electronics

Article

Image Forgery Detection Using Deep Learning by Recompressing Images

Syed Sadaf Ali [1,*,†], Iyyakutti Iyappan Ganapathi [2,†], Ngoc-Son Vu [1], Syed Danish Ali [3], Neetesh Saxena [4] and Naoufel Werghi [2]

[1] ETIS, CY Cergy Paris Université, ENSEA, CNRS, UMR 8051, 95000 Cergy, France; son.vu@ensea.fr
[2] C2PS & KUCARS, Khalifa University, Abu Dhabi 127788, United Arab Emirates; iyyakutti.ganapathi@ku.ac.ae (I.I.G.); naoufel.werghi@ku.ac.ae (N.W.)
[3] Machine Intelligence Research (MIR) Labs Gwalior, Gwalior 474001, India; danish.mirlabs@gmail.com
[4] School of Computer Science and Informatics, Cardiff University, Cardiff CF10 3AT, UK; nsaxena@ieee.org
* Correspondence: sadaf.ali@ensea.fr
† These authors contributed equally to this work.

Abstract: Capturing images has been increasingly popular in recent years, owing to the widespread availability of cameras. Images are essential in our daily lives because they contain a wealth of information, and it is often required to enhance images to obtain additional information. A variety of tools are available to improve image quality; nevertheless, they are also frequently used to falsify images, resulting in the spread of misinformation. This increases the severity and frequency of image forgeries, which is now a major source of concern. Numerous traditional techniques have been developed over time to detect image forgeries. In recent years, convolutional neural networks (CNNs) have received much attention, and CNN has also influenced the field of image forgery detection. However, most image forgery techniques based on CNN that exist in the literature are limited to detecting a specific type of forgery (either image splicing or copy-move). As a result, a technique capable of efficiently and accurately detecting the presence of unseen forgeries in an image is required. In this paper, we introduce a robust deep learning based system for identifying image forgeries in the context of double image compression. The difference between an image's original and recompressed versions is used to train our model. The proposed model is lightweight, and its performance demonstrates that it is faster than state-of-the-art approaches. The experiment results are encouraging, with an overall validation accuracy of 92.23%.

Keywords: convolutional neural networks; neural networks; forgery detection; image compression; image processing

Citation: Ali, S.S.; Ganapathi, I.I.; Vu, N.-S.; Ali, S.D.; Saxena, N.; Werghi, N. Image Forgery Detection Using Deep Learning by Recompressing Images. *Electronics* **2022**, *11*, 403. https://doi.org/10.3390/electronics11030403

Academic Editors: Huy Kang Kim and Gemma Piella

Received: 15 November 2021
Accepted: 24 January 2022
Published: 28 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Due to technological advancements and globalization, electronic equipment is now widely and inexpensively available. As a result, digital cameras have grown in popularity. There are many camera sensors all around us, and we use them to collect a lot of images. Images are required in the form of a soft copy for various documents that must be filed online, and a large number of images are shared on social media every day. The amazing thing about images is that even illiterate people can look at them and extract information from them. As a result, images are an integral component of the digital world, and they play an essential role in storing and distributing data. There are numerous tools accessible for quickly editing the images [1,2]. These tools were created with the intention of enhancing and improving the images. However, rather than enhancing the image, some people exploit their capabilities to falsify images and propagate falsehoods [3,4]. This is a significant threat, as the damage caused by faked images is not only severe, but also frequently irreversible.

There are two basic types of image forgery: image splicing and copy-move, which are discussed below:

- Image Splicing: A portion of a donor image is copied into a source image. A sequence of donor images can likewise be used to build the final forged image.
- Copy-Move: This scenario contains a single image. Within the image, a portion of the image is copied and pasted. This is frequently used to conceal other objects. The final forged image contains no components from other images.

The primary purpose in both cases of image forgery is to spread misinformation by changing the original content in an image with something else [5,6]. Earlier images were an extremely credible source for the information exchange, however, due to image forgery, they are used to spread misinformation. This is affecting the trust of the public in images, as the forging of images may or may not be visible or recognizable to the naked eye. As a result, it is essential to detect image forgeries to prevent the spread of misinformation as well as to restore public trust in images. This can be done by exploring the various artifacts left behind when an image forgery is performed, and they can be identified using various image processing techniques.

Researchers have proposed a variety of methods for detecting the presence of image forgeries [7–9]. Conventional image forgery detection techniques detect forgeries by concentrating on the multiple artifacts present in a forged image, such as changes in illumination, contrast, compression, sensor noise, and shadow. CNN's have gained popularity in recent years for various computer vision tasks, including image object recognition, semantic segmentation, and image classification. Two major features contribute to CNN's success in computer vision. Firstly, CNN takes advantage of the significant correlation between adjacent pixels. As a result, CNN prefers locally grouped connections over one-to-one connections between all pixel. Second, each output feature map is produced through a convolution operation by sharing weights. Moreover, compared to the traditional method that depends on engineered features to detect specific forgery, CNN uses learned features from training images, and it can generalize itself to detect unseen forgery. These advantages of CNN make it a promising tool for detecting the presence of forgery in an image. It is possible to train a CNN-based model to learn the many artifacts found in a forged image [10–13]. Thus, we propose a very light CNN-based network, with the primary goal of learning the artifacts that occur in a tampered image as a result of differences in the features of the original image and the tampered region.

The major contribution of the proposed technique are as follows:

- A lightweight CNN-based architecture is designed to detect image forgery efficiently. The proposed technique explores numerous artifacts left behind in the image tampering process, and it takes advantage of differences in image sources through image recompression.
- While most existing algorithms are designed to detect only one type of forgery, our technique can detect both image splicing and copy-move forgeries and has achieved high accuracy in image forgery detection.
- Compared to existing techniques, the proposed technique is fast and can detect the presence of image forgery in significantly less time. Its accuracy and speed make it suitable for real-world application, as it can function well even on slower devices.

The rest of the paper is organized as follows. Section 2 provides a literature review of image forgery detection methodologies. Section 3 introduces the proposed framework for detecting the presence of forgeries in an image. Section 4 contains a discussion of the experimentation and the results achieved. Finally, in Section 5, we summarize the conclusions.

2. Literature Review

Various approaches have been proposed in the literature to deal with image forgery. The majority of traditional techniques are based on particular artifacts left by image forgery, whereas recently techniques based on CNNs and deep learning were introduced, which are mentioned below. First, we will mention the various traditional techniques and then move on to deep learning based techniques.

In [14], the authors' proposed error level analysis (ELA) for the detection of forgery in an image. In [15], based on the lighting conditions of objects, forgery in an image is detected. It tries to find the forgery based on the difference in the lighting direction of the forged part and the genuine part of an image. In [16], various traditional image forgery detection techniques have been evaluated. In [17], Habibi et al., use the contourlet transform to retrieve the edge pixels for forgery detection. In [18], Dua et al., presented a JPEG compression-based method. The discrete DCT coefficients are assessed independently for each block of an image partitioned into non-overlapping blocks of size 8 × 8 pixels. The statistical features of AC components of block DCT coefficients alter when a JPEG compressed image tampers. The SVM is used to classify authentic and forged images using the retrieved feature vector. Ehret et al. in [19] introduced a technique that relies on SIFT, which provides sparse keypoints with scale, rotation, and illumination invariant descriptors for forgery detection. A method for fingerprint faking detection utilizing deep Boltzmann machines (DBM) for image analysis of high-level characteristics is proposed in [20]. Balsa et al. in [21] compared the DCT, Walsh–Hadamard transform (WHT), Haar wavelet transform (DWT), and discrete Fourier transform (DFT) for analog image transmission, changing compression and comparing quality. These can be used for image forgery detection by exploring the image from different domains. Thanh et al. proposed a hybrid approach for image splicing in [22], in which they try to retrieve the original images that were utilized to construct the spliced image if a given image is proven to be the spliced image. They present a hybrid image retrieval approach that uses Zernike moment and SIFT features.

Bunk et al. established a method for detecting image forgeries based on resampling features and deep learning in [23]. Bondi et al. in [24] suggested a method for detecting image tampering by the clustering of camera-based CNN features. Myung-Joon in [2] introduced CAT-Net, to acquire forensic aspects of compression artifact on DCT and RGB domains simultaneously. Their primary network is HR-Net (high resolution). They used the technique proposed in [25], which tells us that how we can use the DCT coefficient to train a CNN, as directly giving DCT coefficients to CNN will not train it efficiently. Ashraful et al. in [26] proposed DOA-GAN, to detect and localize copy-move forgeries in an image, authors used a GAN with dual attention. The first-order attention in the generator is designed to collect copy-move location information, while the second-order attention for patch co-occurrence exploits more discriminative properties. The affinity matrix is utilized to extract both attention maps, which are then used to combine location-aware and co-occurrence features for the network's ultimate detection and localization branches.

Yue et al. in [27] proposed BusterNet for copy-move image forgery detection. It has a two-branch architecture with a fusion module in the middle. Both branches use visual artifacts to locate potential manipulation locations and visual similarities to locate copy-move regions. Yue et al. in [28] employed a CNN to extract block-like characteristics from an image, compute self-correlations between various blocks, locate matching points using a point-wise feature extractor, and reconstruct a forgery mask using a deconvolutional network. Yue et al. in [3] designed ManTra-Net that is s a fully convolutional network that can handle any size image and a variety of forgery types, including copy-move, enhancement, splicing, removal, and even unknown forgery forms. Liu et al. in [29] proposed PSCC-Net, which analyses the image in a two-path methodology: a top-down route that retrieves global and local features and a bottom-up route that senses if the image is tampered and predicts its masks at four levels, each mask being constrained on the preceding one.

In [30] Yang et al., proposed a technique based on two concatenated CNNs: the coarse CNN and the refined CNN, which extracts the differences between the image itself and splicing regions from patch descriptors of different scales. They enhanced their work in [1] and proposed a patch-based coarse-to-refined network (C2RNet). The coarse network is based on VVG16, and the refined network is based on VVG19. In [31] Xiuli et al., proposed a ringed residual U-Net to detect the splicing type image forgery in the images. Younis et al. in [32] utilized the reliability fusion map for the detection of the forgery. By utilizing the

CNNs, Younis et al. in [33] classify an image as the original one, or it contains copy-move image forgery. In [34] Vladimir et al., train four models at the same time: a generative annotation model GA, a generative retouching model GR, and two discriminators DA and DR that checks the output of GA and GR. Mayer et al. in [35] introduced a system that maps sets of image regions to a value that indicates if they include the same or different forensic traces.

In [36] Minyoung et al., designed an algorithm that leverages the automatically recorded image EXIF metadata for training a model to identify whether an image has self-consistency or if its content may have been generated from a single image. In [37] Rongyu et al., proposed a UNet that consists of a dense convolutional and deconvolutional networks. The first is a down-sampling method for retrieving features, while the second is an up-sampling approach for recovering feature map size. In [38] Lui et al., introduced the CNN segmentation-based approach to find manipulated regions in digital photos. First, a uniform CNN architecture is built to deal with various scales' color input sliding windows. Then, using sampling training regions, they meticulously build CNN training processes.

In [39], an unfixed encoder and a fixed encoder are used to build a Dual-encoder U-Net (D-Unet). The unfixed encoder learns the image fingerprints that distinguish between genuine and tampered regions on its own. In contrast, the fixed encoder offers direction data to facilitate the network's learning and detection. In [40] Francesco et al., tested the efficiency of several image forgery detectors over image-to-image translation, including both ideal settings and even in the existence of compression, which is commonly performed when uploading to social media sites. Kadam et al. in [41] Proposed a method based on multiple image splicing using MobileNet V1.Jaiswal et al. in [42] proposed a framework in which images are fed into a CNN and then processed through several layers to extract features, which are then utilized as a training vector for the detection model. For feature extraction, they employed a pre-trained deep learning resnet-50.

Hao et al. in [43] proposed using an attention method to analyze and refine feature maps for the detection task. The learned attention maps emphasize informative areas to enhance binary classification (false face vs. genuine face) and illustrate the altered regions. In [44], Nguyen et al., developed a CNN that employs a multi-task learning strategy to detect altered images and videos while also locating the forged areas. The information received from one work is shared with the second task, improving both activities' performance. To boost the network's generability, a semi-supervised learning strategy is adopted. An encoder and a Y-shaped decoder are included in the network. Li et al. introduced a deepfake detection method in [45]. The DeepFake techniques can only create fixed-size images of the face, which must be affinely warped to match the source's face arrangement. Due to the resolution disparity between the warped face area and the surrounding context, this warping produces different artifacts. As a result, DeepFake Videos can be identified using these artifacts. Komodakis et al. in [46] suggested a method for learning image features by training CNNs to recognize the two-dimensional rotation that is applied to the picture that it receives as input. The method proposed in [47] is composed of three parts: single image super-resolution, semantic segmentation super-resolution, and feature affinity module for semantic segmentation. In [48] Yu et al., used dual attention upon pyramid visual feature maps to fully examine the visual-semantic relationships and enhance the level of produced sentences. For more details about image forgery and media, forensics readers may refer to [5–13].

The state-of-the-art techniques available for detecting the presence of tampering in the images generally take a very long time to process the images. Most of them can detect either image splicing forgery or copy-move type of forgery, not both. Another major issue with them is that they detect the forgery with low accuracy. Hence, there is a need for a better framework that is fast and more accurate. To address this, we presented a novel image recompression-based system. Apart from achieving better image forgery detection accuracy, our proposed framework has also achieved faster response time. This makes it suitable for real-life applications, as it is more accurate and can be utilized even by slower machines. The proposed framework is detailed in the next section.

3. Proposed Technique

CNNs, which are inspired by the human visual system, are designed to be non-linear interconnected neurons. They have already demonstrated extraordinary potential in a variety of computer vision applications, including image segmentation and object detection. They may be beneficial for a variety of additional purposes, including image forensics. With the various tools available today, image forgery is fairly simple to do, and because it is extremely dangerous, detecting it is crucial. When a fragment of an image is moved from one to another, a variety of artifacts occur due to the images' disparate origins. While these artifacts may be undetectable to the naked eye, CNNs may detect their presence in faked images. Due to the fact that the source of the forged region and the background images are distinct, when we recompress such images, the forged is enhanced differently due to the compression difference. We use this concept in the proposed approach by training a CNN-based model to determine if an image is genuine or a fake.

A region spliced onto another image will most likely have a statistically different distribution of DCT coefficients than the original region. The authentic region is compressed twice: first in the camera, and then again in the fake, resulting in periodic patterns in the histogram [2]. The spliced section behaves similarly to a singly compressed region when the secondary quantization table is used.

As previously stated, when an image is recompressed, if it contains a forgery, the forged portion of the image compresses differently from the remainder of the image due to the difference between the source of the original image and the source of the forged portion. When the difference between the original image and its recompressed version is analyzed, this considerably emphasizes the forgery component. As a result, we use it to train our CNN-based model for detecting image forgery.

Algorithm 1 shows the working of the proposed technique, which has been explained here. We take the forged image A (images shown in Figure 1b tamper images), and then recompress it; let us call the recompressed image as $A_{recompressed}$ (images shown in Figure 1c are recompressed forged images). Now we take the difference of the original image and the recompressed image, let us call it A_{diff} (images shown in Figure 1e are the difference of Figure 1b,c, respectively). Now due to the difference in the source of the forged part and the original part of the image, the forged part gets highlighted in A_{diff} (as we can observe in Figure 1d,e, respectively). We train a CNN-based network to categorize an image as a forged image or a genuine one using A_{diff} as our input features (we label it as a featured image). Figure 2 gives the pictorial view of the overall working of the proposed method.

To generate $A_{recompressed}$ from A, we use JPEG compression. Image A undergoes JPEG compression and produces $A_{recompressed}$ as described in Figure 3. When there is a single compression, then the histogram of the dequantized coefficients exhibits the pattern as shown in Figure 4, this type of pattern is shown by the forged part of the image. Moreover, when there is a sort of double compression then, as described in Figure 5, there is a gaping between the dequantized coefficients as shown in Figure 6, this type of pattern is shown by the genuine part of the image.

We constructed a very light CNN model with minimal parameters in our proposed model (line number 5 to 13 of Algorithm 1). We constructed a model consisting of 3 convolutional layers after which there is a dense fully connected layer, as described below:

- The first convolutional layer consists of 32 filters of size 3-by-3, stride size one, and "relu" activation function.
- The second convolutional layer consists of 32 filters of size 3-by-3, stride size one, and "relu" activation function.
- The third convolutional layer consists of 32 filters of size 7-by-7, stride size one, and "relu" activation function, followed by max-pooling of size 2-by-2.

- Then we have the dense layer that has 256 neurons with "relu" activation function, finally which is connected to two neurons (output neurons) with "sigmoid" activation.

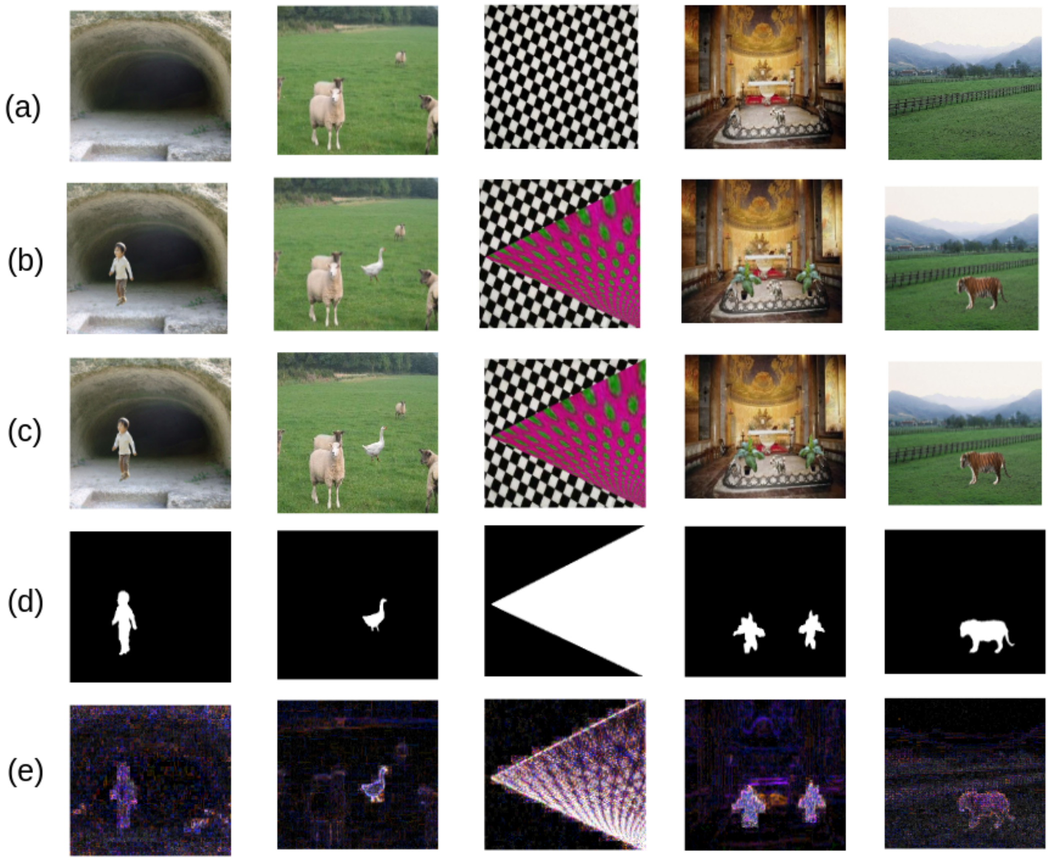


Figure 1. Various sample images and their processed forms: (**a**) the original images (RGB colour format); (**b**) the images with forgery (RGB colour format); (**c**) the recompressed forged images (RGB colour format); (**d**) ground truth of forgery (Binary format); (**e**) difference of the tampered image with its recompressed image (RGB colour format).

Table 1. Details of the CASIA.2.0 image forgery database.

	Genuine Images	Tampered Images	Total Images
CASIA.2.0	7491	5123	12,614
Training (80%)	5993	4098	10,091
Testing (20%)	1498	1025	2523

The feature image (A_{diff}) is resized to 128 × 128 ($A_{reshaped\_diff}$) and then fed to the network. The network learns the presence of any tampering present through the feature images (images shown in Figure 1e). During training, the proposed model learns the existence of the forgery in an image through the numerous artifacts left behind during image forgery. The trained model can identify tampering with high accuracy, discussed in the next section.

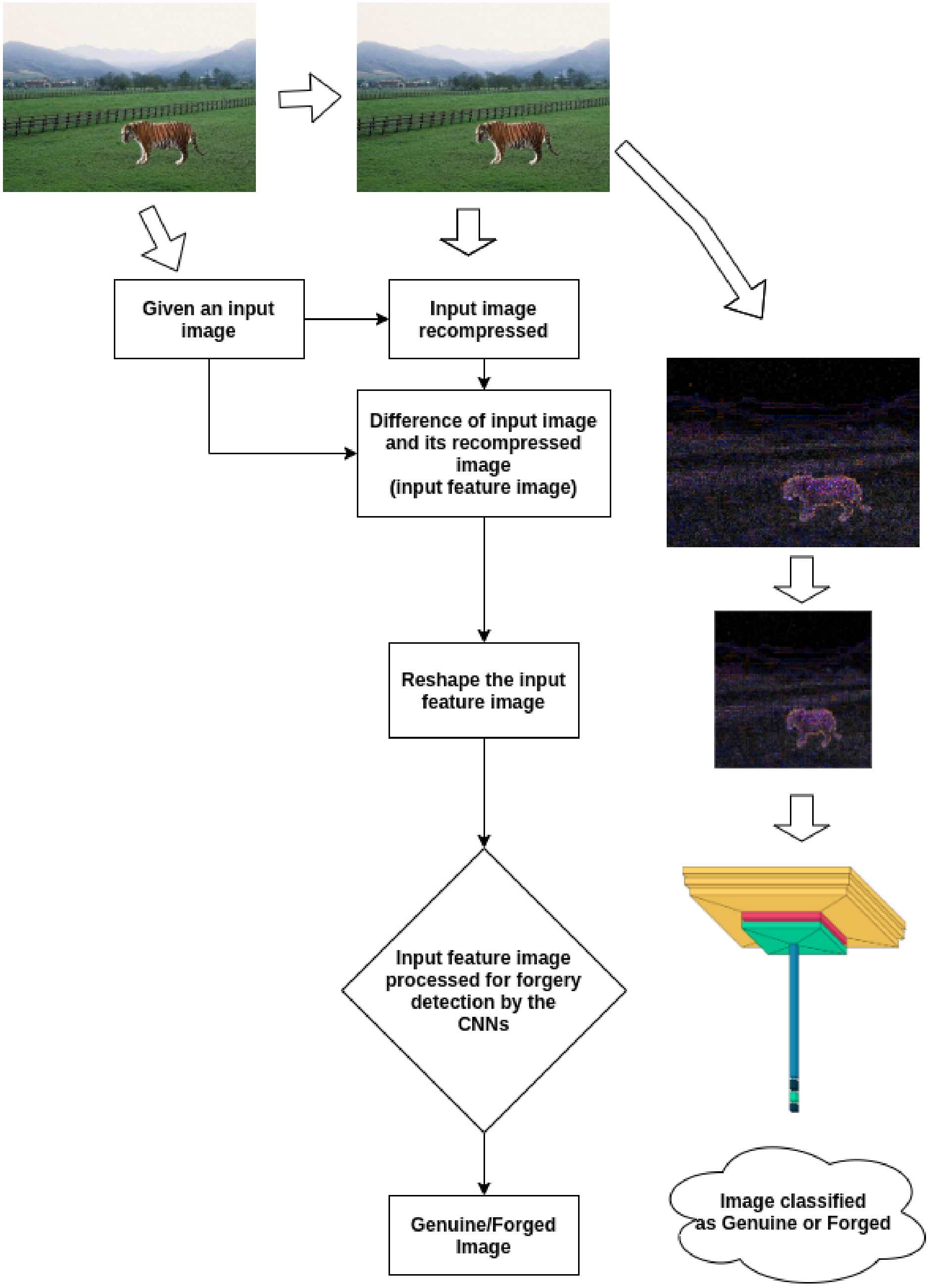

Given an input image
Input image recompressed
Difference of input image and its recompressed image (input feature image)
Reshape the input feature image
Input feature image processed for forgery detection by the CNNs
Genuine/Forged Image
Image classified as Genuine or Forged

Figure 2. Flowchart of the proposed work.

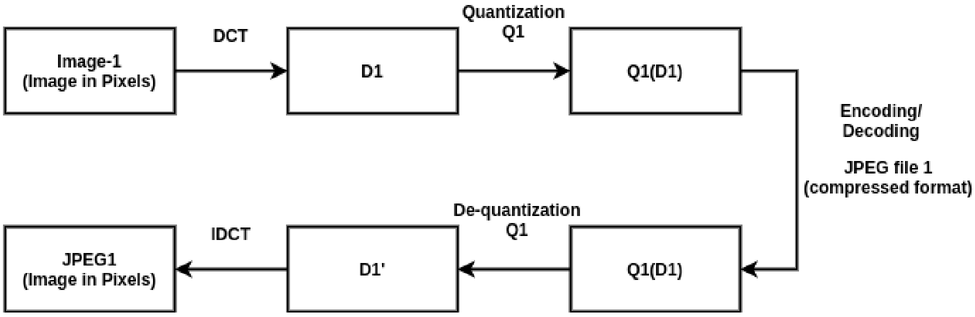


Figure 3. JPEG compression on image pixels, first DCT is applied on the image pixel blocks followed by the quantization. Then decompression of the compressed image is done with through de-quantization followed by IDCT, to obtain the image in pixel format.

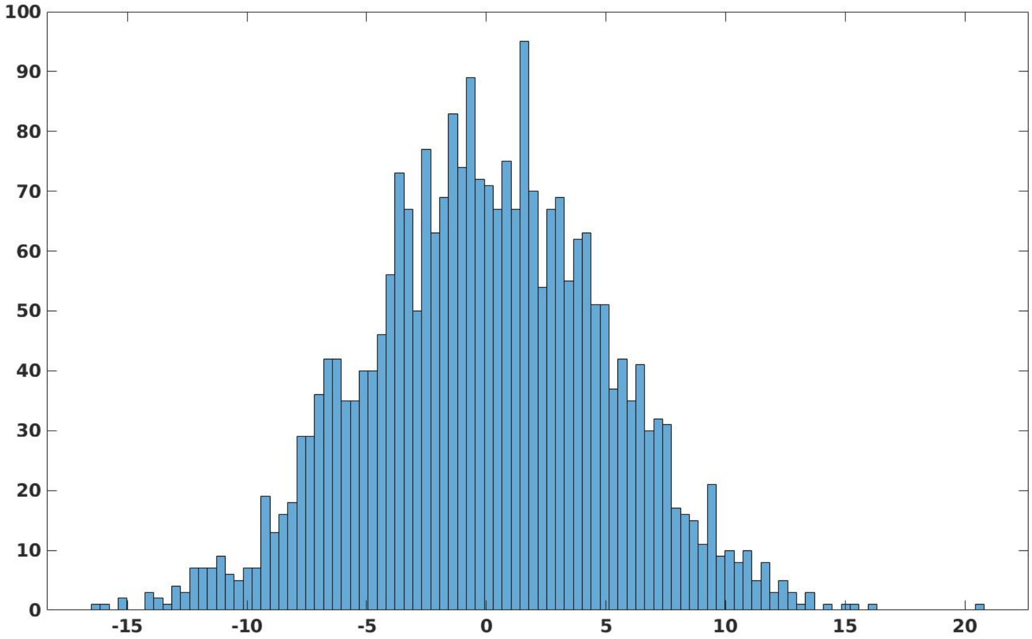


Figure 4. The histogram of DCT coefficients of the forged region in the image, which behaves as singly compressed.

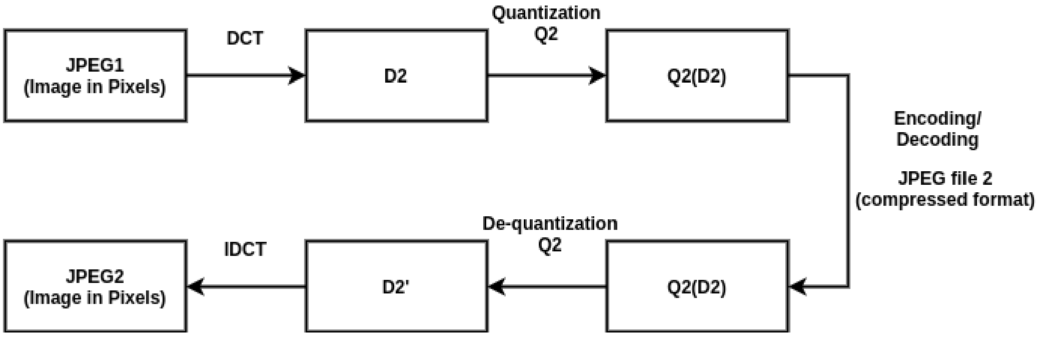


Figure 5. The recompression of the genuine region of the image.

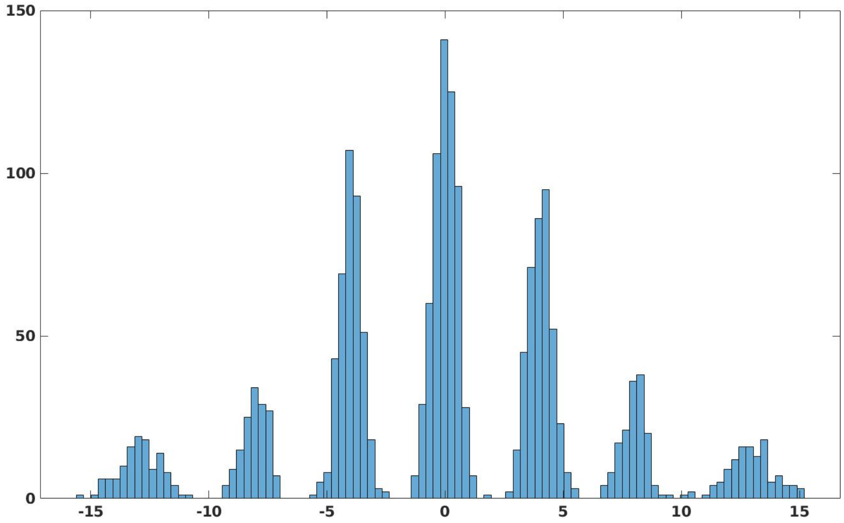


Figure 6. The histogram of DCT coefficients of the genuine region of an image which gets double compressed.

Algorithm 1: Working of the proposed technique for the image forgery detection.

```
 1: /* Model Training (line 2 to 23) */
 2: Input: Image '$A_i$' ($i$ = 1 to $n$), with labels '$L_i$' ($L_i$ = 1 if $A_i$ is tampered image, else $L_i$ = 0).
 3: Output: Trained Model: Image_Forgery_Predictor_Model()

 4: /* Prediction Model Description */
 5: Image_Forgery_Predictor_Model(image with size 128 × 128 × 3)
 6: {
 7:     First convo. layer: 32 filters (size 3 × 3, strid size one, activation: "relu")
 8:     Second convo. layer: 32 filters (size 3 × 3, strid size one, activation: "relu")
 9:     Third convo. layer: 32 filters (size 3 × 3, strid size one, activation: "relu")
10:     Max-pooling of size 2 × 2
11:     Dense layer of 256 neurons with "relu" activation function
12:     Two neurons (output neurons) with "sigmoid" activation
13: }

14: for $epochs$ = 1 to $total\_epochs$ do
15:     $training\_error = 0$
16:     for $i$ = 1 to $n$ do
17:         $A_{recompressed\_i} = JPEG_{Compression}(A_i, Q)$
18:         $A_{diff\_i} = A_i - A_{recompressed\_i}$
19:         $A_{reshaped\_diff\_i} = reshape(A_{diff\_i}, (128, 128, 3))$
20:         $training\_error$ = ($L_i$ − Image_Forgery_Predictor_Model($A_{reshaped\_diff\_i}$)) + $training\_error$
21:     end for
22:     modify_model($training\_error$, Image_Forgery_Predictor_Model(), Adam_optimizer)
23: end for

24: /* Image forgery prediction (line 25 to 32) */
25: Input: Image '$Input\_Image$'
26: Output: '$Input\_Image$' labelled as tampered or untampered
27: $Input\_Image_{recompressed} = JPEG_{Compression}(Input\_Image, Q)$
28: $Input\_Image_{diff} = Input\_Image - Input\_Image_{recompressed}$
29: $Input\_Image_{reshaped\_diff} = reshape(Input\_Image_{diff}, (128, 128, 3))$
30: $Predicted\_label$ = Image_Forgery_Predictor_Model($Input\_Image_{reshaped\_diff}$)
31: /* If $Predicted\_label$ [0][0]>$Predicted\_label$ [0][1], then $Input\_Image$ is tampered
32: /* If $Predicted\_label$ [0][1]>$Predicted\_label$ [0][0], then $Input\_Image$ is untampered
```

4. Experimental Results and Discussion

This section describes the training and testing environment for the proposed approach. Aside from that, we'll examine and contrast its performance with that of other techniques.

4.1. Experimental Setup

We examined the proposed technique on a popular CASIA 2.0 image forgery database [22,49], to evaluate how efficient it is. There are a total of 12,614 images (in BMP, JPG, and TIF format), out of which 7491 are genuine images and 5123 tamper images. CASIA 2.0 includes images from various categories, including animals, architecture, articles, characters, plants, nature, scenes, textures, and indoor images. There are different-different sizes of the images present in the database; the resolution of images varies from 800 × 600 pixels to 384 × 256 pixels. The details about the CASIA 2.0 database are given in Table 1. A processor (Intel(R) Core(TM) i5-2400 CPU @ 3.1 GHz) having 16 GB RAM has been used for the experimentation.

Following terms are initially calculated for the evaluation:

- $Total\_Images$: The total number of images that were tested.
- T_P (true positive): Correctly identified tampered images.
- T_N (true negative): Correctly identified genuine images.
- F_N (false negative): Wrongly identified tampered images, the tampered images which have been identified as genuine images.
- F_P (false positive): Wrongly identified genuine images, the genuine images which have been identified as tampered images.

We calculate the accuracy, precision, recall, and $F_{measure}$ [1] for the evaluation and the comparison of the proposed method with others. These are calculated as given below:

Now the accuracy is defined as given below:

$$Accuracy = \frac{T_P + T_N}{T_{Total\_Images}} \times 100$$

$$Recall = \frac{T_P}{T_P + F_N}$$

$$Precision = \frac{T_P}{T_P + F_P}$$

$$F_{measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \times 100$$

4.2. Model Training and Testing

To evaluate the proposed technique, we randomly divided the CASIA 2.0 database in the ratio of 80% and 20% (Table 1), we used 80% of the images (5993 authentic images, 4099 tampered images, total 10,092 images) for training the model. We used Adam optimizer with an initial learning rate of 1×10^{-5} and a batch size of 64. The remaining 20% images (1498 genuine images, 1024 tampered images, total 2522 images) are for testing the proposed model and comparing it with the other existing frameworks. Figure 7 illustrates the training and testing accuracy of the proposed model when trained on the CASIA 2.0 database with the settings mentioned above.

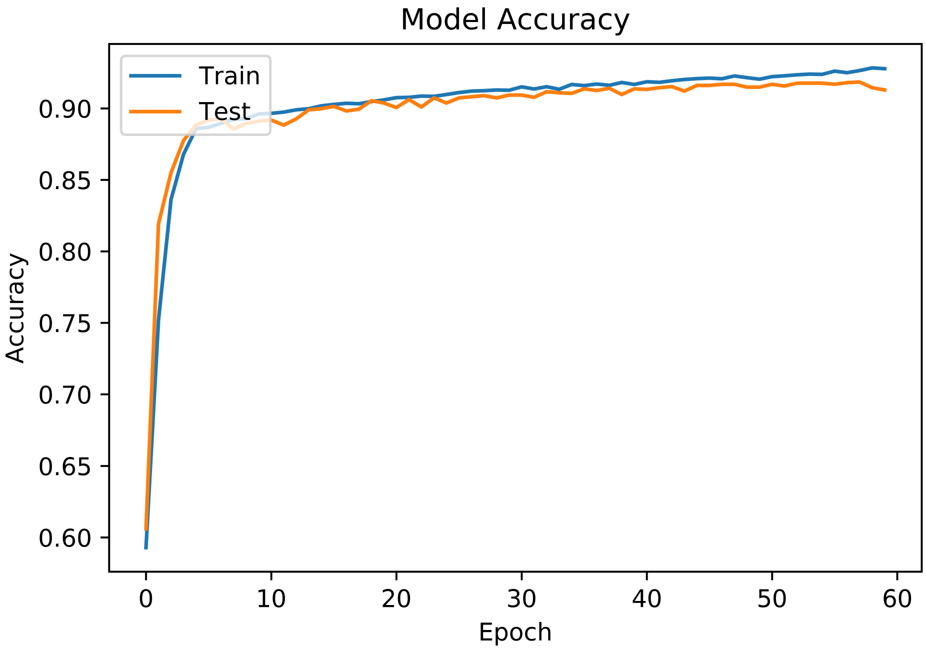


Figure 7. The training and the testing accuracy of the proposed method for the image forgery detection.

4.3. Comparison with Other Techniques

We compare the proposed technique to the other techniques in terms of accuracy and time required for image forgery detection.

4.3.1. Accuracy Comparison

Table 2 shows the image forgery detection accuracy by the various techniques. These techniques and the proposed method have been evaluated on the same set of images of the CASIA 2.0 database. For the techniques mentioned in [2,3,27], when these techniques process an input image, then, if the mask generated from them reports forgery, then we categorize the input image as tampered else; it is considered as genuine image. It must be noted that Buster-Net [27] is basically for copy-move type image forgeries, so we have used copy-move forged images to report its results. Whereas CAT-Net [2] is basically for splicing type of image forgeries, so we have used the images with splicing to report its results. Mantra-Net [3] can handle both image splicing and copy-move type image forgeries. It can be observed that we chose techniques that can handle either image splicing or copy-move, but also techniques that can handle both image splicing and copy-move type of image forgeries. All the techniques have CASIA 2.0 database as a common database for the evaluation. We have used these techniques' publicly available trained model for evaluation. Apart from this, we retrained their models on the same CASIA 2.0 database images, on which the proposed model has been trained. The results obtained by retraining these models are also given in Table 2, along with their original models. After retraining, these techniques' accuracy has improved; however, the proposed technique still outperforms them. CAT-Net [2], Buster-Net [27], and Mantra-Net [3] concentrate more on where the forgery is present (localization, where the output is pixel-level forgery detection) in the given image rather than focusing on that is the image is tampered or genuine (detection, where the output is a binary classification). However, the proposed technique focuses on whether the given image is tampered with or genuine.

The proposed technique achieved better forgery detection accuracy due to the fact that instead of directly using the original pixel image, it uses the feature image, which is the difference of the image with its recompressed image. This helps to detect image forgery better because it can be observed that in the feature image the forged part gets highlighted. Hence, it has resulted in achieving high accuracy. On the other hand, [2,3,27] show poor accuracy in image forgery detection as these techniques try to find image forgery at the pixel level, and due to this there are false positive pixels reported which reduces their overall forgery detection accuracy at the image level.

As mentioned in the previous section, we used JPEG compression to recompress the image; now, various quality factors are available while recompressing the image. So we have evaluated the proposed model for different JPEG quality factors and reported them as well. It is observed that the accuracy is better if the quality factor is kept at more than 90. The proposed technique achieved better accuracy as it utilizes better input features rather than directly using the original image as input features. To verify this we have trained our model by directly using the original images (instead of the better processed features), and its results are also reported in Table 2. It can be observed that in such a scenario the accuracy of the model drops from 92.23% to 72.37%, this shows the effectiveness of the processed input features (the difference of the original image with its recompressed version). Figure 8 show the comparison of the accuracy and the $F_{measure}$ for the proposed method and the other techniques.

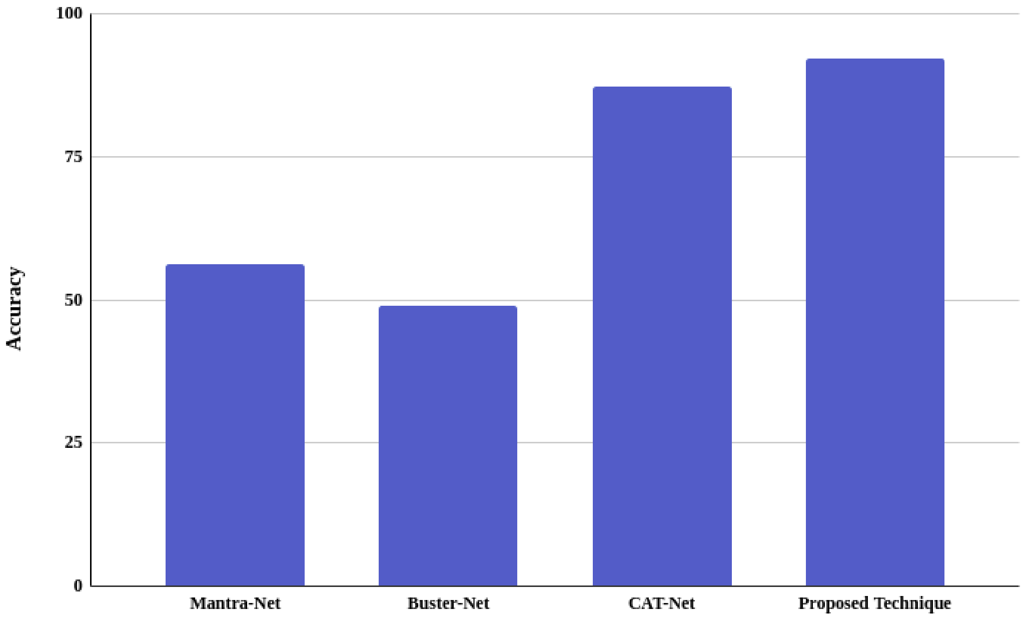
Figure 9 shows a few visual examples demonstrating the proposed approach's performance. The codes and the test images used by the proposed technique are available at codes (https://github.com/sadaf-ali/Image-forgery-detection-using-deeplearning-by-recompressing-the-images, accessed on 23 January 2022).

We have acquired 92.23% accuracy on the test images when the recompressed images' quality factor (JPEG compression) is 98. Compared to the other techniques, our technique achieved much better performance in detecting the presence of the image forgery in the image. It must also be noted that our method can handle both image splicing and copy-move types of image forgeries. In contrast, many of the state-of-the-art techniques can handle only one type of image forgery, and very few can handle both types of image forgery.

Table 2. Accuracy comparison of the proposed technique with other techniques on CASIA 2.0 database.

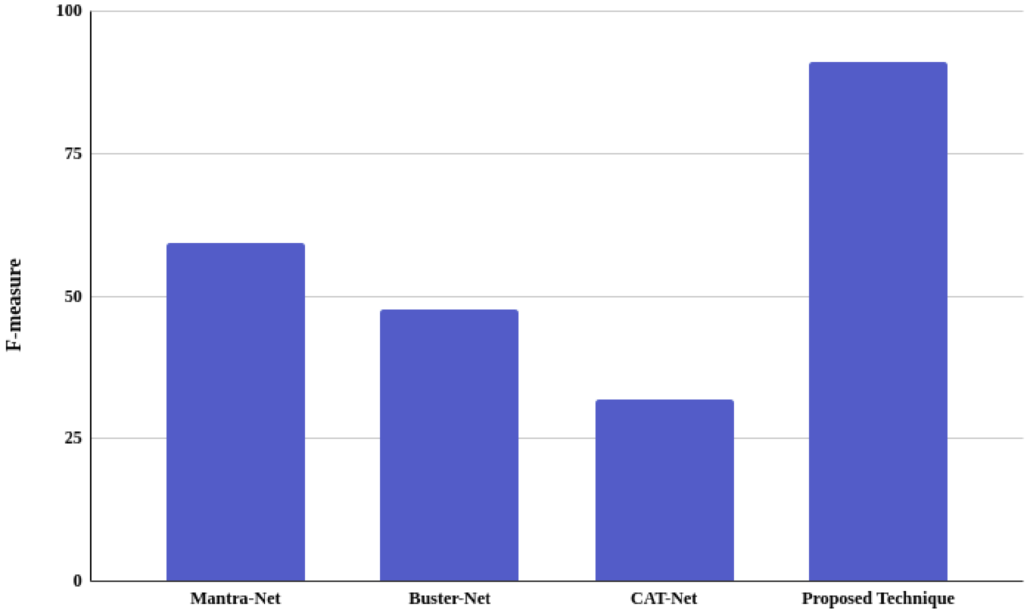
Techniques	Accuracy	Precision	Recall	$F_{measure}$
Mantra-Net [3]	40.62	0.40	1	57.14
Mantra-Net * [3]	56.14	0.48	0.79	59.33
Buster-Net [27]	30.77	0.30	1	46.75
Buster-Net * [27]	49.06	0.35	0.77	47.74
CAT-Net [2]	19.33	0.19	1	31.93
CAT-Net * [2]	87.29	0.62	0.87	72.76
Proposed technique (with normal image)	72.37	0.62	0.79	70.11
Proposed technique (85% quality)	87.71	0.80	0.92	85.99
Proposed technique (90% quality)	91.31	0.83	0.98	90.19
Proposed technique (91% quality)	91.35	0.83	0.97	90.14
Proposed technique (92% quality)	91.39	0.84	0.96	90.15
Proposed technique (93% quality)	90.84	0.83	0.96	89.52
Proposed technique (94% quality)	91.71	0.84	0.97	90.50
Proposed technique (95% quality)	91.83	0.84	0.97	90.68
Proposed technique (96% quality)	92.15	0.85	0.97	90.99
Proposed technique (97% quality)	91.63	0.85	0.96	90.32
Proposed technique (98% quality)	**92.23**	**0.85**	**0.97**	**91.08**
Proposed technique (99% quality)	91.83	0.86	0.94	90.42
Proposed technique (100% quality)	91.79	0.86	0.94	90.29

Note: "*" means that the model has been retrained on CASIA 2.0 database.



(**a**)

Figure 8. *Cont.*



(**b**)

Figure 8. Comparison of the performance of Mantra-Net [3], Buster-Net [27], CAT-Net [2], and the proposed technique in terms of: (**a**) accuracy, (**b**) $F_{measure}$.

4.3.2. Processing Time Comparison

Table 3 shows the average time taken by the various techniques to process an image and to predict whether it is a genuine image or a tampered one. In terms of predicting the presence of forgery in an image, it can be noted that the proposed technique is fast and more efficient than the other state-of-the-art techniques. Figure 10 pictorially shows the comparison of the average time taken by the proposed method and the other techniques for the forgery detection in an image.

This is because we provide an efficient feature image to our model, and the proposed CNN-based model is relatively light compared to other techniques. As a result, it can provide predictions in a much shorter amount of time. This makes our model will be advantageous in real-world scenarios. Table 4 shows the comparison of the proposed technique with the other techniques.

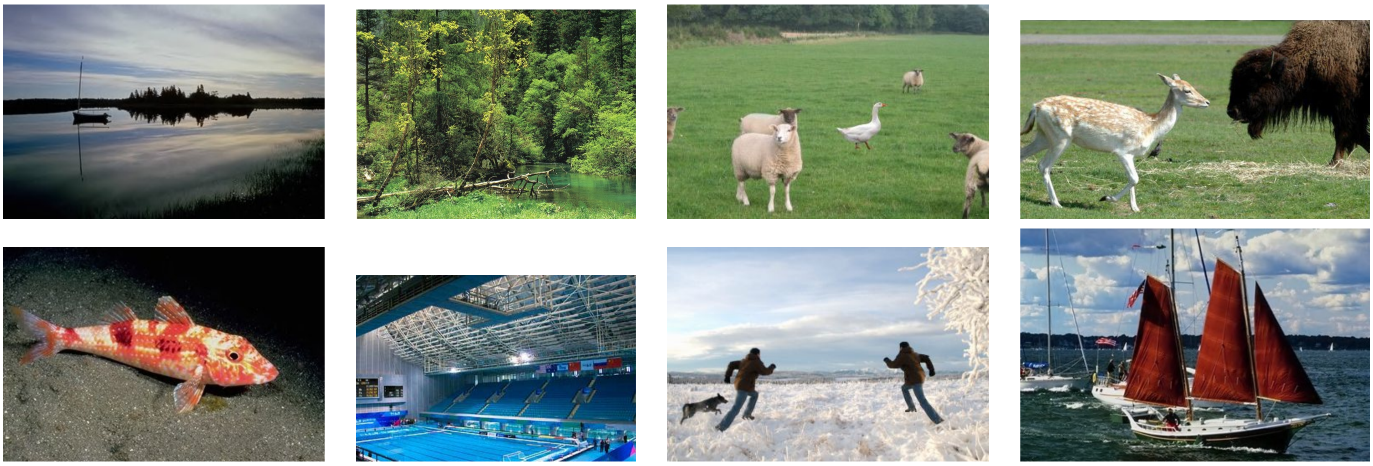

(**a**) (**b**) (**c**) (**d**)

Figure 9. A few visual examples demonstrating the proposed approach's performance: (**a**) true negative cases, (**b**) false negative cases, (**c**) true positive cases, (**d**) false positive cases.

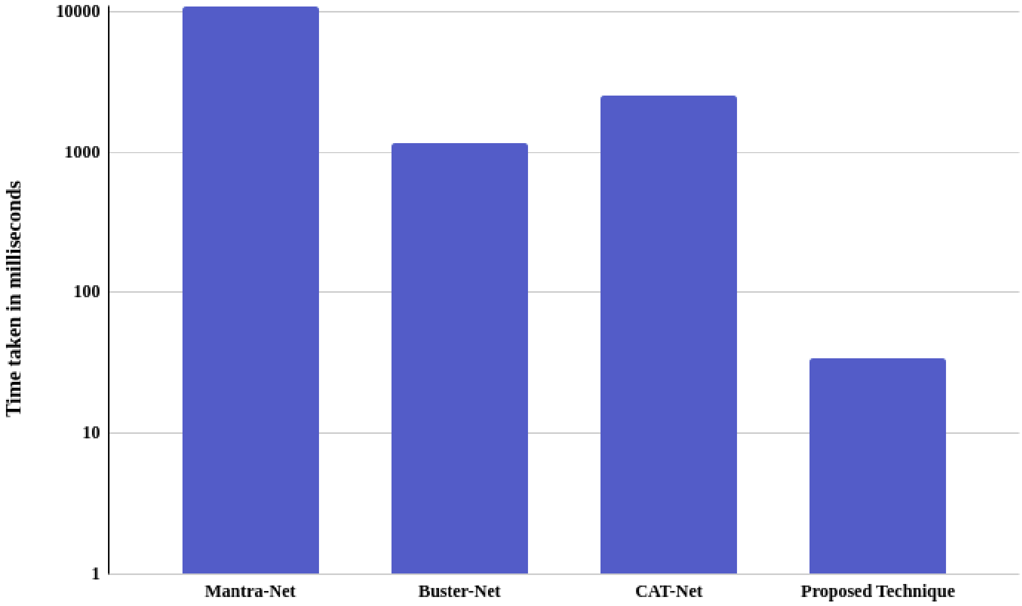


Figure 10. Average time taken by Mantra-Net [3], Buster-Net [27], CAT-Net [2], and the proposed technique to process the image for forgery detection.

Table 3. Comparison of the time to process an image by the proposed technique with other techniques on CASIA 2.0 database (displayed values are in milliseconds).

Techniques	Time Taken
Mantra-Net [3]	10,927
Buster-Net [27]	1160
CAT-Net [2]	2506
Proposed technique	**34**

Table 4. Comparison of the proposed technique with the other techniques [2,3,27].

Sr. No.	Proposed Technique	Other Techniques
1	It focuses on whether the given image is tampered with or genuine (image level forgery detection).	They focus more on where the forgery is present in the given image (pixel level forgery detection).
2	It uses feature image (difference of original image with its recompressed image), in the feature image the forged part gets highlighted, which helps to detect image forgery better.	They directly use the image in the original pixel format, which makes image forgery detection difficult.
3	It is utilizes enhanced features, due to which it is able to handle both image splicing and copy-move types image forgeries.	Most of the techniques do not utilize the enhanced features, due to which they are usually able to handle only one type of image forgery.
4	It attains a high accuracy in classifying the images as tampered or not.	They do pixel level forgery detection. Due to this, they suffer from false positive pixels. Hence, there is degradation in classifying the images as tampered or not.
5	It is much faster, hence, it is suitable for slow machines as well.	These techniques are much slower, hence, they are not suitable for the slower machines.

Hence, from the experimental results, the following observations can be made:

- Unlike other techniques, the proposed technique works well for both image splicing and copy-move types of image forgeries.
- It is highly efficient for image forgery detection and has exhibited significantly better performance than the other techniques.
- The difference in the compression of the forged part and the genuine part of the image is a good feature that can be learned by our CNN based model efficiently, which makes the proposed technique more robust in comparison to the other techniques.
- The proposed model is much faster than the other techniques, making it ideal and suitable for real-world usage, as it can be implemented even on slower machines.

5. Conclusions and Future Work

The increased availability of cameras has made photography popular in recent years. Images play a crucial role in our lives and have evolved into an essential means of conveying information since the general public quickly understands them. There are various tools accessible to edit images; these tools are primarily intended to enhance images; however, these technologies are frequently exploited to forge the images to spread misinformation. As a result, image forgery has become a significant problem and a matter of concern. In this paper, we provide a unique image forgery detection system based on neural networks and deep learning, emphasizing the CNN architecture approach. To achieve satisfactory results, the suggested method uses a CNN architecture that incorporates variations in image compression. We use the difference between the original and recompressed images to train the model. The proposed technique can efficiently detect image splicing and copy-move types of image forgeries. The experiments results are highly encouraging, and they show that the overall validation accuracy is 92.23%, with a defined iteration limit.

We plan to extend our technique for image forgery localization in the future. We will also combine the suggested technique with other known image localization techniques to improve their performance in terms of accuracy and reduce their time complexity. We will enhance the proposed technique to handle spoofing [50] as well. The present technique requires image resolution to be a minimum of 128 × 128, so we will enhance the proposed technique to work well for tiny images. We will also be developing a challenging extensive image forgery database to train deep learning networks for image forgery detection.

Author Contributions: S.S.A. and I.I.G. designed and performed the experiments, conceptualization and methodology, and analyzed the data. S.S.A. and I.I.G. wrote the manuscript in consultation with N.-S.V., N.S., S.D.A. and N.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by CY Cergy Paris Université (CY initiative via contract number 2019-067-C01-A0) and Khalifa University.

Acknowledgments: We are thankful to CNRS for providing us the GPU Cluster Jean Zay (http://www.idris.fr/jean-zay/ accessed on 10 January 2022) to train our network.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiao, B.; Wei, Y.; Bi, X.; Li, W.; Ma, J. Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. *Inf. Sci.* **2020**, *511*, 172–191. [CrossRef]
2. Kwon, M.J.; Yu, I.J.; Nam, S.H.; Lee, H.K. CAT-Net: Compression Artifact Tracing Network for Detection and Localization of Image Splicing. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2021; pp. 375–384.
3. Wu, Y.; Abd Almageed, W.; Natarajan, P. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9535–9544.
4. Ali, S.S.; Baghel, V.S.; Ganapathi, I.I.; Prakash, S. Robust biometric authentication system with a secure user template. *Image Vis. Comput.* **2020**, *104*, 104004. [CrossRef]
5. Castillo Camacho, I.; Wang, K. A Comprehensive Review of Deep-Learning-Based Methods for Image Forensics. *J. Imaging* **2021**, *7*, 69. [CrossRef] [PubMed]

6. Zheng, L.; Zhang, Y.; Thing, V.L. A survey on image tampering and its detection in real-world photos. *J. Vis. Commun. Image Represent.* **2019**, *58*, 380–399. [CrossRef]
7. Jing, L.; Tian, Y. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1. [CrossRef]
8. Meena, K.B.; Tyagi, V. Image Forgery Detection: Survey and Future Directions. In *Data, Engineering and Applications: Volume 2*; Shukla, R.K., Agrawal, J., Sharma, S., Singh Tomer, G., Eds.; Springer: Singapore, 2019; pp. 163–194.
9. Mirsky, Y.; Lee, W. The Creation and Detection of Deepfakes: A Survey. *ACM Comput. Surv.* **2021**, *54*, 1–41. [CrossRef]
10. Rony, J.; Belharbi, S.; Dolz, J.; Ayed, I.B.; McCaffrey, L.; Granger, E. Deep weakly-supervised learning methods for classification and localization in histology images: A survey. *arXiv* **2019**, arXiv:abs/1909.03354.
11. Lu, Z.; Chen, D.; Xue, D. Survey of weakly supervised semantic segmentation methods. In Proceedings of the 2018 Chinese Control Furthermore, Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1176–1180.
12. Zhang, M.; Zhou, Y.; Zhao, J.; Man, Y.; Liu, B.; Yao, R. A survey of semi- and weakly supervised semantic segmentation of images. *Artif. Intell. Rev.* **2019**, *53*, 4259–4288. [CrossRef]
13. Verdoliva, L. Media Forensics and DeepFakes: An Overview. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 910–932. [CrossRef]
14. Luo, W.; Huang, J.; Qiu, G. JPEG Error Analysis and Its Applications to Digital Image Forensics. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 480–491. [CrossRef]
15. Matern, F.; Riess, C.; Stamminger, M. Gradient-Based Illumination Description for Image Forgery Detection. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1303–1317. [CrossRef]
16. Christlein, V.; Riess, C.; Jordan, J.; Riess, C.; Angelopoulou, E. An Evaluation of Popular Copy-Move Forgery Detection Approaches. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1841–1854. [CrossRef]
17. Habibi, M.; Hassanpour, H. Splicing Image Forgery Detection and Localization Based on Color Edge Inconsistency using Statistical Dispersion Measures. *Int. J. Eng.* **2021**, *34*, 443–451.
18. Dua, S.; Singh, J.; Parthasarathy, H. Image forgery detection based on statistical features of block DCT coefficients. *Procedia Comput. Sci.* **2020**, *171*, 369–378. [CrossRef]
19. Ehret, T. Robust copy-move forgery detection by false alarms control. *arXiv* **2019**, arXiv:1906.00649.
20. de Souza, G.B.; da Silva Santos, D.F.; Pires, R.G.; Marana, A.N.; Papa, J.P. Deep Features Extraction for Robust Fingerprint Spoofing Attack Detection. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 41–49. [CrossRef]
21. Balsa, J. Comparison of Image Compressions: Analog Transformations. *Proceedings* **2020**, *54*, 37. [CrossRef]
22. Pham, N.T.; Lee, J.W.; Kwon, G.R.; Park, C.S. Hybrid Image-Retrieval Method for Image-Splicing Validation. *Symmetry* **2019**, *11*, 83. [CrossRef]
23. Bunk, J.; Bappy, J.H.; Mohammed, T.M.; Nataraj, L.; Flenner, A.; Manjunath, B.; Chandrasekaran, S.; Roy-Chowdhury, A.K.; Peterson, L. Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1881–1889.
24. Bondi, L.; Lameri, S.; Güera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. Tampering Detection and Localization Through Clustering of Camera-Based CNN Features. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1855–1864.
25. Yousfi, Y.; Fridrich, J. An Intriguing Struggle of CNNs in JPEG Steganalysis and the OneHot Solution. *IEEE Signal Process. Lett.* **2020**, *27*, 830–834. [CrossRef]
26. Islam, A.; Long, C.; Basharat, A.; Hoogs, A. DOA-GAN: Dual-Order Attentive Generative Adversarial Network for Image Copy-Move Forgery Detection and Localization. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4675–4684.
27. Wu, Y.; Abd-Almageed, W.; Natarajan, P. BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
28. Wu, Y.; Abd-Almageed, W.; Natarajan, P. Image Copy-Move Forgery Detection via an End-to-End Deep Neural Network. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Munich, Germany, 8–14 September 2018; pp. 1907–1915.
29. Liu, X.; Liu, Y.; Chen, J.; Liu, X. PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization. *arXiv* **2021**, arXiv:2103.10596
30. Wei, Y.; Bi, X.; Xiao, B. C2R Net: The Coarse to Refined Network for Image Forgery Detection. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy in Computing Furthermore, Communication, New York, NY, USA, 1–3 August 2018; pp. 1656–1659.
31. Bi, X.; Wei, Y.; Xiao, B.; Li, W. RRU-Net: The Ringed Residual U-Net for Image Splicing Forgery Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June 2019.
32. Younisand, Y.; Xu, M.; Qiao, T.; Wu, Y.; Zheng, N. Image Forgery Detection and Localization via a Reliability Fusion Map. *Sensors* **2020**, *20*, 6668.
33. Abdalla, Y.; Iqbal, M.T.; Shehata, M. Convolutional Neural Network for Copy-Move Forgery Detection. *Symmetry* **2019**, *11*, 1280. [CrossRef]

34. Kniaz, V.V.; Knyaz, V.; Remondino, F. The Point Where Reality Meets Fantasy: Mixed Adversarial Generators for Image Splice Detection. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2019; Volume 32.
35. Mayer, O.; Stamm, M.C. Forensic Similarity for Digital Images. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1331–1346. [CrossRef]
36. Huh, M.; Liu, A.; Owens, A.; Efros, A.A. Fighting Fake News: Image Splice Detection via Learned Self-Consistency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
37. Zhang, R.; Ni, J. A Dense U-Net with Cross-Layer Intersection for Detection and Localization of Image Forgery. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2982–2986.
38. Liu, Y.; Guan, Q.; Zhao, X.; Cao, Y. Image Forgery Localization Based on Multi-Scale Convolutional Neural Networks. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 85–90.
39. Bi, X.; Liu, Y.; Xiao, B.; Li, W.; Pun, C.M.; Wang, G.; Gao, X. D-Unet: A Dual-encoder U-Net for Image Splicing Forgery Detection and Localization. *arXiv* **2020**, arXiv:2012.01821.
40. Marra, F.; Gragnaniello, D.; Cozzolino, D.; Verdoliva, L. Detection of GAN-Generated Fake Images over Social Networks. In Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 10–12 April 2018; pp. 384–389.
41. Kadam, K.; Ahirrao, D.S.; Kotecha, D.K.; Sahu, S. Detection and Localization of Multiple Image Splicing Using MobileNet V1, 2021. *arXiv* **2020**, arXiv:2108.09674.
42. Jaiswal, A.A.K.; Srivastava, R. Image Splicing Detection using Deep Residual Network. In Proceedings of the 2nd International Conference on Advanced Computing and Software Engineering (ICACSE), San Francisco, CA, USA, 13–15 October 2019.
43. Dang, H.; Liu, F.; Stehouwer, J.; Liu, X.; Jain, A.K. On the Detection of Digital Face Manipulation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5780–5789.
44. Nguyen, H.H.; Fang, F.; Yamagishi, J.; Echizen, I. Multi-task Learning for Detecting and Segmenting Manipulated Facial Images and Videos. In Proceedings of the 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 23–26 September 2019; pp. 1–8.
45. Li, Y.; Lyu, S. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Nashville, TN, USA, 19–25 June 2019.
46. Komodakis, N.; Gidaris, S. Unsupervised representation learning by predicting image rotations. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
47. Wang, L.; Li, D.; Zhu, Y.; Tian, L.; Shan, Y. Dual Super-Resolution Learning for Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3773–3782.
48. Yu, L.; Zhang, J.; Wu, Q. Dual Attention on Pyramid Feature Maps for Image Captioning. *arXiv* **2022**, arXiv:2011.01385.
49. Dong, J.; Wang, W.; Tan, T. CASIA Image Tampering Detection Evaluation Database. In Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 422–426.
50. Ali, S.S.; Iyappan, G.I.; Prakash, S. Fingerprint Shell construction with impregnable features. *J. Intell. Fuzzy Syst.* **2019**, *36*, 4091–4104. [CrossRef]