ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2021.0348

TRANSFER LEARNING APPROACH FOR SPLICING AND COPY-MOVE IMAGE TAMPERING DETECTION

Nagaveni K. Hebbar and Ashwini S. Kunte
Department of Electronics and Telecommunication Engineering, Thadomal Shahani Engineering College, India

Abstract

Image authentication before using in any security critical applications has become necessary as the image editing tools are increasing and are handy to use in today's world. Images could be tampered in different ways, but a universal method is required to detect it. Deep learning has gained its importance because of its promising performance in many applications. In this paper a new framework for image tampering detection using Error Level Analysis (ELA) and Convolutional Neural Network (CNN) with transfer learning approach is proposed. In this method, the images are pre-processed using ELA to highlight the tampered region and are used to fine tune the entire model. Six different pre-trained models are used in the proposed framework to compare the performance in classifying the tampered and authentic images. The complexity and processing time of the proposed method is low with respect to most of the existing methods as the images are not divided into patches. The performance of the model obtained is also considerably good with an accuracy of 97.58% with Residual Network 50(ResNet50).

Keywords:

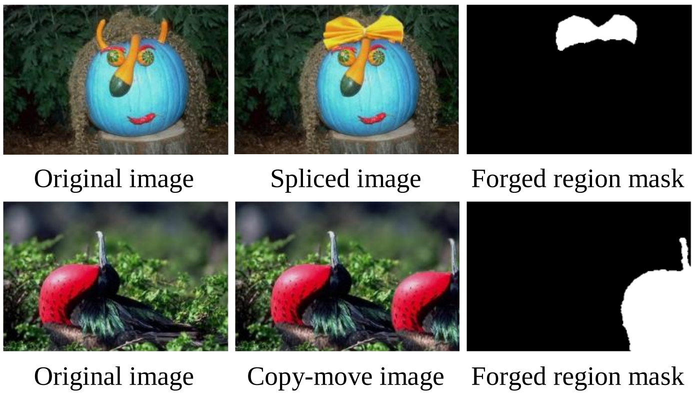
Tampering Detection, Transfer Learning, Copy-Move, Splicing

1. INTRODUCTION

In this digital era, the images have become one of the most important ways of information exchange in different applications like social media, medical, television and many other applications over the internet. With the increase in different type of image editing tools and software that are available in handy devices like mobile and laptops, it has become possible to modify the images easily for different purposes. The images could be modified for some good intention but if the images are modified with some bad intent, then it is called a forgery. The image forgery could be done to conceal some meaningful information like hiding some person or object in the image. The manipulated images are used as false evidence in court, to make money by getting more viewers on social media, getting popularity or publicity, etc. There is a necessity to verify the integrity of the images to prevent spreading or promoting of false information and also to avoid trusting and considering the edited images as evidence in the court of law.

There are different image manipulation types and among them Copy-move and Splicing are the major types as shown in Fig.1. Copy-move forgery [2]-[7] is done by copying a small region of an image and pasting in the same image to change the information conveyed by the image. Splicing forgery [10]-[12] is done by replacing a portion of an image by a part of a different image to manipulate the information conveyed. The tampering is done in such a way that the changes are not easily identified by the naked eyes. During the process, the forged region may be made to undergo some transformations [2] [3] like rotation, scaling, blurring, etc. to match it to the surroundings and to hide from detection. Sometimes the forged images are made to undergo post-processing operation like smoothing to remove the traces that arise at the edges during manipulation of images. Hence, it is a big challenge to detect the forged images as it could be used for unethical purposes.

Many tampering detection methods were proposed to detect either copy-move or splicing detection. Then the researchers thought of universal methods to detect both the forgery types and hence there were several approaches emerged with DCT [13], SURF [14] and LBP [15] features with SVM classification. When CNN emerged in several applications and proved its importance, it was also used in detecting tampered images [16]-[18]. With the advancement in the processing capacity using GPU, the deeper architectures also emerged [24] [25] to detect tampering in images.



Original image Spliced image Forged region mask

Original image Copy-move image Forged region mask

Fig.1. Image tampering types

All the above research work motivated me to propose a new framework for forged image classification problems. As the images are mostly compressed to store and transfer efficiently, ELA [32] is used to highlight the tampered part. As we need a huge dataset to train deep CNN architecture for better accuracy, the transfer learning approach is used in this work to overcome this problem. Transfer learning uses already trained weights in one application over a huge dataset to initialize the network in another application, which considerably reduces the training time and gives better accuracy with smaller datasets. Some existing methods [11] [20] convert images into patches to get better accuracy which increases the complexity and time of training and testing, but the proposed method trains the complete images without creating patches which reduces the complexity and processing time. In the proposed architecture, instead of flattening all the feature maps we used a GAP [31] layer to get an average feature vector. It helps us in reducing a lot of trainable parameters, which further helps to reduce the training time and to avoid overfitting problem.

Further, in this paper the contents are organized as follows. Related work gives the overview of existing techniques in this area. The methodology section explains about the proposed approach in detail step by step. The following section to methodology, the results and discussion of the proposed approach is elaborated. Finally, the conclusion section gives the remarks about the experimentation undertaken and the future work.

2. RELATED WORK

Over several years, many techniques were proposed by researchers to detect tampering in the images, but most of the methods are focused on a particular type of forgery. The authors proposed a copy-move detection and localization method in [3] using a robust clustering approach with j-linkage. Several approaches invariant to various transformations emerged to detect copy-move forgery using Zernike moments [4], SIFT [2] and SURF [5] [6] key points. A segmentation-based approach [7] was used to detect copy-move forgery by creating the image patches and patch matching. A new method was proposed in [8] by using polar cosine transform and in [9] using an expanding block approach to detect copy-move forgery. Authors exploited Markov's features extracted in the DCT domain and more features from the DWT domain in [10] for image splicing detection. When images are spliced, there will be changes in the statistical distributions in the image as the tampered part is from another image. Noise variations in multiple scales are used in [11] for image splicing detection. A new image splicing detection was proposed in [12] by finding local features from co-occurrence of residuals in image and synthetic features were extracted using that. As the method proposed for one type of forgery was not useful in detecting other types of forgery, researchers tried to explore methods to detect multiple forgery types. Authors in [13] used DCT and SURF features and SVM to detect copy-move and splicing detection.

In [14] statistical Gaussian Mixture Model and in [15] combination of DCT and LBP for feature extraction and then the classification is done using SVM in both the approaches.

Before using machine learning approaches for image forgery detection, researchers used to design feature extractors based on edge detection, illumination of light, etc. These techniques exploit visual information present in the image. Since CNN also is based on the visual cortex, it is more powerful in learning complex features and can detect the forged part which may not be detected by naked eyes. In CNN initial layers learn the features like lines, edges and textures. The next few layers learn complex textures and patterns and end layers learn the main features required to classify in an application. Many image tampering detection approaches have emerged using CNN in the past few years as CNN has proved its importance in extracting complex features from the images which helps in detecting forgeries in the images.

In [16], the authors used a new CNN architecture to learn tampered features by suppressing image content to highlight the tampered region. The forged images were detected in [17] by dividing the images into smaller patches and autoencoders were stacked to learn the deep features in each patch. In the method proposed in [18], the weights of the first CNN layer are initialized with filter coefficients used in the Spatial Rich Model (SRM). As the deep CNN has proven its importance in object classification and detection in [19], this approach is used in image forgery classification in [20].

In [21], the authors have used the DCT coefficient quantization effect for image tampering detection and in [22], the authors have used JPEG artifacts to localize the forged region. The experimentation in [23] shows the importance of ELA in image manipulation detection. In the proposed approach ELA is used to process the image to highlight the tampered region. A modified MobileNetV2 architecture is used in [25] for finding forgeries in the images. Authors in [24], have used ELA and VGG16 [28], a pre-trained model to reduce the training time by transferring the already learned features in forgery detection. If the number of layers is increased in the VGG network to improve performance, it results in vanishing gradient problem which is overcome by using deep residual learning network (ResNet) [29], and densely connected convolutional networks (DenseNet) [30]. Hence, we have adopted these networks also in our experimentation.

3. METHODOLOGY

Deep, CNN based techniques [28]-[30] outperformed all other methods in the ImageNet classification problem, which used millions of images to train the network. The pre-trained weights of several models are available on Keras [34] and open to all researchers to +use it in solving any other similar problem. This approach is called transfer learning where knowledge acquired while finding solutions to one problem is used in solving other similar problems.

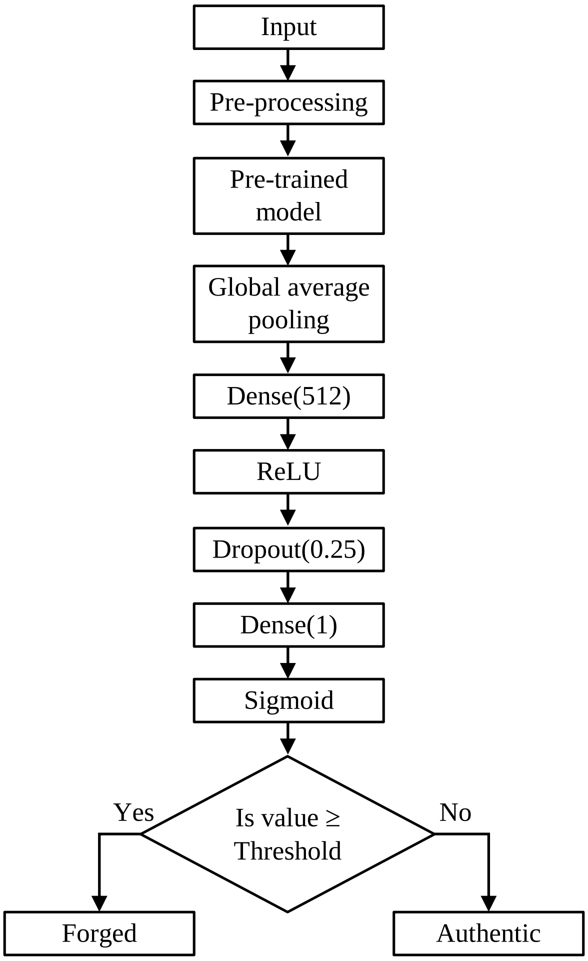


Fig.2. Proposed Framework

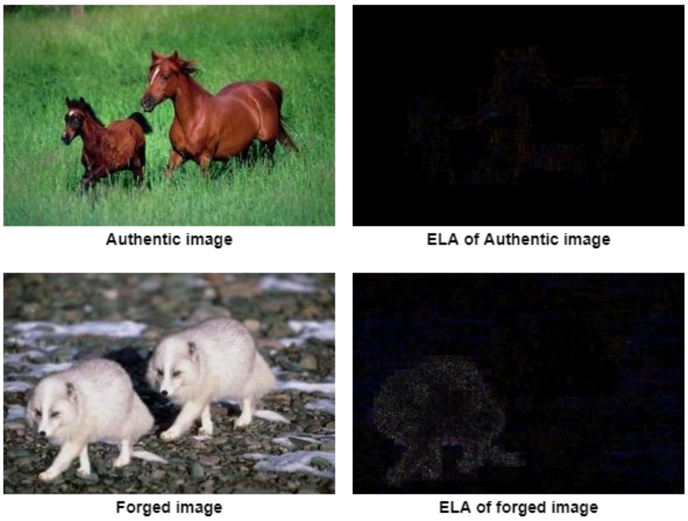
In this paper the transfer learning approach is used to compare the performance of six different pre-trained models in the proposed framework as shown in Fig.2. The classifying layers are modified in the pre-trained models to suit our problem. The model is fine-tuned by training the entire network with pre-processed images of CASIA2.0 [1] dataset.

3.1 INPUT DATASET

In this method CASIA2.0 [1] Image tampering detection evaluation dataset is used that has 7491 authentic and 5123 fake images of splicing and copy-move type with varying image resolution and different file type. This dataset has fake images with geometric operation and post-processing operations like rotation, resizing, distortion, blurring and JPEG compression to hide detection.

3.2 DATA PRE-PROCESSING

Usually, when the images are stored or transmitted, they are compressed. JPEG is a well-known compression technique used, where the image pixels are divided into 8X8 blocks and each block is compressed. When the images are compressed, that leaves compression artifacts which are uniform all over the image. If the image has a tampered part, the compression artifacts are different in the forged part which gets highlighted when it undergoes ELA. The Fig.3 shows the effect of ELA on real and fake images. For calculating ELA, the image is compressed to 90% error quality. Then it is decompressed and the difference is calculated with the original image. To highlight the error levels, the resulting image is enhanced. The resulting images are resized to 256 X 256 and then normalization is applied to the images to make sure that all the images have similar data distribution.



Authentic image

ELA of Authentic image

Forged image

ELA of forged image

Fig.3. ELA of authentic and tampered image

3.3 PRE-TRAINED MODELS

In the proposed architecture, the bottleneck layers of VGG16, VGG19, ResNet50, DenseNet121, DenseNet169 and DenseNet201 pre-trained models are used for experimentation with new classification layers.

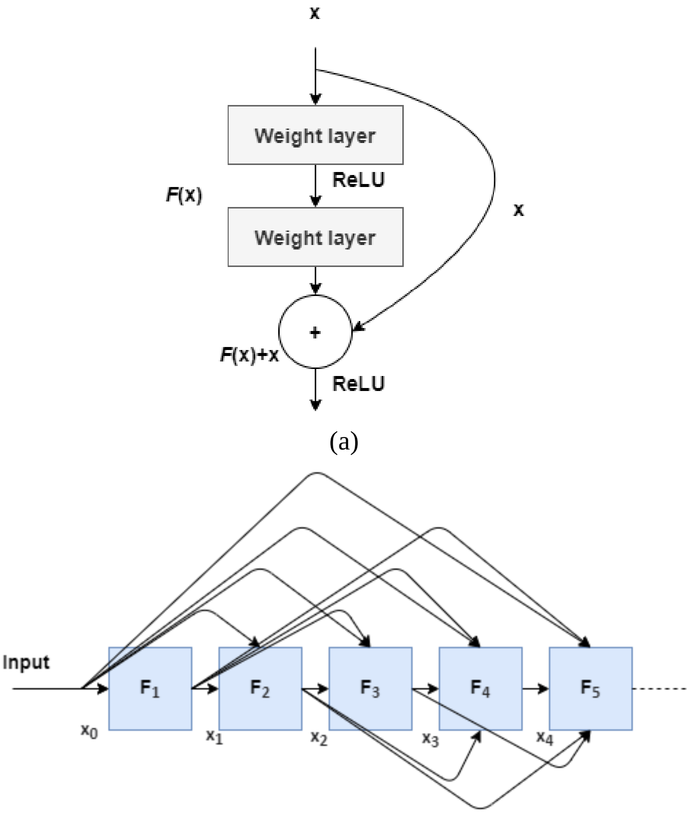
VGG16 [28] is a deep CNN architecture used for ImageNet classification in 2014. This is one of the best models used for image classification even now. In VGG16 input is passed to two convolution layers of 3×3 filters and the same padding. After every convolution layer, the ReLU activation function is applied, which introduces non-linearity.

After every two or three convolution layers, a pooling layer is used with 2×2 pixel window and stride 2, to reduce the spatial dimensions of feature maps. VGG19 [28] architecture has one more convolution layer stack to make the number of layers to 19.

ResNet50 [29] has 50 convolutional layers with skip connections. In the plain sequential network, if the number of layers is increased, the training becomes difficult as it suffers a vanishing gradient problem. This problem results in an increased error rate and the accuracy starts saturating or may degrade also. Residual network solves this problem by using a feed-forward identity connection as shown in Fig.4(a). The feed forward structure results in a compact model by reducing the trainable parameters. A ResNet block is represented as:

$$x_l = F(x_{l-1}) + x_{l-1} \tag{1}$$

where $F(x_{l-1})$ is the convolutional layers and x_{l-1} is the identity connection from previous input.



(a)

(b)

Fig.4. (a) ResNet block (b) DenseNet block with 5 layers

DensenNet121, DendeNet169 and DenseNet201 [30] are also trained with the ImageNet dataset. Each layer in DenseNet gets feed-forward input of feature maps from all previous layers which in turn is passed on to all subsequent layers. Each layer receives the collective knowledge from all preceding layers as shown in Fig.4(b), which makes the network compact but the features obtained are complex. Every Dense block performs batch normalization, ReLU activation, and 3×3 convolution. After a few dense blocks, a transition block is used which includes batch normalization, 1×1 convolution, and an average pooling layer. A DenseNet block is represented as:

$$x_l = F_l([x_0, x_1, x_2, \ldots, x_{l-1}]) \qquad (2)$$

where x_l is the l^{th} layer gets the feature maps x_0, x_1, x_2,…, x_{l-1} as input from all previous layers and F_l is the l^{th} dense block.

All the pre-trained models were imported using TensorFlow and the bottleneck layers were used in forged image classification with some modification to the classification layers and fine-tuning with the dataset.

3.4 CLASSIFICATION

In the proposed model, the features of the pre-trained model pass through the GAP layer, which generates an average over all the features and gives the optimized feature vectors. After the GAP layer, a dense layer is used with 512 features, which is followed by a dropout with a factor of 0.25. The dropout helps to eliminate the problem of overfitting by deactivating some neurons, which forces the layer to some new representation. Then a dense layer is used along with the sigmoid activation function to predict if the image is forged or authentic.

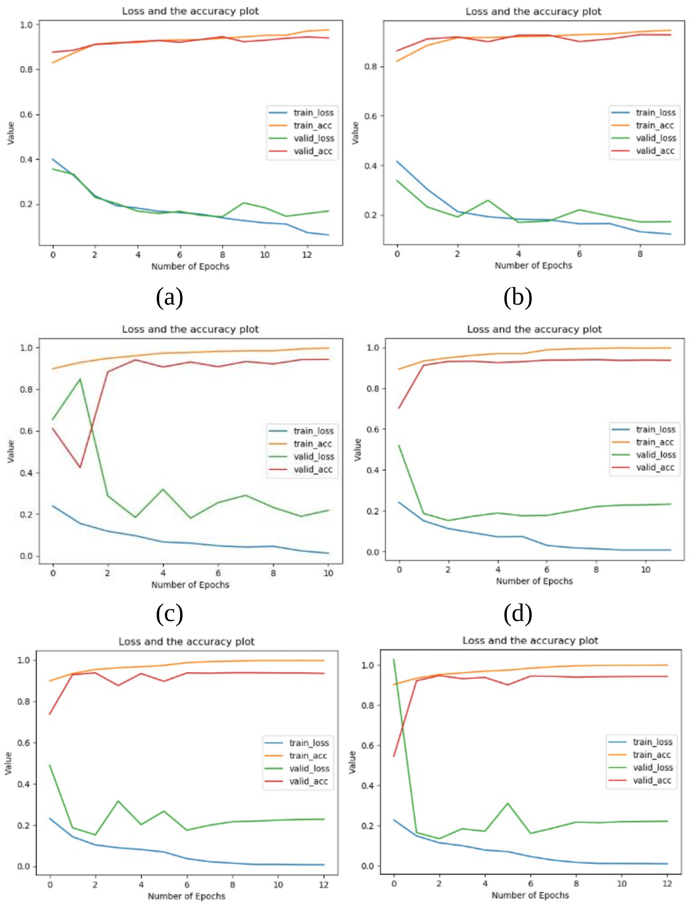
4. EXPERIMENT AND RESULTS

We have implemented our method using Python3.8 with TensorFlow2.2 [33] framework. The model is trained on a computer system with Intel i5-7400, 16 GB of RAM and a GTX 1060 6GB GPU. The details of pre-trained models are given in Table.1 in terms of model size, number of parameters and number of convolution layers. CASIA2.0 Image Tampering Detection Evaluation Dataset was used as this dataset has both copy-move and splicing forged images with different transformation and post-processing. For the training and testing purpose, we have split the dataset, where 80% of the images (real and fake), were used for training the model. The remaining 20% of the images in the dataset were used for validation and testing of the model. All the images were pre-processed using the ELA to highlight forged parts, which are then passed as the input to the model. For each epoch, the model takes around 20 minutes that can be reduced using better system configuration.

Table.1. Comparison of pre-trained models [34]

Pre-trained Models	Model Size	Convolutional Layers	Parameters
VGG16	528MB	16	138,357,544
VGG19	549MB	19	143,667,240
ResNet50	98MB	50	25,636,712
DenseNet121	33MB	121	8,062,504
DenseNet169	57MB	169	14,307,880
DenseNet201	80MB	201	20,242,984

The hyperparameters were chosen after some initial experiments with the dataset. We have used binary cross-entropy as the loss function with Adam optimizer and 1e^{-4} as the initial learning rate. We have used the ReduceLROnPlateau callback to reduce the learning rate when the loss stops improving.



(a) (b)

(c) (d)

(e) (f)

Fig.5. Training loss and training accuracy with respect to number of epochs while finetuning the models: (a) VGG16, (b)VGG19, (c) ResNet50, (d) DenseNet121, (e) DenseNet169, (f) DenseNet210

All the images were resized to 256×256 and were given to the model in a batch of 16 to train the model. EarlyStopping callback was also used to stop the training when the loss rate stops improving in consecutive epochs. The graph for training accuracy, validation accuracy, training loss and the validation loss for all the models are shown in Fig.5.

Table.2. Performances of the proposed method with different pre-trained models

Models ↓	Accuracy (%)	Precision	Recall	F1-Score
VGG16	95.903	0.954	0.941	0.947
VGG19	93.85	0.947	0.9	0.923
DenseNet121	95.96	0.972	0.927	0.949
DenseNet169	97.06	0.970	0.954	0.963
DenseNet201	97.42	0.978	0.956	0.967
ResNet50	97.58	0.958	0.979	0.968
Phan-Xaun et al. [25]	95.15	0.973	0.906	0.938

The Table.2 shows an overview of the quantitative results of all the models on the CASIA2.0 test dataset. From the table, we can see that VGG16 performs better than VGG19, which indicates that just increasing the number of layers does not improve the performance. To avoid this issue, we have experimented with the various DenseNet models, where we can see improvement in the accuracy with increased model depth. When all the three DenseNet models are compared with each other, we can see that the DenseNet201 achieves an accuracy of 97.42 % which is 1.46 % higher than DenseNet121. After that, we have experimented with the Resnet50, which gives 0.16% improvement over the DenseNet201. When we compare all the models, we can clearly see that ResNet50 gives the best performance. ResNet and DenseNet in the proposed framework out performs the similar approach proposed in [25] in which the authors have used MobileNet-V2 model. In [25] the images are divided into patches, which increases complexity and the processing time but in the proposed method, since the images are not divided into patches, the model is simple and fast.

5. CONCLUSION

We propose an approach to classify the forged images by transfer learning using pretrained models. We have used the ELA to preprocess the images to highlight the tampered pixels in terms of error level. From the results, we can see that just increasing the depth of the network does not increase the performance, instead the performance decreases. This decrease in performance is due to overfitting of the models. This overfitting is overcome by using the DenseNet and ResNet50, where previous layers feature maps are also used in the next layer. As the network was trained with the whole images without creating patches, the complexity and processing time are reduced when compared with the models that used image patches. The performance was better with the ResNet50 model with accuracy of 97.58% among all six models used. In future the image patches could be used to train the model to improve the performance but that may increase the complexity. The image pre-processing technique could be changed to check the performance of the proposed model.

REFERENCES

[1] J. Dong, W. Wei and T. Tieniu, “Casia Image Tampering Detection Evaluation Database”, *Proceedings of IEEE China Summit and International Conference on Signal and Information Processing*, pp. 1-8, 2013.

[2] I. Amerini, L. Ballan, R. Caldelli, A.D. Bimbo and G. Serra, “A Sift-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery”, *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 3, pp. 1099-1110, 2011.

[3] I. Amerini, L. Ballan, R. Caldelli, D.B. Alberto, D.T. Luca and S. Giuseppe, “Copy-Move Forgery Detection and Localization by Means of Robust Clustering with J-Linkage”, *Signal Processing: Image Communication*, Vol. 28, No. 6, pp. 659-669, 2013.

[4] S.J. Ryu, M.J. Lee and H.K. Lee, “Detection of Copy-Rotate-Move Forgery using Zernike Moments”, *Information Hiding*, Vol. 6387, pp 51-65, 2010.

[5] X. Bo, W. Junwen, L. Guangjie and D. Yuewei, “Image Copy-Move Forgery Detection based on Surf”, *Proceedings of International Conference on Multimedia Information Networking and Security*, pp. 889-892, 2010.

[6] B.L. Shivakumar and S. Baboo, “Detection of Region Duplication Forgery in Digital Images using Surf”, *Proceedings of International Conference on Multimedia Information Networking and Security*, pp. 889-892, 2010.

[7] J. Li, X. Li, B. Yang and X. Sun, “Segmentation-Based Image Copy-Move Forgery Detection Scheme”, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 3, pp. 507-518, 2014.

[8] Y. Li, “Image Copy-Move Forgery Detection based on Polar Cosine Transform and Approximate Nearest Neighbor Searching”, *Forensic Science*, Vol. 224, No. 1-3, pp. 59-67, 2013.

[9] G. Lynch, Y.S. Frank and M.L. Hong Yuan, “An Efficient Expanding Block Algorithm for Image Copy-Move Forgery Detection”, *Information Sciences*, Vol. 239, pp. 253-265, 2013.

[10] Z. He, W. Lu, W. Sun and J. Huang, “Digital Image Splicing Detection based on Markov Features in DCT and DWT Domain”, *Pattern Recognition*, Vol. 45, No. 12, pp. 4292-4299, 2012.

[11] C. Pun, L. Bo and Y. Xiao-Chen, “Multi-Scale Noise Estimation for Image Splicing Forgery Detection”, *Journal of Visual Communication and Image Representation*, Vol. 38, pp. 195-206, 2016.

[12] D. Cozzolino, G. Poggi and L. Verdoliva, “Splicebuster: A New Blind Image Splicing Detector,” *Proceedings of IEEE Workshop on Information Forensics and Security*, pp. 1-6, 2015.

[13] S.D. Lin and W. Tszan, “An Integrated Technique for Splicing and Copy-Move Forgery Image Detection”, *Proceedings of International Congress on Image and Signal Processing*, pp. 1-8, 2011.

[14] W. Fan, W. Kai and C. François, “General-Purpose Image Forensics using Patch Likelihood under Image Statistical Models”, *Proceedings of IEEE International Workshop on Information Forensics and Security*, pp. 1-8, 2015.

[15] M.M. Islam, J. Kamruzzaman, G. Karmakar, M. Murshed and G. Kahandawa, “Passive Detection of Splicing and Copy-Move Attacks in Image Forgery”, *Proceedings of International Conference on Neural Information Processing*, pp. 555-567, 2018.

[16] B. Bayar and M.C. Stamm, “A Deep Learning Approach to Universal Image Manipulation Detection using a New Convolutional Layer”, *Proceedings of ACM Workshop on Information Hiding and Multimedia Security*, pp. 5-10, 2016.

[17] Y. Zhang, J. Goh, L.L. Win and V.L. Thing, “Image Region Forgery Detection: A Deep Learning Approach”, *Proceedings of International Conference on Information Technology*, pp. 1-11, 2016.

[18] X. Qiu, H. Li, W. Luo and J. Huang, “A Universal Image Forensic Strategy based on Steganalytic Model”, *Proceedings of ACM Workshop on Information Hiding and Multimedia Security*, pp. 165-170, 2016.

[19] S. Akcay, M.E. Kundegorski, C.G. Willcocks and T.P. Breckon, “Using Deep Convolutional Neural Network Architectures for Object Classification and Detection within X-Ray Baggage Security Imagery”, *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 9, pp. 2203-2215, 2018.

[20] Y. Rao and J Ni, "A Deep Learning Approach to Detection of Splicing and Copy-Move Forgeries in Images", *Proceedings of IEEE International Workshop on Information Forensics and Security*, pp. 1-6, 2016.

[21] W. Wang, J. Dong and T. Tan, "Exploring DCT Coefficient Quantization Effects for Local Tampering Detection", *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 10, pp. 1653-1666, 2014.

[22] T. Bianchi and A. Piva, "Image Forgery Localization via Block-Grained Analysis of Jpeg Artifacts", *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 3, pp. 1003-1017, 2012.

[23] N.B. Abd Warif, M.Y.I. Idris, A.W.A. Wahab and R. Salleh, "An Evaluation of Error Level Analysis in Image Forensics", *Proceedings of IEEE International Conference on System Engineering and Technology*, pp. 23-28, 2015.

[24] I.B. Sudiatmika, and R. Fathur, "Image Forgery Detection using Error Level Analysis and Deep Learning", *Telkomnika*, Vol. 17, No. 2, pp. 653-659, 2019.

[25] H. Phan-Xuan, T. Le-Tien, T. Nguyen-Chinh, T. Do-Tieu, Q. Nguyen-Van and T. Nguyen-Thanh, "Preserving Spatial Information to Enhance Performance of Image Forgery Classification", *Proceedings of International Conference on Advanced Technologies for Communications*, pp. 1-5, 2019.

[26] Z. Ding, and F. Yun, "Robust Transfer Metric Learning for Image Classification", *IEEE Transactions on Image* Processing, Vol. 26, No. 2, pp. 660-670, 2016.

[27] D. Han, L. Qigang, and F. Weiguo, "A New Image Classification Method using CNN Transfer Learning and Web Data Augmentation", *Expert Systems with Applications*, Vol. 95, pp. 43-56, 2018.

[28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *Proceedings of IEEE International Conference on System Engineering and Technology*, pp. 354-355, 2014.

[29] K. He, Z. Xiangyu, R. Shaoqing and S. Jian, "Deep Residual Learning for Image Recognition", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[30] G. Huang, L. Zhuang, V.D.M. Laurens and Q.W. Kilian "Densely Connected Convolutional Networks", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700-4708, 2017.

[31] Qiang Chen and Shuicheng Yan. "Network in Network", *Proceedings of IEEE International Conference on System Engineering and Technology*, pp. 1-7, 2013.

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard and M. Kudlur, "Tensorflow: A System for Large-Scale Machine Learning", *Proceedings of Symposium on Operating Systems Design and Implementation*, pp. 265-283, 2016.

[33] F. Chollet, "Keras", Available at https://keras.io, Accessed at 2015.