Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1978 – 1993
© TÜBİTAK
doi:10.3906/elk-2005-37

Image forgery detection based on fusion of lightweight deep learning models

Amit DOEGAR[1,*], Srinidhi HIRIYANNAIAH[2], Siddesh Gaddadevara MATT[2], Srinivasa Krishnarajanagar GOPALIYENGAR[1], Maitreyee DUTTA[1]

[1]Department of Computer Science and Engineering, National Institute of Technical Teachers Training & Research (NITTTR), Chandigarh, India

[2]Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, India

Received: 04.05.2020 • **Accepted/Published Online:** 17.03.2021 • **Final Version:** 26.07.2021

Abstract: Image forgery detection is one of the key challenges in various real time applications, social media and online information platforms. The conventional methods of detection based on the traces of image manipulations are limited to the scope of predefined assumptions like hand-crafted features, size and contrast. In this paper, we propose a fusion based decision approach for image forgery detection. The fusion of decision is based on the lightweight deep learning models namely SqueezeNet, MobileNetV2 and ShuffleNet. The fusion decision system is implemented in two phases. First, the pretrained weights of the lightweight deep learning models are used to evaluate the forgery of the images. Secondly, the fine-tuned weights are used to compare the results of the forgery of the images with the pre-trained models. The experimental results suggest that the fusion based decision approach achieves better accuracy as compared to the state-of-the-art approaches.

Key words: Image forensics, image forgery detection, deep learning, convolutional neural network

1. Introduction

In this digital era, images and videos are being used as influential sources of evidence in a variety of contexts like evidence during trials, insurance fraud, social networking [1], etc. The easy adaptability of editing tools for digital images, especially without any visual proof of manipulation, give rise to questions about their authenticity [1]. It is the job of image forensics authorities to develop technological innovations that would detect the forgeries of images. There are three primary classes of manipulation or forgery detectors studies until now: those supported features descriptors, those supported inconsistent shadows and eventually those supported double JPEG compression [1].

With sophisticated software, it is easy to tamper the contents of the image to influence the opinions of others. Image forgery techniques are broadly classified into two categories namely copy-move and splicing [2]. For copy-move forgery, elements of the image content area are traced and smudge inside a similar image, whereas for splicing forgery, parts of the image content smudge from alternative pictures. To reconstruct the trust in pictures, various image forgery detection techniques have been proposed over the past few years. Many previous studies have tried to extract totally different properties from the image to spot the copy-paste or splicing of forged areas, such as the lighting, shadows, sensing element noise, and camera reflections [3].

Researchers [4–9] determined the credibility of the image wherever it is known either as authentic or

[*]Correspondence: amit@nitttrchd.ac.in

forged. Currently, there are many techniques to spot forged regions [7–15] that exploits the artefacts left by multiple JPEG compression and other techniques of image manipulation to sight the forged regions. Camera primarily based ways [16] have additionally analyzed where the detection relies on demosaicing regularity or sensing element pattern noise wherever the irregularities of the sensing element patterns area unit extracted and compared for anomalies [17].

Forged or manipulated pictures can mislead people and may threaten individuals' life. This paper [2] aims to find the manipulated pictures by automating the method of feature extraction instead of feature engineering or feature extraction through the manual process. Deep learning to make use of highly correlated pixels in a vicinity, thus taking into account grouped native connections [18].

The motivation to use lightweight models in favour to prevent overfitting of the convolutional neural network (CNN) architectures and can be easily deployed on resource constrained hardware and can learn enriched representations [19–23]. ShuffleNet makes more feature map channels for a given computation complexity budget [24], which helps to encode more information and is especially important to the efficiency of small networks. MobileNet [21], makes use of deep-separable convolutions and gains state-of-the-art results and [21] demonstrated the effectiveness of MobileNet when applied to a broad range of tasks. SqueezeNet [25], optimizing the architecture for fast processing speed CNN system with 50 ×, fewer parameters than AlexNet and retains standard accuracy. The lightweight models can be deployed effectively on resource-restricted hardware and can learn enriched representation.

In this paper, the decision fusion of lightweight deep learning based models is proposed for the detection of image forgery. The proposed approach consists of two phases on the pretrained and fine-tuned lightweight deep learning models namely SqueezeNet [25], MobileNetV2 [22], ShuffleNet [24]. In the first phase, features from the images are extracted using lightweight deep learning models without regularization. In the second phase, fine-tuned lightweight deep learning models with fusion and regularization are used to detect image forgery. The main contributions of this paper are:

- An approach of decision fusion based system is proposed using the lightweight for the image forgery detection. The lightweight models used for the fusion decision are SqueezeNet, MobileNetV2, and ShuffleNet.
- The fusion of the decision system is implemented in two phases. First, the pretrained weights for the lightweight models are used to evaluate the forgery detection of the images. Second, the fine-tuned weights are used to compare the results of the forgery detection of the images with the pre-trained models.
- The utilization of the lightweight models leads to the reduction of the number of false matches, thereby reduce the false positive rate and ultimately increase the accuracy of the approach.

The paper is further organized as follows. In Section 2 the related work is discussed on the image forgery detection methods and deep learning models that are used for image forgery detection. In Section 3, the fusion model is proposed and it is followed the regularization applied on the fusion model. In Section 4, the experiments and results are discussed and Section 5 is followed by the conclusion.

2. Related work

In the current world, digital pictures became an awfully vital supply of information. A widespread and easy to use sophisticated software packages could be utilized by any novice users to manipulate the digital pictures in such a way that it does not leave a plain trace. Individuals can share manipulated or forged pictures for amusing on social media. However, forged pictures could also be employed in several serious cases, like scientific publications and media manipulations [26]. Image forgery detection detects whether an image is original or not, an adequate number of features are needed for the detection. Deep learning models are effective for such classification as more features can be extracted. In this section first we identify the different methods of image forgery detection and followed by how the deep learning models can be used to detect it.

There are two primary classes of passive authentication techniques: image-splicing techniques and copy-move forgery detection techniques [8, 9, 26]. The primary attempt to detect copy move forgery was delineated in [27]. Deep learning models have shown their capability of extracting the relevant and robust features from the images to learn their representations, perform computer vision tasks, image classification and recognition. It is also used by the forensics community for the image and video manipulation detection. In [1], one of the methods used for image forgery detection is the trigonometric function remodel for discovering the single and double JPEG compression and manipulated images.

In this way, CNNs are used for the detection of manipulation in the images. The authors [28] used the splicing detection method that is based on principal component analysis (PCA) and support vector machine (SVM). The method first converts the RGB image into a grayscale image on the basis of chrominance components. The features extracted are subsequently used with PCA to increase the efficiency of the image classification using SVM.

To detect the image forgery, the histogram of orientated gradients [29] based model is used. A CNN model with a blocking strategy was used in [30] for image forgery detection. In this method, the image is divided into blocks namely tight blocks and marginal blocks. The blocks were fed into CNN that is recurrent in nature for the forgery detection with SVM as the classifier model. One more CNN model is used in [31] to detect the copy and move image forgery. It uses the Siamese neural network for the forgery detection with 3 convolutional layers, 2 max-pooling layers and 2 fully connected layers. A deep learning model based on Autoencoder is also used for the detection of the forged images [3]. It uses two stages stacked on top of each other.

A CNN edge response model was used in [32] to detect the forgery. The model was trained on the edge patches to detect whether the image is authentic or forged. The patches of the edges in the image were used to localize the spliced region to detect the forgery. A CNN model is proposed in [33] to suppress the image content and learn the manipulated portions of the images. In this model, the filters are used to suppress the image content instead of learning the representations of the image. A localization and resampling method was proposed in [34] for the classification of tampered patches. In [35] authors used VGG-16 based deep learning model for the detection of image forgery. It used a sliding window mechanism to scan the image and obtain the fragment of the image to obtain the manipulated region of the image for the detection of the forgery. A region based CNN (R-CNN) is used for the detection of the image forgery in [36]. It fused the streams of the images to localize the manipulated portions of the images. A deep learning model was proposed in [23] where the original image was manipulated in shape and size to detect the forgery. It used the MobileNetV2 model [22] for the detection of image modification. The extracted features from the model are assembled to conclude whether the image is forged or not.

In [37] for image forgery detection, the extracted features are assigned to a set of predefined bins called codewords. These codewords are used to generate the set of feature vectors namely bag of features (BoF). This is used for the classification as forged or not instead of the fully connected layer. A CNN model is used in [38] that is based on the semantic segmentation of boundaries in the images. The segmentation is carried out based on the semantics of the pixels in the image. It uses nonaligned JPEG forgery methods to detect the semantics of the pixels and assign a label to it. The semantic based method helps to determine the accurate boundaries of the image.

A lightweight model is used in [39] to detect the face. The metrics used for the evaluation are based on image-based and pixel based [26]. In the image based methods, the metric is based on whether the image is forged or authentic. In the pixel based methods, the pixels of the image are classified as copy-move or authentic. In this paper, the proposed system uses the image based measures for the evaluation of the accuracy and other metrics. In the existing methods discussed in this section, the deep learning CNN based models are used for the extraction of the explicit features of the images including the geometry, wavelet, texture and transformations. Most of these methods use the pretrained weights and need to be altered every time for a new set of images. In the proposed system, a fusion of decision making is involved to detect the manipulation in the images. The proposed fusion model is discussed in the further sections.

3. Proposed fusion system

The architecture of the proposed decision fusion is based on the lightweight deep learning models as shown in Figure 1. The lightweight deep learning models chosen are SqueezeNet [25], MobileNetV2 [22], and ShuffleNet [24]. The proposed system is implemented in two phases i.e. with pre-trained and fine-tuned deep learning models. In the pre-trained models implementation, regularization is not applied and the pre-trained weights are used and for the fine-tuned implementation, regularization is applied to detect image forgery. Each phase consists of three stages namely, data pre-processing, classification and fusion. In the data pre-processing stage, the image in the query is pre-processed based on the dimensions required by the deep learning models. SVM is used for the classification of the image as forged or non-forged. Initially, we discuss the lightweight deep learning models and then the strategy used for the regularization is discussed in the further sections.

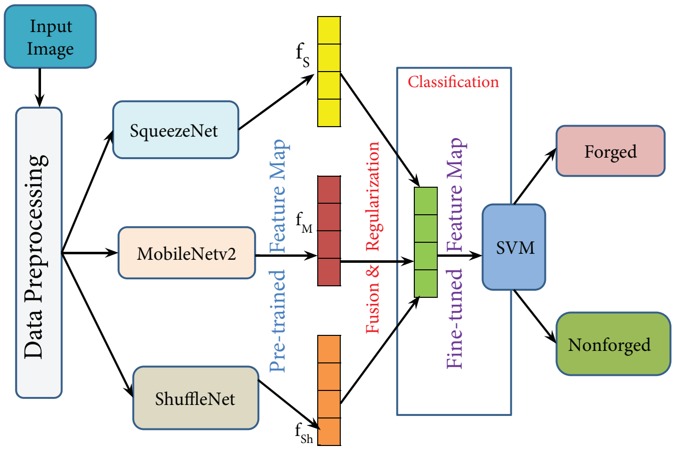

Figure 1. Fusion based decision model for forgery detection.

3.1. Data preprocessing

In this stage, the image in a query that needs to be identified whether it is forged or not is subjected to preprocessing. The height and width of the image required for SqueezeNet is 227 × 227. The height and width of the image required for MobileNetV2 is 224 × 224. The height and width of the image required for ShuffleNet is 224 × 224. The input image is preprocessed first based on the dimensions required for each of the models. Each model then takes the input image to produce feature vector in further stages.

3.2. Lightweight deep learning models

The different lightweight deep learning models that are considered for fusion are SqueezeNet, MobileNetV2, and ShuffleNet. These models are used for the image classification problems numerously. In this section, these models are discussed briefly. The lightweight models[1] considered are summarized as shown in the Table 1. It represents the depth, parameters and the image input size required for the lightweight models namely, SqueezeNet, MobileNetV2, and ShuffleNet.

3.2.1. SqueezeNet

It is a CNN trained on the ImageNet dataset with 18 layers deep and can classify the images up to 1000 categories. The network has learned rich representations of the images with 1.24 million parameters [25]. It requires only a few floating point operations for the image classification.

3.2.2. MobileNetV2

It is a CNN trained on the ImageNet dataset with 53 layers deep and can classify the images up to 1000 categories [22]. The performance of the classification is improved based on the learning of the rich representations of the images.

3.2.3. ShuffleNet

It is a CNN that is also trained on the ImageNet dataset with 50 layers deep and can classify the images up to 1000 categories [24].

Table 1. Parameters of lightweight deep learning models. (Depth represents the largest number of sequential convolutional or fully connected layers on a path from the input layer to the output layer, parameter represents the total number of learnable parameters in each layers and image input size represents the required input image size).

Models	Depth	Parameter (millions)	Image input size
SqueezeNet	18	1.24	227 × 227
MobileNetV2	53	3.5	224 × 224
ShuffleNet	50	1.4	224 × 224

3.3. Classifier

SVM is used as a classifier. SVM is popular and efficient [40] for binary classification. The performance of the proposed approach is evaluated at the image level by calculating the performance metrics like precision, recall also known as true positive rate (TPR), false positive rate (FPR), F-score and accuracy.

[1] MathWorks Inc. (2020). MATLAB [online]. Website https://in.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html [accessed 08 April 2020].

3.4. Fusion model and regularization

The proposed system is first implemented with lightweight deep learning models using pretrained weights for the image forgery detection, afterward, the proposed system is implemented as a fusion of the decision of lightweight models as discussed in the previous section. Initially, the input image is passed to the lightweight models to obtain their feature maps respectively. The feature map from the SqueezeNet is denoted as f_s, the feature map from the MobileNetV2 is denoted as f_m, the feature map from the ShuffleNet is denoted as f_{sh}. For the fusion model, the pretrained lightweight deep learning model's output feature mapping f_p is used. This feature map f_p is a combination of the feature maps obtained from the lightweight models as shown in Equation (1).

$$f_p = f_s + f_m + f_{sh} \tag{1}$$

The fusion model uses feature map f_p as a local descriptor for an input patch to extract the features of the image. The image for the fusion model is represented as a function Y_{fusion} = f(x) where x is the patch in the input image. For a test image size m × n, a sliding window of size p × p is used to compute the local descriptor Y_{fusion} is computed as shown in the equation (2) where $Y_1, Y_2, , Y_T$ represents the descriptors of the patches of the image obtained from the deep learning models. It is obtained as a concatenation of all the input patches x_i and the new image representation is given by equation (3) where s is the size of the stride used for transforming the input patch, this new image representation f_{fusion} is used as the feature map for the classification by the SVM as forged or nonforged.

$$Y_{fusion} = [Y_1 + Y_2 + ... + Y_T] \tag{2}$$

$$f_{fusion} = \frac{m-w}{s} + 1 * \frac{n-w}{s} + 1 \tag{3}$$

For fine tuning of the parameters of the fusion model, the initialization of the weight kernels is used as shown in Equation (4).

In this equation W_f represents the weights of the fusion model, W_s represents the weights of the SqueezeNet model, W_m represents the weights of the MobileNetV2 model and W_{sh} represents the weights of the ShuffleNet model. The weight of the fusion model W_f is initialized as shown in Equation (5). The initialization of the weights acts as a regularization term and facilitates the fusion model to learn the robust features of detecting the forgery rather than the complex image representations.

$$W_f = [W_{sj} W_{mj} W_{shj}] j = 1, 2, 3 \tag{4}$$

$$W_f = [W_s^{4k-2} W_m^{4k-2} W_{sh}^{4k}] \text{ where } k = [[j+1] mod 11] + 1 \tag{5}$$

4. Experiments and results

In this section, the experiments and results of the proposed fusion model are discussed. The experiment is carried out in two stages. In the first stage, the lightweight deep learning models are used with the pretrained weights, in the second stage, the fusion model with the strategy of weight initialization as discussed in the previous section is used to detect image forgery. The configuration of the system used for the experiments is as shown in the Table 2.

Table 2. System configuration details.

Hardware	Intel(R) Xeon(R) Silver 4110 CPU with 2.10 GHZ, 128 GB RAM
GPU	Tesla P4
Software	Ubuntu 18.04 with MATLAB R2019b

4.1. Dataset

The dataset used for the experiment is benchmark publicly available MICC-F220 [41] of 110 nonforged images and 110 forged images with 3 channels i.e. color images of size 722 × 480 to 800 × 600 pixels. As shown in Figure 2, Figures 2a–2j are forged images with 10 different combinations of geometrical and transformations attacks and Figure 2k is the nonforged image. From the dataset 154 images are chosen randomly for training purposes and remaining for testing purpose.

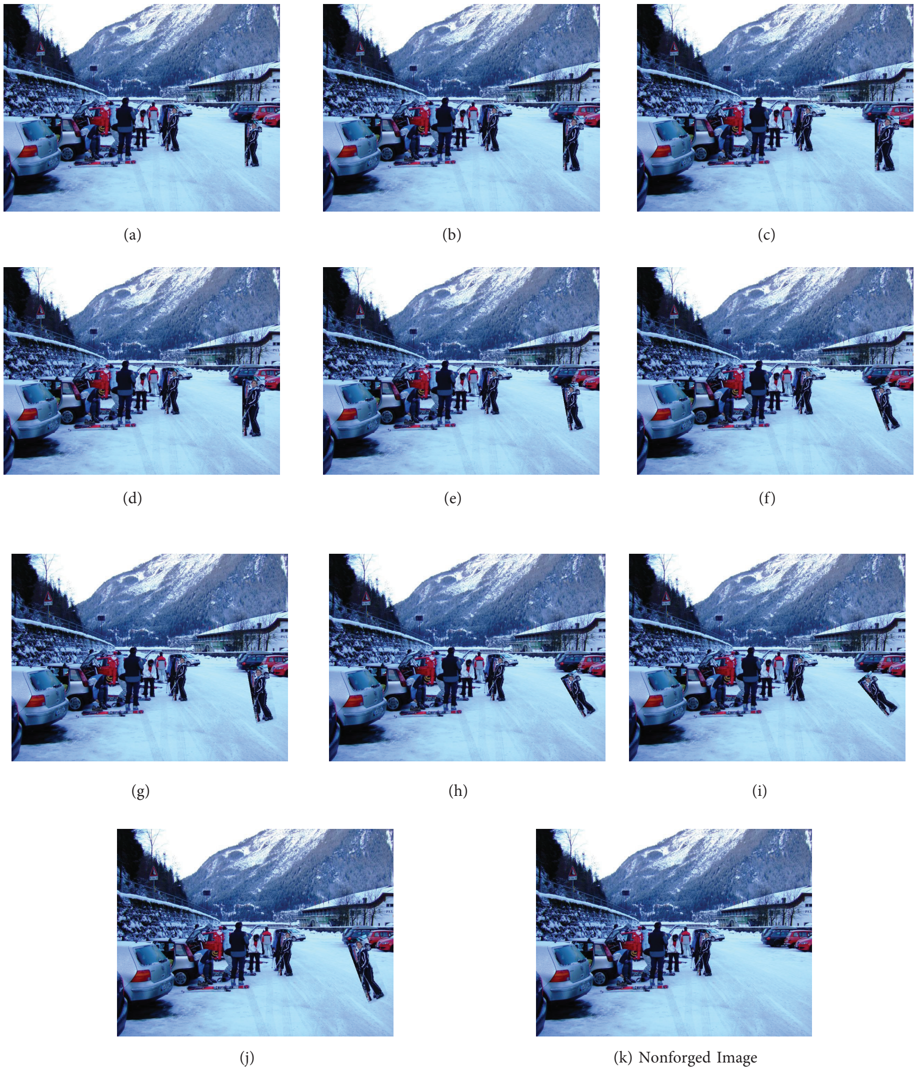
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) Nonforged Image

Figure 2. Dataset with 10 different combinations of geometrical and transformation attacks; (a–j), forged; (k), nonforged images.

4.2. Baseline models and metrics

The baseline models that are used for the comparison of the fusion model are summarized as follows.

- SIFT: It uses the forensic method of the image forgery detection using a scale invariant features transform (SIFT) approach [41].
- SURF: It uses a speeded up robust features (SURF) and hierarchical agglomerative clustering (HAC) for the image forgery detection [42].
- DCT: It uses discrete cosine transform (DCT) features for each block and through lexicographical sorting of block-wise DCT coefficients for the image forgery detection [43].
- PCA: It uses PCA on the image blocks to reduce the dimension space and perform lexicographical sorting for the image forgery detection [44].
- CSLBP: It uses center-symmetric local binary pattern (CSLBP) based on the combined features of Hessian points for the image forgery detection [45].
- SYMMETRY: It uses the local symmetry value of an image to compute the key points for image forgery detection [46].
- CLUSTERING strategy: It uses SIFT features with a clustering strategy to detect image tampering [47].

The basic metrics that are used for the evaluation of the fusion model are recall (R), precision (P), F-score and accuracy as shown in Equations (eqs. (7) to (10)). The confusion matrix is used as the basis for the evaluation of the forged and nonforged images as shown in the Table 3 and the notations used are:

- TPn: Forged Image detected as forged,
- FNn: Forged Image detected as nonforged,
- FPn: Nonforged Image detected as forged,
- TNn: Nonforged Image detected as nonforged.

$$FPR = \frac{FPn}{FPn + TNn} \tag{6}$$

$$Precision = \frac{TPn}{TPn + FPn} \tag{7}$$

$$Recall = \frac{TPn}{TPn + FNn} \tag{8}$$

$$\text{F-score} = 2 * \frac{P * R}{P + R} \tag{9}$$

$$Accuracy = \frac{TPn + TNn}{TPn + TNn + +FPn + FNn} \tag{10}$$

ROC curve is used to estimate the values of the AUC for the pre-trained and also for the fine-tuned lightweight deep learning models.

Table 3. Confusion matrix for evaluation of image forgery.

Actual	Predicted forged	Predicted nonforged
Forged	True positive (TPn)	False negative (FNn)
Nonforged	False positive (FPn)	True negative (TNn)

4.3. Pretrained lightweight deep learning models

In this section, the results of the pretrained lightweight models are discussed. The three models SqueezeNet, MobileNetV2 and ShuffleNet are used with the pretrained weights for the image forgery detection.

The Table 4 shows the confusion matrix and accuracy for the SqueezeNet, MobileNetV2 and ShuffleNet models. It can be observed that the accuracy of the SqueezeNet model is 89.39% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 39.39%. However, the wrong forged prediction is 10.61%. Accuracy of the MobileNetV2 model is 92.42% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 42.42%. However, the wrong forged prediction is 7.58%. Accuracy of the ShuffleNet model is 90.90% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 40.91%. However, the wrong nonforged prediction is 9.09%.

Table 4. Confusion matrix and accuracy for pretrained models.

	SqueezeNet			MobileNetV2			ShuffleNet		
	Forged	Nonforged	Accuracy	Forged	Nonforged	Accuracy	Forged	Nonforged	Accuracy
Forged	33	0	89.93%	33	0	92.42%	33	0	90.90%
Nonforged	7	26		5	28		6	27	

The ROC curve is used to estimate the AUC values for the pretrained lightweight convolutional neural networks as shown in the Figure 3. Figure 3a represents the ROC curve for the SqueezeNet represents the ROC with AUC of 90.08%. Figure 3b represents the ROC curve for the MobileNetV2 represents the ROC with AUC of 91.73%. Figure 3c represents the ROC curve for the ShuffleNet represents the ROC with AUC of 91.36%.

4.4. Fine-tuned lightweight deep learning models

In this section, the results of the fine-tuned lightweight models are discussed. The three models namely SqueezeNet, MobileNetV2, and ShuffleNet are used with the fine-tuned weights for the image forgery detection. Table 5 shows the confusion matrix and accuracy for the fine-tuned SqueezeNet, MobileNetV2, and ShuffleNet models. It can be observed that the accuracy of the SqueezeNet model is 93.93% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 43.94%. However, the wrong forged prediction is 6.06%. Accuracy of the MobileNetV2 model is 95.45% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 45.45%. However, the wrong forged prediction is 4.55%. Accuracy of the ShuffleNet model is 95.45% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 45.45%. However, the wrong forged prediction is 4.55%. Figure 4a represents the ROC curve for the SqueezeNet with AUC of 90.35%. Figure 4b represents the ROC curve for the MobileNetV2 with AUC of 94.12%. Figure 4c represents the ROC curve for the ShuffleNet with AUC of 94.49%.

The ROC curve is used to estimate the AUC values for the fine-tuned lightweight deep learning models as shown in Figure 4.

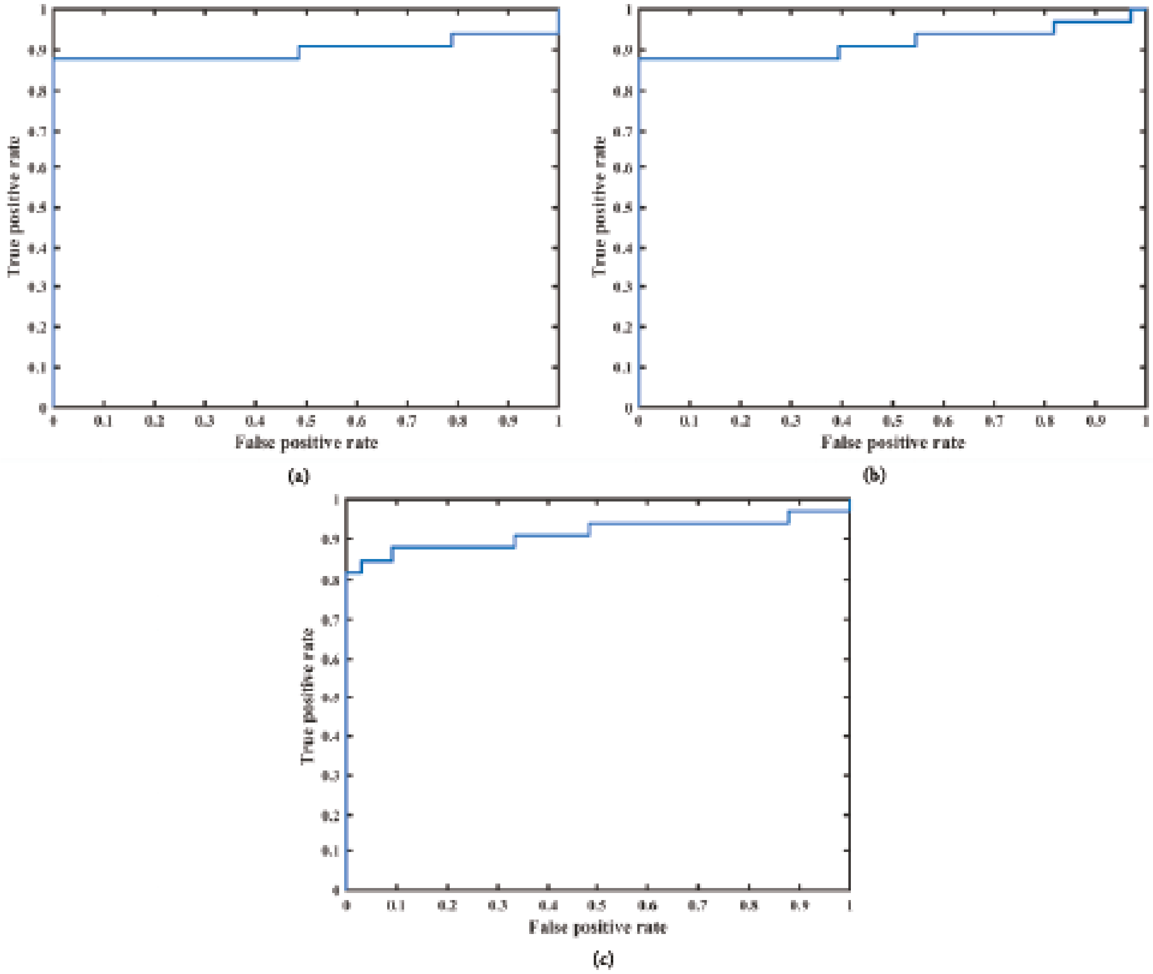

Figure 3. ROC for the pretrained models. (a) SqueezeNet with AUC of 90.08%, (b) MobileNetV2 with AUC of 91.73%, (c) ShuffleNet with AUC of 91.36%.

Table 5. Confusion matrix and accuracy for fine-tuned models.

	SqueezeNet			MobileNetV2			ShuffleNet		
	Forged	Nonforged	Accuracy	Forged	Nonforged	Accuracy	Forged	Nonforged	Accuracy
Forged	33	0	93.93%	33	0	95.45%	33	0	95.45%
Nonforged	4	29		3	30		3	30	

4.5. Fusion model

In this section, the results of the fusion models are discussed. Table 6 shows the confusion matrix and accuracy for the pretrained fusion model and fine-tuned fusion model. It can be observed that the accuracy of the pretrained fusion model is 93.93% and the percentage of the prediction of the correct forged is 50% and correct nonforged is 43.94%. However, the wrong forged prediction is 6.06%. It can be clearly observed that the percentage of wrong nonforged prediction is less as compared to the pretrained lightweight convolutional deep learning models. The accuracy of the pretrained fusion model is higher than the pretrained lightweight deep

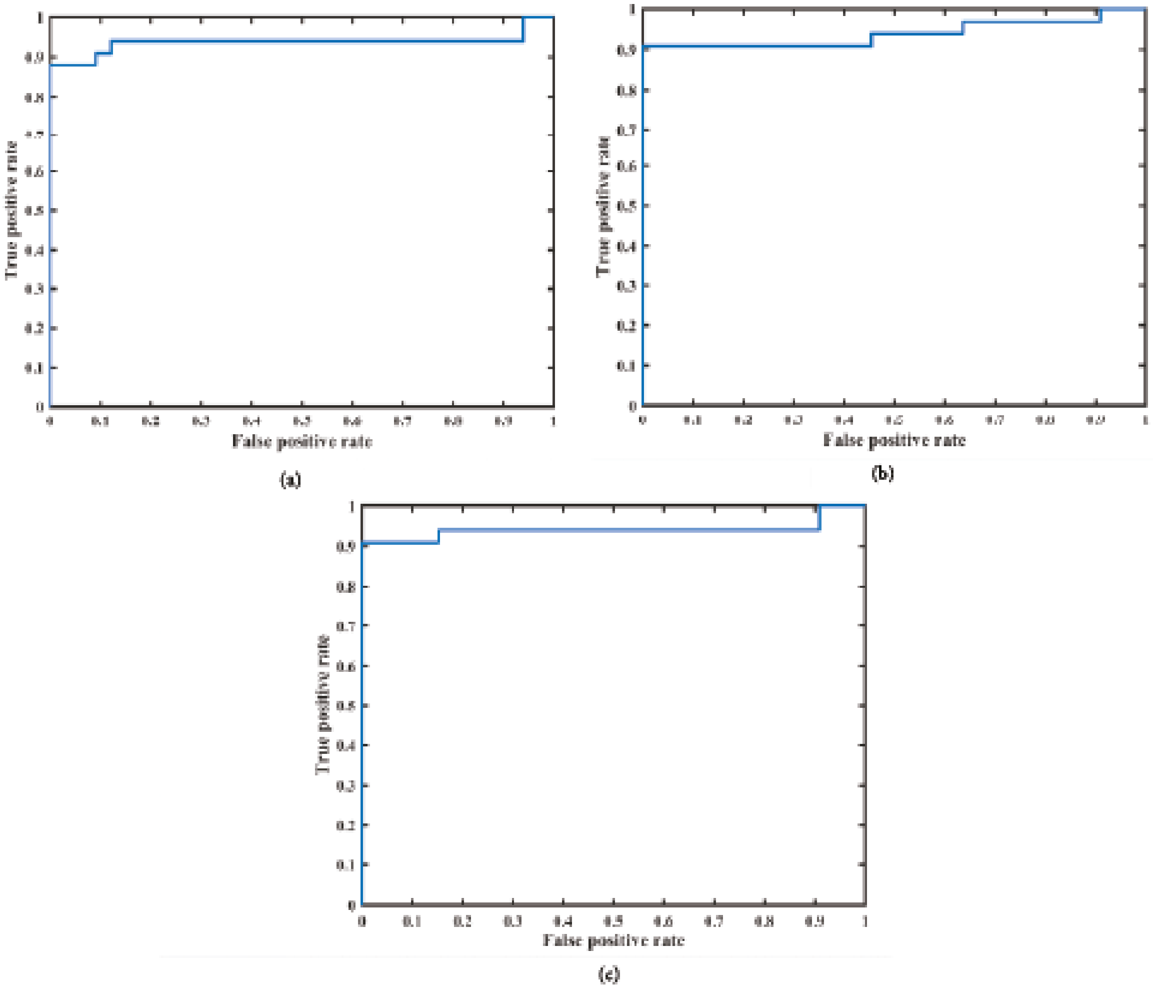

Figure 4. ROC for the fine-tuned models. (a) SqueezeNet with AUC of 90.35%, (b) MobileNetV2 with AUC of 94.12%, (c) ShuffleNet with AUC of 94.49%.

learning models.

Table 6. Confusion matrix and accuracy obtained for different fusion models.

Test dataset	Pretrained fusion model			Fine-tuned fusion model		
	Forged	Nonforged	Accuracy	Forged	Nonforged	Accuracy
Forged	33	0	93.93%	33	0	96.87%
Nonforged	4	29		2	31	

It can be observed that the accuracy of the fine-tuned fusion network is 96.87% and the percentage of the prediction of the correct forged is 50% and the correct nonforged is 46.97%. However, the wrong forged prediction is 3.03%. It can be clearly observed that the percentage of wrong forged prediction is less as compared to the fine-tuned lightweight deep learning models. The accuracy of the fine-tuned fusion model is higher than the fine-tuned lightweight deep learning models.

4.6. Performance comparison

4.6.1. Performance comparison with pretrained lightweight models

In this section, the performance comparison of the fusion model is carried out with pretrained lightweight models. The metrics used for the comparison are precision, recall, F-score and accuracy. The results of the performance comparison are as shown in the Table 7.

Table 7. Performance metrics obtained with pretrained lightweight models.

Model	Precision (%)	Recall (%)	F-score(%)	Accuracy (%)
SqueezeNet	82.50	100	90.41	89.39
Mobilenet-v2	86.84	100	92.95	92.42
ShuffleNet	84.61	100	91.66	90.90
Fusion model	89.18	100	94.28	93.93

The precision and recall metrics are important to determine the effectiveness of the deep learning models. According to Equations (eqs. (7) to (10)) the values in the Table 7 were obtained. Table 7 shows values of precision, recall, F-score and accuracy of the pretrained models. The precision of the SqueezeNet, MobileNetV2, and ShuffleNet are 82.50, 86.84, and 84.61, respectively. The recall is 100 for all the of lightweight deep learning models. The F-score of the SqueezeNet, MobileNetV2, and ShuffleNet are 90.41, 92.95, and 91.66, respectively. The accuracy of the SqueezeNet, MobileNetV2, and ShuffleNet are 89.39, 92.42, and 90.90, respectively. The proposed fusion model has precision of 89.18, recall of 100, F-score of 94.28 and accuracy of 93.93 respectively. Hence, it can be seen that the proposed fusion model achieves 2× times more F-score and 3× times more precision and better accuracy as compared to the other models.

4.6.2. Performance comparison with fine-tuned lightweight models

In this section, the performance comparison of the fusion model is carried out with fine-tuned lightweight models. The metrics used for the comparison are precision, recall, F-score and accuracy. The results of the performance comparison are as shown in the Table 8.

Table 8. Performance metrics obtained with fine-tuned lightweight models.

Model	Precision (%)	Recall (%)	F-score (%)	Accuracy (%)
SqueezeNet	89.19	100	94.29	93.93
Mobilenet-v2	91.66	100	95.65	95.45
ShuffleNet	91.66	100	95.65	95.45
Fusion model	94.11	100	96.96	96.87

Table 8 shows values of precision, recall, F-score and accuracy of the fine-tuned models. The precision of the SqueezeNet, MobileNetV2, and ShuffleNet are 89.19, 91.66, and 91.66, respectively. The recall is 100 for all the of fine-tuned learning models. The F-score of the SqueezeNet, MobileNetV2, and ShuffleNet are 94.29, 95.65, and 95.65, respectively. The accuracy of the SqueezeNet, MobileNetV2, and ShuffleNet are 93.93, 95.45, and 95.45, respectively. The proposed fusion model has precision of 94.11, recall of 100, F-score of 96.96 and accuracy of 96.87, respectively. Hence, it can be seen that the proposed fusion model achieves more precision and better accuracy as compared to the other models.

The results of the Tables 7 and 8 clearly demonstrate the proposed fusion model with the pretrained and fine-tuned parameters has accuracy of 93.93 and 96.87 respectively. The accuracy is improved by 3 × times in the fine-tuned models and thus helps in the better classification of the image forgery. The improvement in the accuracy is obtained using the regularization of the weights in the proposed fusion model.

The results of the performance comparison of the fusion model with the baselines are as shown in the Table 9. The metrics used for the comparison are the FPR and TPR as they give the correctness of the model for the image forgery detection. The FPR for the baseline 1 [41] is 8%, baseline 2 [42] is 3.64%, baseline 3 [43] is 84%, baseline 4 [44] is 86%, baseline 5 [45] is 2.89%, baseline 6 [46] is 5.45%, baseline 7 [47] is 7.63%, proposed pretrained fusion model is 12.12% and proposed fine-tuned fusion model is 6.06%. The TPR for the baseline 1 [41] is 100%, baseline 2 [42] is 73.64%, baseline 3 [43] is 89%, baseline 4 [44] is 87%, baseline 5 [45] is 96%, baseline 6 [46] is 83.64%, baseline 7 [47] is 97.87%, proposed pretrained fusion model is 100% and proposed fine-tuned fusion model is 100%. Therefore, it can be observed that the fusion model has higher TPR and less FPR as compared to the baseline models due to weight initialization strategy used for the fusion model.

Table 9. Comparative performance of proposed fusion models with baselines.

Approach	FPR (%)	TPR (%)
SIFT [41]	8	100
SURF [42]	3.64	73.64
DCT [43]	84	89
PCA [44]	86	87
CSLBP [45]	2.89	96
SYMMETRY [46]	5.45	83.64
CLUSTERING strategy [47]	7.63	97.87
Pretrained fusion model (proposed)	12.12	100
Fine-tuned fusion model (proposed)	6.06	100

5. Conclusion

Image forgery detection helps to differentiate between the original and the manipulated or fake images. In this paper, a decision fusion of lightweight deep learning based models is implemented for image forgery detection. The idea was to use the lightweight deep learning models namely SqueezeNet, MobileNetV2, and ShuffleNet and then combine all these models to obtain the decision on the forgery of the image. Regularization of the weights of the pretrained models is implemented to arrive at a decision of the forgery. The experiments carried out indicate that the fusion based approach gives more accuracy than the state-of-the-art approaches. In the future, the fusion decision can be improved with other weight initialization strategies for image forgery detection.

References

[1] Amerini I, Uricchio T, Ballan L, Caldelli R. Localization of JPEG double compression through multi-domain convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); Honolulu, HI, USA; 2017. pp. 1865-1871. doi: 10.1109/CVPRW.2017.233

[2] Xiao B, Wei Y, Bi X, Li W, Ma J. Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. Information Sciences 2020; 511: 172-191. doi: 10.1016/j.ins.2019.09.038

[3] Zhang Y, Goh J, Win LL, Thing VL. Image region forgery detection: a deep learning approach. SG-CRC 2016; 2016: 1-11.

[4] Goh J, Thing VL. A hybrid evolutionary algorithm for feature and ensemble selection in image tampering detection. International Journal of Electronic Security and Digital Forensics 2015; 7 (1): 76-104.

[5] Sutthiwan P, Shi YQ, Zhao H, Ng TT, Su W. Markovian rake transform for digital image tampering detection. In: Shi YQ, Emmanuel S, Kankanhalli MS, Chang S-F, Radhakrishnan R (editors). Transactions on Data Hiding and Multimedia Security VI. Lecture Notes in Computer Science, Vol. 6730. Berlin, Germany: Springer; 2011, pp. 1-17.

[6] He Z, Lu W, Sun W, Huang J. Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recognition 2012; 45 (12): 4292-4299.

[7] Chang IC, Yu JC, Chang CC. A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. Image and Vision Computing 2013; 31 (1): 57-71.

[8] Rhee KH. Median filtering detection based on variations and residuals in image forensics. Turkish Journal of Electrical Engineering & Computer Science 2017; 25 (5): 3811-3826.

[9] Lamba AK, Jindal N, Sharma S. Digital image copy-move forgery detection based on discrete fractional wavelet transform. Turkish Journal of Electrical Engineering & Computer Science 2018; 26 (3): 1261-1277.

[10] Lin Z, He J, Tang X, Tang CK. Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition 2009; 42 (11): 2492-2501.

[11] Chen YL, Hsu CT. Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection. IEEE Transactions on Information Forensics and Security 2011; 6 (2): 396-406.

[12] Bianchi T, Piva A. Image forgery localization via block-grained analysis of JPEG artifacts. IEEE Transactions on Information Forensics and Security 2012; 7 (3): 1003-1017.

[13] Zach F, Riess C, Angelopoulou E. Automated image forgery detection through classification of JPEG ghosts. In: Springer 2012 Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium; Berlin, Heidelberg; 2012. pp. 185-194.

[14] Thing VL, Chen Y, Cheh C. An improved double compression detection method for JPEG image forensics. In: IEEE International Symposium on Multimedia; Irvine, CA, USA; 2012. pp. 290-297.

[15] Wang W, Dong J, Tan T. Exploring DCT coefficient quantization effects for local tampering detection. IEEE Transactions on Information Forensics and Security 2014; 9 (10): 1653-1666.

[16] Amerini I, Caldelli R, Cappellini V, Picchioni F, Piva A. Estimate of PRNU noise based on different noise models for source camera identification. International Journal of Digital Crime and Forensics 2010; 2 (2): 21-33.

[17] Popescu AC, Farid H. Exposing digital forgeries in color filter array interpolated images. IEEE Transactions on Signal Processing 2005; 53 (10): 3948-3959. doi: 10.1109/TSP.2005.855406

[18] Hadji I, Wildes RP. What do we understand about convolutional networks? arXiv 2018; preprint arXiv:1803.08834.

[19] Khan A, Sohail A, Zahoora U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks. arXiv 2019; preprint arXiv:1901.06032.

[20] Rao Y, Ni J, Zhao H. Deep learning local descriptor for image splicing detection and localization. IEEE Access 2020; 8: 25611-25625.

[21] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv 2017; preprint arXiv:1704.04861.

[22] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Salt Lake City, UT, USA; 2018. pp. 4510-4520.

[23] Phan-Xuan H, Le-Tien T, Nguyen-Chinh T, Do-Tieu T, Nguyen-Van Q et al. Preserving spatial information to enhance performance of image forgery classification. In: IEEE ATC International Conference on Advanced Technologies for Communications; Hanoi, Vietnam; 2019. pp. 50-55.

[24] Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: IEEE 2018 Conference on Computer Vision and Pattern Recognition (CVPR); Salt Lake City, UT, USA; 2018. pp. 6848-6856.

[25] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv 2016; preprint arXiv:1602.07360.

[26] Al-Qershi OM, Khoo BE. Evaluation of copy-move forgery detection: datasets and evaluation metrics. Multimedia Tools and Applications 2018; 77 (24):31807-31833. doi: 10.1007/s11042-018-6201-4

[27] Al-Qershi OM, Khoo BE. Passive detection of copy-move forgery in digital images: state-of-the-art. Forensic Science International 2013; 231 (1-3): 284-95. doi: 10.1016/j.forsciint.2013.05.027

[28] Hakimi F, Hariri M, GharehBaghi F. Image splicing forgery detection using local binary pattern and discrete wavelet transform. In: IEEE 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI); Tehran, Iran; 2015. pp. 1074-1077.

[29] Lee JC, Chang CP, Chen WK. Detection of copy–move image forgery using histogram of orientated gradients. Information Sciences 2015; 321: 250-262. doi: 10.1016/j.ins.2015.03.009

[30] Zhou J, Ni J, Rao Y. Block-based convolutional neural network for image forgery detection. In: Springer International Workshop on Digital Watermarking; Magdeburg, Germany; 2017. pp. 65-76.

[31] Zhang J, Zhu W, Li B, Hu W, Yang J. Image copy detection based on convolutional neural networks. In: Springer Chinese Conference on Pattern Recognition; Singapore; 2016. pp. 111-121.

[32] Chen C, McCloskey S, Yu J. Image splicing detection via camera response function analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Honolulu, HI, USA; 2017. pp. 5087-5096.

[33] Bayar B, Stamm MC. A deep learning approach to universal image manipulation detection using a new convolutional layer. In: ACM 4th ACM Workshop on Information Hiding and Multimedia Security; Vigo, Spain; 2016. pp. 5-10. doi: 10.1145/2909827.2930786

[34] Bunk J, Bappy JH, Mohammed TM, Nataraj L, Flenner A et al. Detection and localization of image forgeries using resampling features and deep learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops; Honolulu, HI, USA; 2017. pp. 1881-1889.

[35] Kuznetsov A. Digital image forgery detection using deep learning approach. Journal of Physics: Conference Series 2019; 1368 (3): 032028. doi: 10.1088/1742-6596/1368/3/032028.

[36] Zhou P, Han X, Morariu VI, Davis LS. Learning rich features for image manipulation detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Salt Lake City, UT, USA; 2018. pp. 1053-1061.

[37] Passalis N, Tefas A. Training lightweight deep convolutional neural networks using bag-of-features pooling. IEEE Transactions on Neural Networks and Learning Systems 2018; 30 (6): 1705-1715.

[38] Alipour N, Behrad A. Semantic segmentation of JPEG blocks using a deep CNN for non-aligned JPEG forgery detection and localization. Multimedia Tools and Applications 2020; 1-17. doi: 10.1007/s11042-019-08597-8

[39] Triantafyllidou D, Nousi P, Tefas A. Lightweight two-stream convolutional face detection. In: IEEE 25th European Signal Processing Conference (EUSIPCO); Kos, Greece; 2017. pp. 1190-1194. doi: 10.23919/EUSIPCO.2017.8081396

[40] Mushtaq S, Mir AH. Forgery detection using statistical features. In: IEEE Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH); Ghaziabad, India; 2014. pp. 92-97. doi: 10.1109/CIPECH.2014.7019062

[41] Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G. A sift-based forensic method for copy–move attack detection and transformation recovery. IEEE Transactions on iInformation Forensics and Security 2011; 6 (3): 1099-1110.

[42] Kekre HB, Mishra D, Halarnkar PN, Shende P, Gupta S. Digital image forgery detection using Image hashing. In: IEEE International Conference on Advances in Technology and Engineering (ICATE); Mumbai, India; 2013. pp.1-6. doi: 10.1109/ICAdTE.2013.6524736

[43] Fridrich AJ, Soukal BD, Lukáš AJ. Detection of copy-move forgery in digital images. In: Proceedings of Digital Forensic Research Workshop; Cleveland, OH, USA; 2003.

[44] Popescu AC, Farid H. Exposing digital forgeries by detecting duplicated image regions. Computer Science Technical Report 2004; TR2004-515: 1-11.

[45] Uliyan DM, Jalab HA, Wahab A, Ainuddin W, Sadeghi S. Image region duplication forgery detection based on angular radial partitioning and Harris key-points. Symmetry 2016; 8 (7): 62. doi: 10.3390/sym8070062

[46] Vaishnavi D, Subashini TS. Application of local invariant symmetry features to detect and localize image copy move forgeries. Journal of Information Security and Applications 2019; 44: 23-31. doi: 10.1016/j.jisa.2018.11.001

[47] Abdel-Basset M, Manogaran G, Fakhry AE, El-Henawy I. 2-Levels of clustering strategy to detect and locate copy-move forgery in digital images. Multimedia Tools and Applications 2018; 79: 5419-5437. doi: 10.1007/s11042-018-6266-0