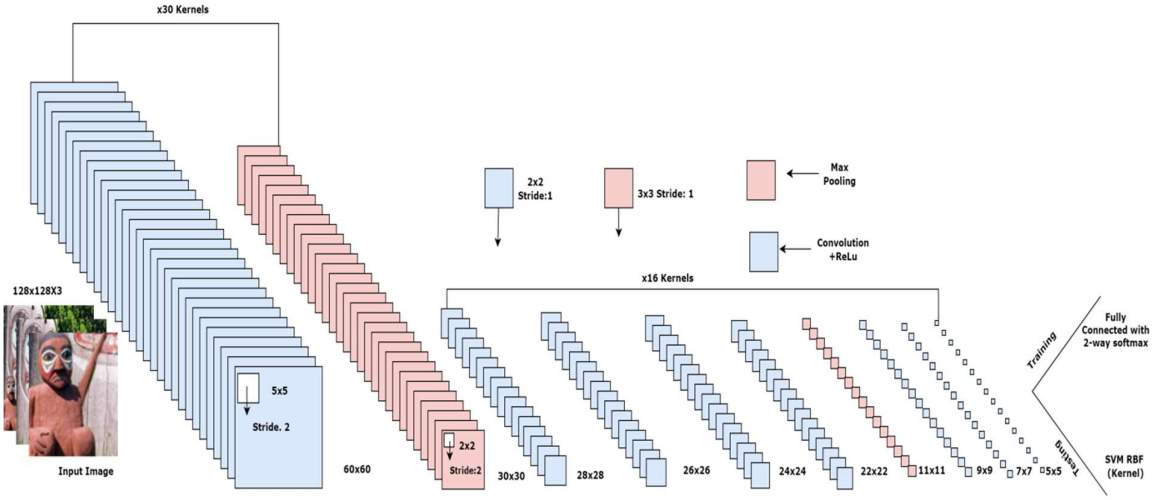

Figure 2: The proposed CNN model

CNN Training: The pre-trained CNN acquires a feature mapping f which transforms an input patch $X \in R^{\mathbb{P}\times \mathbb{P}}$ (where patch of size $\mathbb{P} \times \mathbb{P}$) to a considerable summarized representation $Y = f(X) \in R^{j}$ features assigns local descriptors for input patches based on - J-dimension. A dataset is used to extract image patches that concentrate on local regions of the artefacts and learn to identify them. A total of $128x128x3$ patches were extracted. An eight-stride patched-size sliding window was applied to the whole image for the extraction process. In the next step, we separate tampered patches from non-tampered patches. To determine which patches contain tampered regions, we compare them with the equivalent patches in the mask of this image, as shown in Figure 3. Moreover, we only keep two random tampered patches per image, as training the CNN with many extracted patches would be computationally expensive. Regarding the non-tampered patches, we apply the same technique now on the equivalent authentic image and randomly select two patches.

Figure 3: Example of patch extraction.

Finally, to improve the generalization ability of CNN and avoid overfitting, we augment the patches extracted by rotating them four times by a step of 90 degrees. These patches are fed into the CNN, which extracts a 400-D (5x5x16) feature representation of the patches. These features are then passed to a FCL with a 2-way softmax classifier that uses dropout (Krizhevsky et al., 2017). More specifically, the neurons of the FCL are set to zero with a probability of 50%. Similarly to (Rao & Ni, 2016), we used only one fully-connected layer to reduce the parameters.

SVM Classification: Our proposed model used a SVM classifier to predict and classify scores. The results indicate accuracy and training loss over the number of epochs. The CNN architecture detects unseen forgeries by extracting features from various convolution layers and classifying the label of the tampered image with high accuracy using SVM classifiers. The forged and pristine parts of the image were finally classified using an SVM classifier. Misclassified data are generated by SVM classification as well as a confusion matrix. A CNN network is trained first, followed by an SVM classifier. Every possible ₱ × ₱ patch was first extracted from the original and forged images by using a sliding window and scanning the whole image using a stride of s for the classification of the images. A CNN is used to generate feature representations Y_i $(400 - D)$ resulting from n new patches per image. The SVM needs these representations merged into a single representation ¥ $[j]$ before it can be used. The mean or max pooling is applied across all the dimensions of Y_i all, the n patch extracted:

$$¥\,[j] = Mean\ or\ Max\{Y_1[j] \ldots\ldots Y_n[j]\}$$

Where $j \in [1,\ 400]$ dimensions. The SVM uses the resulting $400 - D$ feature vector to determine whether the image is original or tampered with. The ten-fold cross-validation method was employed to divide a dataset into training and testing sets, the most commonly used and most suitable method. Each fold in the dataset was divided into ten equal folds, each with a similar ratio of forged and real images. Training and validation were conducted with nine folds each time, and testing was done with onefold.

4. Experimental Analysis and Discussion

The proposed work was conducted on a system equipped with an Intel core i7, 4 GB of RAM, and a 64-bit processor running Windows 10. PyTorch2, scikit learn, and PyTorch 3.5 implements the proposed technique. On the basis of accuracy and training loss, a comprehensive evaluation of the proposed technique is provided. Furthermore, the results obtained are compared with the existing techniques. The accuracy is measured as the percent ratio of the images that are accurately classified, and the following equation calculates it:

$$Accuracy\ (\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

TP (True Positive) and FN (False Negative) are defined as the number of tampered images labelled as tampered and original, respectively. TN (True Negative) and FP (False Positive) are the numbers of original images labelled as original and tampered, respectively.

The proposed model is evaluated on the publicly available image CASIA v2.0 datasets (Dong & Wang, 2011). It is divided into two parts one is authenticated, and another is a tamper. Figure 4 shows a few images from the CASIA v2.0 dataset. In the proposed approach whole dataset is taken for the experiment purpose. The databases are described in detail in Table 2.

Figure 4: Sample of the images from the CASIA v2.0 dataset.

Table 2: Overview of CASIA v2.0 dataset

Database	Size	Components	Format
CAISA V2.0	7200 authentic and 5123 tampered	From 320 X 240 to 800 X 600 color image	JPEG, BMP, TIFF

The patches that extract the image features are obtained using a stride s of 128 for CASIA v2.0. The proposed model selected a larger stride to maintain the same number of patches per image. All the CNNs are trained for 250 epochs using the cross-entropy loss and optimizing the network via Stochastic Gradient Descent (SGD). These parameters were selected for every CNN trained to improve its convergence. An initial learning rate of 0.001 and a batch size of 200 images were used in the proposed model to train the CNN on the CASIA v2.0 dataset. The SVM classified used the RBF kernel for every run and optimized the γ and C accordingly, performing an exhaustive grid search with the values 0.001 and 1. Secondly, the same tests use augmented datasets, where the images are rotated four times by a step of 90 degrees with a batch size of 128 images. Augmentation improved the classification accuracy of the datasets, as mentioned in figure 5. In the CASIA v2.0 dataset, classification accuracy is 98.91% with augmentation and 93.57% without augmentation. The accuracy increased rapidly just after the 50 epochs in the augmented datasets, but without augmentation; the dataset has a similar pattern from the 50 epoch to 250 epochs

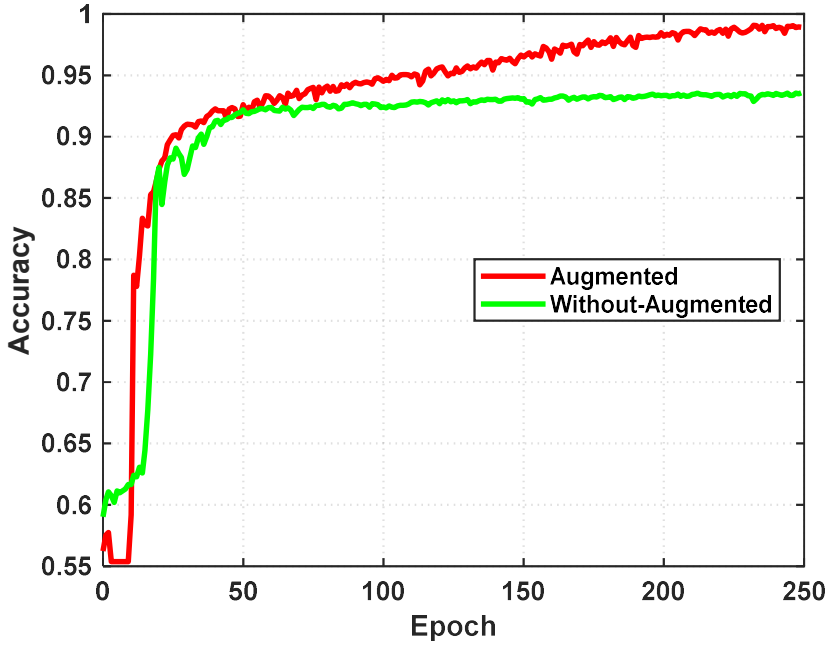

Figure 5: Comparative analysis of the accuracy of the proposed model with and without augmented data on the CASIA v2.0 dataset.

The SVM classification accuracy on the CASIA v2.0 dataset after the 10-fold cross-validation is mention in the Table 3. It shows that the augmented classification accuracy is higher than the without augmented dataset. All three combination of accuracy is measured for the comparative analysis between with and without the augmentation process. The graphical representation of the mean and standard deviation accuracy is presented in figure 6.

Table 3: Accuracy analysis of CASIA v2.0 dataset

Batch Size	Accuracy	Mean	Std.
128	98.91%	93.39%	8.75%
200	93.57%	90.25%	7.98%

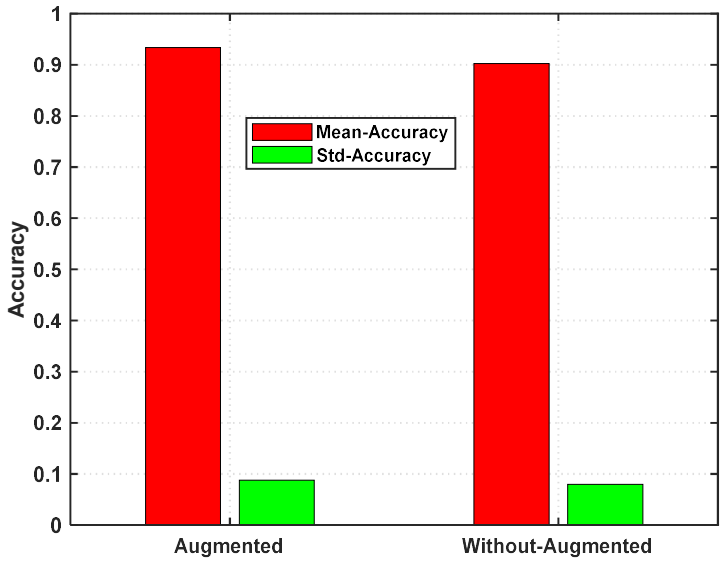

Figure 6: Comparative accuracy analysis of the proposed model's mean and standard deviation with and without augmented data on the CASIA v2.0 dataset.

Moreover, the corresponding confusion matrix was computed using a random 80-20 split and can be found in Table 4. In particular, the SVM correctly classified 1008 tampered and 1426 original images, while it misclassified only 17 tampered images and 72 original images from the augmented data. Similarly, SVM correctly classified 1,013 tampered and 1,308 original images while misclassifying only 12 tampered images and 190 original images without data augmentation. Moreover, comparing the classification with and without rotations makes it apparent that augmenting the data boosts the system's performance irrespective of the dataset used.

Table 4: Confusion matrix of proposed model on CASIA v2.0 dataset

CASIA2	Data Augmentation		Without-Augmentation	
	Predicted Authentic	Predicted Tampered	Predicted Authentic	Predicted Tampered
Actual Authentic	1426	72	1308	190
Actual Tampered	17	1008	12	1013

The training loss with and without the augmented data is depicted in Figure 7. The training loss continuously decreases with the augmented data, but without augmented data, have a constant pattern. The CASIA v2.0 has decreased pattern with the data augmentation from 50 to 250 epochs sharply. Regarding the networks with the augmented data, the loss for CASIA v2.0 after 250 epochs is much lower (0.0109) than without data augmentation (0.064). In particular, the loss of CASIA v2.0 rapidly decreases in the first epochs, and then it keeps decreasing at a lower rate until it becomes stable in the last epochs, as shown in figure 8.

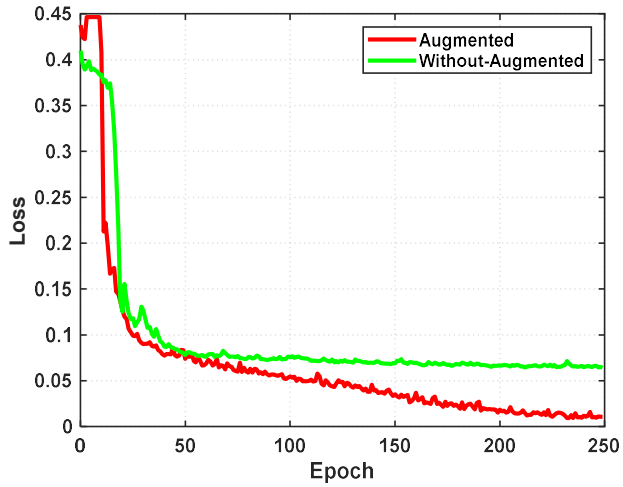

Figure 7: Comparative analysis of training loss of the proposed model with and without augmented data on the CASIA v2.0 dataset.

The evaluation of the proposed model is compared with the existing model of the image splicing forgery detection and classification on the CASIA v2.0 dataset. Figure 8 demonstrates the accuracy (%) performance of the proposed model compared with the Markov Features+DCT+DWT (He et al., 2012), SPT+LBP (Muhammad et al., 2014), CNN+SRM (Rao & Ni, 2016), DWT+LBP (Kaur & Gupta, 2016), Markov Features+QDCT (C. Li et al., 2017), Deep Learning (Zhang et al., 2016), GLRLM Texture Features (Mushtaq & Mir, 2014), Markov Feature (He et al., 2012) and Deep Learning+HWT (Abd El-Latif et al., 2019), existing techniques. The proposed model achieves 98.9 % accuracy as compared to the existing techniques.

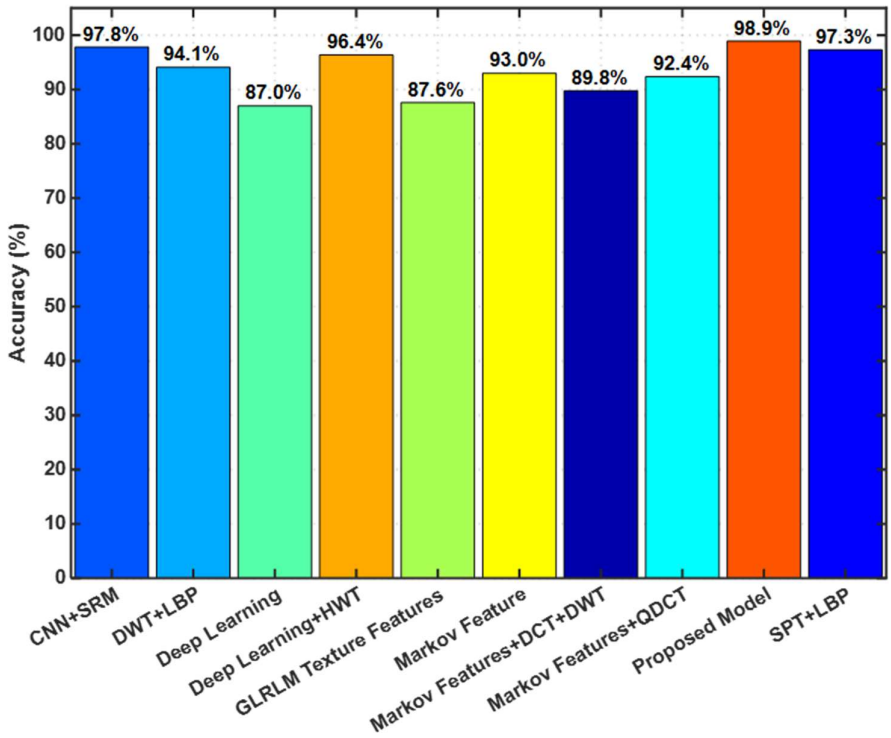

Figure 5: The accuracy of the proposed model comparison with existing models on the dataset CASIA v2.0.

5. Conclusion

In current scenario, no trustworthy, commercial, real-world applications offer a viable solution to copy-move forgery in digital images. Nevertheless, few applications provide limited solutions for image forgery, such as forensic beta, which works as a microscope to reveal hidden details in an image. Metadata can be investigated using the MagNET forensic application. This study proposes an image analysis method based on CNN-SVM for detecting copy-move forgeries. This model utilizes a CNN model as a local patch descriptor, which is pre-trained using labelled patch samples drawn individually for tampered images. To determine the final discriminative features for SVM classification a feature fusion technique is incorporated with the pre-trained CNN.

References

1. Abd El-Latif, E. I., Taha, A., & Zayed, H. H. (2019). A Passive Approach for Detecting Image Splicing using Deep Learning and Haar Wavelet Transform. *International Journal of Computer Network and Information Security*, *10*(5), 28.
2. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011). A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, *6*(3), Article 3.
3. Armas Vega, E. A., González Fernández, E., Sandoval Orozco, A. L., & García Villalba, L. J. (2021). Copy-move forgery detection technique based on discrete cosine transform blocks features. *Neural Computing and Applications*, *33*, 4713–4727.
4. Asghar, K., Habib, Z., & Hussain, M. (2017). Copy-move and splicing image forgery detection and localization techniques: A review. *Australian Journal of Forensic Sciences*, *49*(3), Article 3.
5. Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B. S., & Roy-Chowdhury, A. K. (2019). Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, *28*(7), 3286–3300.
6. Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. *European Conference on Computer Vision*, 404–417.
7. Bayram, S., Avcıbaş, İ., Sankur, B., & Memon, N. (2005). Image manipulation detection with binary similarity measures. *2005 13th European Signal Processing Conference*, 1–4.
8. Bianchi, T., & Piva, A. (2012). Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, *7*(3), 1003–1017.
9. Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, 778–792.
10. Chen, J., Kang, X., Liu, Y., & Wang, Z. J. (2015). Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, *22*(11), 1849–1853.
11. Cozzolino, D., Poggi, G., & Verdoliva, L. (2015). Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, *10*(11), 2284–2297.

12. Dixit, R., Naskar, R., & Mishra, S. (2017). Blur-invariant copy-move forgery detection technique with improved detection accuracy utilizing SWT-SVD. *IET Image Processing*, *11*(5), Article 5.
13. Dong, J., & Wang, W. (2011). *CASIA tampered image detection evaluation database*.
14. Garg, T., & Saini, H. (2017). A review on various techniques of image forgery detection. *Int J Eng Technol Sci Res*, *4*(4), 490–493.
15. Gouri, M. S., & Balan, R. S. (2017). Enhancement of multimedia security using random permutation with wavelet function. *Computers & Electrical Engineering*, *63*, 41–52.
16. He, Z., Lu, W., Sun, W., & Huang, J. (2012). Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recognition*, *45*(12), Article 12.
17. Huang, H.-Y., & Ciou, A.-J. (2019). Copy-move forgery detection for image forensics using the superpixel segmentation and the Helmert transformation. *EURASIP Journal on Image and Video Processing*, *2019*(1), 1–16.
18. Huo, Y., & Zhu, X. (2019). High dynamic range image forensics using cnn. *ArXiv Preprint ArXiv:1902.10938*.
19. Kaur, M., & Gupta, S. (2016). A passive blind approach for image splicing detection based on DWT and LBP histograms. *International Symposium on Security in Computing and Communication*, 318–327.
20. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386
21. Li, C., Ma, Q., Xiao, L., Li, M., & Zhang, A. (2017). Image splicing detection based on Markov features in QDCT domain. *Neurocomputing*, *228*, 29–36.
22. Li, J., Li, X., Yang, B., & Sun, X. (2014). Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security*, *10*(3), 507–518.
23. Li, Y., & Wang, H. (2012). An efficient and robust method for detecting region duplication forgery based on non-parametric local transforms. *2012 5th International Congress on Image and Signal Processing*, 567–571.
24. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.
25. Lowe, G. (2004). Sift-the scale invariant feature transform. *Int. J*, *2*(91–110), 2.
26. Mahdian, B., & Saic, S. (2007). Detection of copy–move forgery using a method based on blur moment invariants. *Forensic Science International*, *171*(2–3), 180–189.
27. Muhammad, G., Al-Hammadi, M. H., Hussain, M., & Bebis, G. (2014). Image forgery detection using steerable pyramid transform and local binary pattern. *Machine Vision and Applications*, *25*, 985–995.
28. Mushtaq, S., & Mir, A. H. (2014). Novel method for image splicing detection. *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2398–2403.
29. Park, C.-S., & Choeh, J. Y. (2018). Fast and robust copy-move forgery detection based on scale-space representation. *Multimedia Tools and Applications*, *77*(13), Article 13.

30. Rao, Y., & Ni, J. (2016). A deep learning approach to detection of splicing and copy-move forgeries in images. *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6.
31. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571.
32. Soni, B., Das, P. K., & Thounaojam, D. M. (2018). CMFD: a detailed review of block based and key feature based techniques in image copy-move forgery detection. *IET Image Processing*, *12*(2), 167–178.
33. Sridevi, M., Mala, C., & Sandeep, S. (2012). Copy–move image forgery detection in a parallel environment. *Computer Science & Information Technology, CS&IT*, 19–29.
34. Tyagi, S., & Yadav, D. (2023). ForensicNet: Modern convolutional neural network-based image forgery detection network. *Journal of Forensic Sciences*.
35. Wu, Y., Abd-Almageed, W., & Natarajan, P. (2018a). Busternet: Detecting copy-move image forgery with source/target localization. *Proceedings of the European Conference on Computer Vision (ECCV)*, 168–184.
36. Wu, Y., Abd-Almageed, W., & Natarajan, P. (2018b). Image copy-move forgery detection via an end-to-end deep neural network. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1907–1915.
37. Yang, H., Niu, Y., Jiao, L., Liu, Y., Wang, X., & Zhou, Z. (2018). Robust copy-move forgery detection based on multi-granularity Superpixels matching. *Multimedia Tools and Applications*, *77*(11), Article 11.
38. Zhang, Y., Goh, J., Win, L. L., & Thing, V. L. (2016). Image region forgery detection: A deep learning approach. *SG-CRC*, *2016*, 1–11.