



Article Detecting Images in Two-Operator Series Manipulation: A Novel Approach Using Transposed Convolution and Information Fusion

Saurabh Agarwal ^{1,2}, Dae-Jea Cho¹ and Ki-Hyun Jung ^{1,*}

- ¹ Department of Software Convergence, Andong National University, Gyeongbuk 36729, Republic of Korea; saurabhnsit2510@gmail.com (S.A.); djcho@anu.ac.kr (D.-J.C.)
- ² Amity School of Engineering & Technology, Amity University Uttar Pradesh, Noida 201313, India
- * Correspondence: kingjung@anu.ac.kr; Tel.: +82-54-820-7968

Abstract: Digital image forensics is a crucial emerging technique, as image editing tools can modify them easily. Most of the latest methods can determine whether a specific operator has edited an image. These methods are suitable for high-resolution uncompressed images. In practice, more than one operator is used to modify image contents repeatedly. In this paper, a reliable scheme using information fusion and deep network networks is presented to recognize manipulation operators and the operator's series on two operators. A transposed convolutional layer improves the performance of low-resolution JPEG compressed images. In addition, a bottleneck technique is utilized to extend the number of transposed convolutional layers. One average pooling layer is employed to preserve the optimal information flow and evade the overfitting concern among the layers. Moreover, the presented scheme can detect two operator series with various factors without including them in training. The experimental outcomes of the suggested scheme are encouraging and better than the existing schemes due to the availability of sufficient statistical evidence.

Keywords: image forensics; deep neural network; image manipulation detection; image forgery detection



Citation: Agarwal, S.; Cho, D.-J.;

Jung, K.-H. Detecting Images in

Two-Operator Series Manipulation: A

Convolution and Information Fusion.

Novel Approach Using Transposed

Summetry 2023, 15, 1898. https://

doi.org/10.3390/sym15101898

Academic Editor: Deming Lei

Received: 20 June 2023

Revised: 11 August 2023

Accepted: 23 August 2023 Published: 10 October 2023

(†)

Copyright: © 2023 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/

(cc)

4.0/).

1. Introduction

Due to the accessibility of high-accuracy yet simple-to-use image-altering tools, digital photographs are frequently the target of modification. Image forensics is required to determine the image's origin, processing history, and veracity. There are numerous ways [1–3] to identify the image source device. Since a fake image is typically constructed using two or more photographs, the mismatch of sources aids in identifying image forgery. The majority of fake images use several processes to appear genuine. Finding fake images is made simple by the identification of actions like median filtering [4,5], sharpening [6,7], and resizing [8–10]. It is also possible to concurrently identify the image forgery processes using various universal methods [11–17]. Some schemes [18–20] are discussed to detect image forgery rather than for detecting an operator. Schemes [18,19] also trace the external objects in the fake image. Method [20] is suitable both for splicing and copy-move forgery detection.

Nonetheless, general techniques identify one operator processing accurately on the image. The image is subjected to multiple operations in a practical scenario. In this paper, a series of procedures may be precisely identified to provide the image processing timeline. There are not many methods [21–26] that can identify the operations and their series. Although performance varies depending on the operation, JPEG compression—the most popular format—plays a crucial role in forensic investigation. When JPEG compression is considered, the performance of recent approaches is degraded.

Convolutional neural networks have shown favorable outcomes in many applications in the current period of a deep learning network. The identification of resizing, brightness changes, median filtering, general image alteration, manifold JPEG compression, image forgery, etc., uses a deep network. For the first time, a deep network was introduced [11] to detect additive white noise, scaling, median, and Gaussian filtering. In particular, the starting layer of the deep network uses a constraint. Outcomes from the experiment are provided for 227 \times 227 size pictures. However, no results were offered for images of JPEG-compressed and small sizes. With the aid of a better CNN model [12], the proposed concept of constraint is further expanded. A constraint convolutional layer is trailed by four groups of layers in the enhanced deep network, each of which has a convolutional layer, batch normalization layer, rectified linear unit layer, and pooling layer. Additionally, multiple SoftMax classification layers are employed. A randomized tree classifier is utilized to categorize the results. By deducting the outcome value from the filter window's center value, constrained convolutional layer filters to guess mistakes. Each iteration's training phase involves enforcing the constraint. When two operations are operated on an image in succession, even on a high-resolution image, the performance of the CNN model generally suffers. Boroumand and Fridrich [13] proposed a deep network and multilayer perceptron exemplary to identify four operations— denoising, tone modification, low-pass, and highpass filtering. In the proposed CNN, eight convolutional layers are employed. Moments are calculated in the end stage of the deep network and are used by the multilayer perceptron to classify the images. The manual feature extraction method is contrasted with the previously discussed technique. Only 512×512 size images are covered in the experiments. The calculation of the out-of-bag error by Li et al. [14] allowed them to choose a few sub-models from the SRM. The selection procedure may significantly lower the feature dimension. Eleven image processes, including spatial filtering, image enhancement, and JPEG compression, are examined in the results. The proposed method also promises the successful identification of four anti-forensic procedures, including median filtering, resampling, contrast enhancement, and JPEG compression. However, for small-size images, the performance suffers. A deep network with two convolutional layers was introduced by Singhal et al. [15] to detect seven different sorts of operations. The deep network uses the discrete cosine transform factors of the median kernel residual as the input array. The convolutional layer employs large dimension kernels. The Siamese network was used by Xue et al. [16] to identify activities such as adding text, an emblem, and a black chunk to an image and operators like Gamma correction, Gaussian noise, and image resampling. The Siamese network used ResNet-18 and AlexNet. Uncompressed images are taken into account in experiments. Image procedures like median filtering, image scaling, and histogram equalization were discovered by Barni et al. [17]. Two neural networks are used to extract the features [11,27]. The robust characteristics are chosen from the CNN network using a random feature selection strategy, and a support vector machine classifier is then used to determine the kind of attack.

Detecting the order of image operations is a crucial concern when comprehensively analyzing the history of image processing. Various efforts have been made to determine the correct sequence of operations applied to an image [21–26] to address this challenge. These research endeavors aim to develop methodologies and techniques to accurately and automatically identify the specific order in which image processing operations were applied. Researchers and practitioners can gain valuable insights into the image's processing history by successfully determining the operation order. This information is essential for understanding the transformations and manipulations that an image has undergone, which is particularly important in fields like forensics, image analysis, and restoration.

The cited references [21–26] likely represent a collection of previous works in image processing and forensics that have contributed to the ongoing efforts to solve this challenging problem. Exploring these prior studies helps build on existing knowledge and lays the foundation for further image operation order detection advancements. As researchers continue to investigate and refine these techniques, they move closer to achieving more accurate and reliable solutions for unraveling the history of image processing operations. A framework based on mutual knowledge is proposed in [21,22] to analyze the causes of the

operator series order's non-detection. Some operator series are impossible for the algorithm to recognize. The prior approach fails to identify JPEG-compressed images. Comesaa [23] has discussed operator order detection's theoretical potential. In order to estimate the order of operations, Bayar and Stamm [24] deliberated on a deep network that contains a constrained convolution layer. Liao et al. [25] discussed a dual-stream deep network to identify the operators and their corresponding sequences. The approach claimed the detection of an operator with unknown parameters exhausting transfer learning, though tailored preprocessing is necessary to employ for a particular operator. Cho et al. [26] proposed detecting operators and their respective orders. In this scheme, tailored preprocessing is not required, although the detection performance can be improved by considering some modifications in the deep network.

In this paper, a scheme that can guarantee improved performance on two-operator chain series detection is proposed. The following are key points of the proposed network's specific contributions:

- The proposed scheme can detect an operated image of two-operator and the operation series. The successful detection of many operations include Gaussian blurring, median filtering, unsharp masking, and image upscaling;
- The transposed convolutional layer is considered instead of the convolutional layer to reduce the classification error. As the proposed scheme is suitable for challenging scenarios like the usage of small-size images;
- The bottleneck strategy helps to lower the training parameters. Therefore, the proposed method using a bottleneck strategy can insert more transposed convolutional layers into the convolutional neural network;
- The pooling layer is avoided among the convolutional layers to save the most statistical data possible. Subsequently, it might reduce the computing expense at the overhead of pertinent inherited operation impressions;
- Information fusion is applied to the features of trained networks using multiple optimizers. Information fusion enhances performance drastically;
- Without specific preprocessing requirements, the proposed method can guarantee improved performance in demanding situations with low-resolution compressed images and two-operator series manipulation.

The rest of the paper is organized as follows. A problem for two-operator manipulation detection in various contexts is formulated in Section 2. Section 3 explains the proposed scheme. In Section 4, a comprehensive experimental analysis is carried out along with a comparative analysis. In Section 5, the benefits and limitations of the proposed scheme are emphasized as conclusions.

2. Formulation of Two-Operator Series Problem

In this section, two topics are covered to discuss the significance of detecting operator series. First, the issue of operator series detection is examined for uncompressed images. Moreover, the history of compressed image processing will be uncovered in the second.

2.1. Uncompressed Image Operator Series

The uncovering of two-operator series can be considered a numerous class categorization issue. The following are the five categories that are based on processing history and two operators, u & v:

- C1: Non-processed image;
- C2: An image is processed using the operator *u*;
- C3: An image is processed using the operator v;
- C4: An image is first processed using operator u and then processed using operator v;
- C5: An image is first processed using operator v and then processed using operator u.

Image quality analysis is performed to recognize the complexity of the two-operator series detection. The image quality provides an overview of the image statistics. Any form

of the operator will cause changes in the statistics. One image is considered to comprehend how an image evolves after being subjected to several operators. An unprocessed image of the BOSSBase 1.01 [28] dataset is displayed as the first image in Figure 1. The remaining images are GAB_1.0, MDF_5 × 5, GAB_1.0, MDF_5 × 5, MDF_5 × 5, and GAB_1.0 operated images. The five images can also be defined as C1, C2, C3, C4, and C5 class images for $u = GAB_{-1.0}$ and $v = MDF_{-5} × 5$. GAB_1.0 represents Gaussian blurring (standard deviation = 1.0), and MDF_5 × 5 represents a median filter of size 5 × 5 correspondingly. The images displayed in the figure cannot easily be distinguished because of their small size. Therefore, the Perception-based Image Quality Evaluator (PIQE) [29] quality parameter is taken to show the effect of operators. PIQE assesses distortion without requiring training data because it is a no-reference and opinion-aware quality indicator. For quality prediction, PIQE uses local feature extraction. Only relevant spatial areas are used perceptually in the quality calculation to mirror human behavior. PIQE is the best for the UNF image (a small quantity represents a good-quality image).



UNF PIQE = 37.65



Figure 1. UNF and filtered images.

PIQE = 72.48



MDF_5 × 5 PIQE = 66.06



GAB_1.0 MDF_5 × 5 PIQE = 77.26



MDF_5 × 5 GAB_1.0 PIQE = 88.13

One more quality indicator is also considered. Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [30] is a no-reference quality indicator that captures the point-wise statistical of localized standardized luminance impulses and assesses the degree of picture naturalness based on observed variances from a natural image's statistical model. Detailed qualitative analysis is performed on ten thousand images of dimension 64×64 pixels using BRISQUE and PIQE (Figure 2). These images are created by taking the central pixels of the BOSSBase dataset. The BRISQUE shows that there is little difference in the five class images. PIQE displays a significant difference between unfiltered and other processed images, although the difference between filtered images is less when compared with filtered images.



Figure 2. Image quality analysis.

The pair of image scaling and Gaussian blurring processes are detected in one trial [22]. The frequency domain is used to visualize the image features. However, the outcomes could be better for the operator series. The C4 and C5 classes cannot be detected. The artifacts of one operator in a two-operator series can be muted by another, even though each operator's strength differs.

2.2. JPEG Compressed Image Operator Series

JPEG is a popular image format. JPEG format is typically used as the default of images for digital devices. A forged image is typically produced using JPEG files. The creation of the forged image involves several processes. The forged image contains double JPEG compression artifacts since it needs to be stored again. The process of creating the forged image involves double JPEG compression. As a result, the following five categories can be created using JPEG compression of quality factors, QF1 & QF2, and two operators, u & v:

- C1: An image is compressed with QF1, though not treated by any operator;
- C2: An image is compressed with QF1 and treated by the operator *u*, and then compressed with QF2;
- C3: An image is compressed with QF1 and treated by operator *v*, and then compressed with QF2;
- C4: An image is treated by *u*, compressed with QF1, and the image is treated again by *v*, and then compressed with QF2;
- C5: An image is treated by *v*, compressed with QF1, and the image is treated again by *u*, and then compressed with QF2.

The operator artifacts are reduced via JPEG compression. The intricacy of the issue can increase with two operators and double JPEG compression.

In this paper, a general operator identifier is put forth for different parameters, which is a more realistic case. The proposed deep model can handle a single operator and a series of two operators. The unlike parameters are taken into account within a specific range. By the proposed CNN, the proposed technique can automatically train features to recognize operator series. Traditional machine learning requires handcrafted feature extraction and selection, which is unnecessary for the proposed scheme. According to earlier research, the detection of different operators requires different preprocessing. The proposed method does not call for a preprocessing of the images. However, some earlier studies [15,25] claimed that preprocessing was necessary. The proposed CNN can successfully classify the five categories mentioned above and draw attention to the statistical anomaly.

3. The Proposed Scheme

Deep networks have established their value in solving various problems, including image categorization, fake face identification, and image forgery detection. The detection of one and two operators in operated images is discussed in this paper, along with a resilient deep architecture and information fusion. In a compressed and uncompressed scenario, the proposed scheme performs well. As with some earlier methods [15,25], the proposed CNN design can eliminate the requirement of any preprocessing layer. According to the operator, an exclusive preprocessing was necessary for the earlier methodologies, but this needed to be practicable and limited the network performance for specific operators. When two processes are performed simultaneously, the second operator may reduce the artifacts of the first operator. Various pairs of operators are considered to examine how operations on the BOSSBase [28] image database behave as in Figure 3. Five operators are taken into account: Gaussian blurring (GAB_1.0), median filtering of filter size 5×5 $(MDF_5 \times 5)$, unsharp masking $(USM_3.0)$, and upscaling $(USL_1.5)$. The covariance plot of the entropy is considered for uncompressed and compressed images for two scenarios to visualize the behavior of five categories of images. The covariance plot estimates the power spectral density (power/frequency) of a discrete-time signal (entropy values for our problem) discovered by the covariance approach. Each column's power spectral density is calculated separately. The estimate for the power spectral density is calculated using

an autoregressive model. Uncompressed images are considered in the first row as per the discussion of Section 2.1. JPEG compressed images are taken with QF1 = 75 and QF2 = 85 in the second row as per the discussion of Section 2.2. The gap between some lines is less, and overlapping is more in compressed images (first image of the second row) than in uncompressed images (first image of the first row) for operators $u = GAB_{1.0}$ and $v = MDF_{5} \times 5$. This fact is also reflected in the experimental analysis, i.e., the classification error is more in compressed images compared to uncompressed images. A similar behavior is also followed by operators $u = USM_{3.0}$ and $v = USL_{1.5}$. Therefore, a common solution can be proposed to deal with this issue.



Figure 3. Effect of operators on images while considering entropy.

The proposed architecture is better able to withstand the operator series issue. CNN uses a number of layers and kernels to divide the different types of images into categories. Figure 4 displays the framework of the proposed CNN. The resulting feature map from the transposed convolution has a higher spatial dimensionality than the feature map from the input image. The standard convolution reduces the input dimension by employing sliding convolutional kernels. By flattening the input and output, we can represent the convolution operation as $Z = M^*X + S$, where Z is the output, M is the convolution matrix, and S is the bias vector. These parameters are obtained from the layer's weights and biases.



Figure 4. Framework of the proposed deep network.

On the other hand, the transposed convolution is employed to expand the input using sliding convolutional kernels. The process involves adding padding to all the edges of the input, where the padding size is determined by subtracting one from the kernel's edge size. This is performed to achieve upsampling instead of downsampling. When both the input and output are flattened, the transposed convolution can be equivalently expressed as $Z = M^{T*}X + S$. A conventional convolution layer's backward function can be compared to this process.

The transposed convolutional (TConv) layer overrode the results of a typical convolutional layer by retaining the connection pattern. Thus, the original input is returned, in contrast to how a typical convolutional layer functions. The padding is applied to the result rather than the input image in the transposed convolutional layer. Padding is applied to the output in the transposed convolution rather than the input in regular convolution. Transposed convolution is the regular convolution reversed, but only by dimension, not by value. The proposed CNN has twelve transposed convolutional layers. The bottleneck approach is followed. In some applications, such as image steganalysis [31], the bottleneck technique produces superior results compared to the standard approach. Figure 5 shows an abstract representation of the bottleneck technique. The batch normalization (Batch Norm) and the rectified linear unit (ReLU) layers trail two consecutive TConv layers. Pointwise convolution is carried out in the first TConv layer, and depth-wise convolutions are performed by the second TConv layer in two successive TConv layers. When used with a depth-wise convolution in steganalysis [32], 1×1 point-wise convolution enhances the outcomes. Using a 1×1 filter and a 3×3 filter in that order helps lessen the computational complexity for TConv layers. Experiments also reveal a performance improvement. The percentage detection error using TConv 1×1 is less than (approximately 2%) TConv 3×3 while considering a 1×1 filter and a 3×3 filter order compared to a 3×3 filter and a 1×1 filter order. TConv 1×1 training takes less time than TConv 3×3 training, and the percentage detection error also decreases. Therefore, using the bottleneck technique has two substantial advantages.



Figure 5. Bottleneck approach.

The network architecture consists of transposed convolution layers, specifically the first and second, with eighty filters each. These filters have sizes of 1×1 and 3×3 , respectively. The number of filters in each block matches the 1×1 filter size in the 3×3 transposed convolutional layers. The transposed convolutional layers have a stride of one.

To enhance the training process, several techniques are employed. First, the network initialization sensitivity is reduced, which helps stabilize the training. Additionally, a Batch Norm layer is utilized, which accelerates the training rate and reduces the inner covariant shift [33]. The Batch Norm layer updates the learning parameters based on the mean and variance of each mini-batch during training. Once the training is complete, the Batch Norm layer's final mean and variance values predict unseen data.

A ReLU layer [34] is applied after the Batch Norm layer to improve the network's performance. This layer substitutes negative values with zeros, which enhances the network's ability to learn and generalize.

The proposed network uses one global average pooling (GAP) layer because the internal statistical information details are vital, and the image size is small. The GAP layer increases accuracy in steganalysis [35,36]. One element is obtained using GAP from each feature map. The activation function follows the GAP layer. The activation function extracts features from the trained deep network while considering different optimizers. The information fusion process combines the feature vectors obtained from the activation function, as illustrated in Figure 6.



Figure 6. Information fusion.

In the feature vector extraction phase, three different optimizers are employed: Optimizer 1 (Adam), Optimizer 2 (RMSprop), and Optimizer 3 (SGDM). The training process runs for 50 epochs to prevent any unfair bias towards unseen data, and the data are shuffled before each epoch. A learning rate of 0.01 is considered. These optimizers are used to train the network and extract essential features from the data.

The Global Average Pooling (GAP) layer is utilized to process the feature vectors, proceeded by SoftMax and classification layers. The experimental paper demonstrates that including the GAP layer reduces the percentage classification error by up to 2.7%. Additionally, the GAP layer preserves the operation fingerprints and mitigates the overfitting problem [37]. Although multiple pooling layer experiments were conducted, only one GAP layer is considered in the final experimental analysis. Given that the last transposed convolutional layer in the proposed CNN has 24 filters, the GAP layer generates 24 features as a result.

The fully connected layer is critical in consolidating all the knowledge acquired from the preceding layers. It combines the extracted features to make a comprehensive decision. The SoftMax function is applied to handle the output of the fully connected layer, assigning probabilities to each class. The crucial characteristic of the SoftMax function is that the total likelihood across all categories must equal 1, ensuring that it represents a valid probability distribution. Consequently, the classification layer employs a cross-entropy loss function to determine the exclusive class for classification.

The importance of CNN's weight initialization cannot be overstated, as it significantly impacts the network's overall performance. In the prior stage, random values are collected for network initialization. However, this method is impractical, leading to inconsistent performance across different runs due to varying weight initializations. Glorot and Bengio [38] introduced a weight initialization strategy that improves performance and speeds up convergence to address this issue. This strategy works particularly well with less dense networks like the proposed CNN. The weights are initialized based on the number of inputs and hidden nodes, promoting more stable training and better generalization. For classification purposes, an SVM (Support Vector Machine) classifier is employed, as illustrated in Figure 7. This classifier takes the features extracted by CNN and uses them to classify the input data into specific classes.



Figure 7. Training and testing process.

4. Experimental Analysis

In this paper, a robust scheme is proposed to detect processed images of single and two operators with their sequence. Numerous experiments are run to verify the resilience and adaptability of the proposed network. A total of twenty-six thousand images are taken in equal proportion from BOSSBase [28], UCID [39], LIRMM [40], and never-compressed (NC) [41] image databases, which contain 10,000, 1338, 10,000, and 5150 uncompressed color images, respectively, to build the experimental dataset. First, the 256×256 pixel middle block of each image is used, and then sixteen non-overlapping blocks with a dimension of 64×64 pixels are produced. In the end, 416,000 patches with a dimension of 64×64 pixel images are created. Each class uses twenty-four thousand images for training and six thousand images for validation. Tests are conducted on fifteen thousand image patches. The image patches that are used in training are never reused in testing. In the experimental paper, five operators are taken into account: Gaussian blurring (GAB_X), median filtering of filter sizes 3×3 and 5×5 (MDF_3 $\times 3$, MDF_5 $\times 5$), unsharp masking (USM_X), and upscaling (USL_X) with various X parameters. The operator is applied to the image while taking into account symmetric padding. It is crucial to note that thirty thousand image patches of size 64×64 pixels are chosen randomly to obtain unbiased findings for each operator. Using an NVIDIA GTX2070 GPU and 32 GB RAM, the experiments are conducted. As discussed in Section 2.1, the C1 class stands refers to the original picture, the C2 class refers to images processed by operator u, the C3 class refers to images processed by operator v, the C4 class refers to images processed by operator u followed by operator v, and the C5 class refers to images processed by operator v followed by operator u. Likewise, the classes are defined for compressed images, as described in Section 2.2. The proposed scheme can categorize two-operator processed images in their order. The proposed scheme's classification error is less than the existing schemes. T1, T2, and T3 signify when the proposed CNN is trained using Adam, RMSprop, and SGDM optimizers. The SoftMax classifier is utilized for T1, T2, and T3. TF represents the features of T1, T2, and T3 when they are concatenated. The SVM classifier with a linear kernel is considered for TF.

Table 1 displays specific data for the percentage detection error for various operator pairings. In Table 1, the following operators are taken into consideration: Gaussian blurring with standard deviation 0.7 (GAB_0.7), 1.0 (GAB_1.0), median filtering of filter size 3×3 (MDF_3 $\times 3$), and 5×5 (MDF_5 $\times 5$). The mean percentage detection error (MPDE) and standard deviation of percentage (STD) detection error are also defined for T1, T2, T3, and TF. In the case of GAB_1.0, the classification errors are lesser, though GAB_0.7 is high. Due to the low blurring of images, misclassification between different classes increased. The TF provides the least MPDE, and the standard deviation is the lowest for TF.

U	GAB_1.0				GAB_1.0				GAB_0.7			
v	$MDF_3 \times 3$				$MDF_5 \times 5$				$MDF_5 \times 5$			
	T1	T2	T3	TF	T1	T2	T3	TF	T1	T2	T3	TF
C1	0.24	0.16	0.37	0.09	0.35	0.24	0.41	0.13	0.97	0.80	0.69	0.39
C2	3.72	2.96	2.81	1.75	1.19	1.07	0.69	0.51	0.53	0.53	0.55	0.33
C3	0.47	0.52	1.52	0.43	8.83	3.35	3.06	2.67	15.07	15.95	12.73	11.06
C4	2.58	1.17	1.18	0.55	1.07	3.05	3.02	1.33	6.81	5.33	9.11	6.61
C5	1.53	1.62	1.87	0.83	0.46	0.60	0.59	0.29	0.14	0.26	0.27	0.19
MPDE	1.71	1.29	1.55	0.73	2.38	1.66	1.56	0.99	4.70	4.58	4.67	3.72
STD	1.46	1.09	0.89	0.63	3.63	1.44	1.36	1.05	6.41	6.69	5.85	4.93

Table 1. Operator series analysis of Gaussian blurring and median filtering.

Table 2 considers several operators, including GAB_0.7, GAB_1.0, and unsharp masking with a 2.0 radius (USM_2.0) and 3.0 radius (USM_3.0). Among all the scenarios, the classification error is the highest in class C5, which corresponds to the combined application of USM and GAB (USM GAB). This observation is supported by the confusion matrix of GAB_1.0 and USM_3.0 for the TF (transfer function), as shown in Figure 8. The confusion matrix reveals that a significant number of 1290 images belonging to class C5 are misclassified as class C1 (GAB_1.0). Similarly, 579 images from class C1 (GAB_1.0) are misclassified as class C5. However, despite these classification errors, the information fusion technique (TF) still yields the best overall results.

Table 3 shows the analyses of various operators, including GAB_0.7, GAB_1.0, and upscaling with factors 1.2 and 1.5. Among these scenarios, TF (transfer function) consistently outperforms other operators and demonstrates superior stability in delivering the results. The Mean Percentage Difference Errors (MPDEs) associated with TF are significantly lower than the other operator series presented in Table 2. Specifically, when considering GAB_1.0 in conjunction with upscaling, the results show that using USL_1.5 as the upscaling factor yields better outcomes than USL_1.2. The upscaling process introduces interpolation, which affects the intrinsic statistical evidence of the data, making the classification process easier. However, it is noteworthy to mention that the classification error is lower when using GAB_1.0 and USL_1.5 together than when using GAB_1.0 and USL_1.2. This is because, as seen in USL_1.5, high-factor upscaling disturbs the intrinsic statistical evidence to a lesser extent than low-factor upscaling, as observed in USL_1.2. The interference caused by high-factor upscaling is comparatively less, leading to improved classification accuracy for the combination of GAB_1.0 and USL_1.5.

U GAB_1.0 GAB_1.0 GAB_0.7 v USM_2.0 USM_3.0 USM_3.0 T1 TF TF T2 **T**3 T1 T2 **T**3 TF T1 T2 **T**3 C1 2.39 1.83 4.75 1.90 6.37 1.81 2.71 1.74 4.17 2.92 4.65 2.56 C2 3.34 3.88 9.20 5.40 8.87 5.49 4.79 3.89 3.71 3.18 1.53 2.89 C3 1.79 2.75 6.63 5.01 2.14 2.59 6.03 3.04 2.22 4.415.31 6.22 C4 0.09 0.19 0.05 0.05 0.09 0.23 1.09 0.55 0.42 0.140.06 1.11 C5 18.00 14.12 9.35 9.01 10.66 13.13 11.11 9.04 9.55 11.31 5.98 8.61 MPDE 6.09 5.01 5.12 3.79 5.34 6.19 4.79 3.70 4.49 4.30 4.96 2.92 STD 7.06 5.42 4.13 3.49 4.53 5.03 4.103.27 2.87 3.38 4.141.99

Table 2. Operator series analysis of Gaussian blurring and unsharp masking.

	C1	C2	C3	C4	C5
C1	14,739	4	253	3	1
C2	3	14,417	0	1	579
C3	329	0	14,667	4	0
C4	6	0	3	14,991	0
C5	1	1290	0	1	13,708

Figure 8. Confusion matrix of u = GAB_1.0 and v = USM_3.0.

Table 3. Operator series analysis of Gaussian blurring and upscalir	Tab	ole 3.	C)perator	series	anal	ysis	of	G	aussian	b	lurring	; and	u	pscal	in	g.
--	-----	--------	---	----------	--------	------	------	----	---	---------	---	---------	-------	---	-------	----	----

U	GAB_1.0					GAB_1.0				GAB_0.7				
V	USL_1.2					USL	1.5		USL_1.5					
	T1	T2	T3	TF	T1	T2	T3	TF	T1	T2	T3	TF		
C1	0.09	0.11	0.15	0.07	0.13	0.07	0.15	0.05	1.04	0.41	0.53	0.31		
C2	6.18	6.50	15.89	7.67	3.58	3.73	4.34	2.47	0.67	0.64	0.25	0.22		
C3	0.24	0.47	0.31	0.15	0.23	0.14	0.17	0.12	3.85	3.49	2.25	1.45		
C4	0.17	0.13	0.17	0.11	0.08	0.55	0.44	0.08	0.69	0.43	0.63	0.70		
C5	7.75	7.98	2.08	4.09	1.29	1.08	0.94	1.27	0.03	0.08	0.07	0.03		
MPDE	2.88	3.04	3.72	2.42	1.06	1.11	1.21	0.80	1.26	1.01	0.75	0.54		
STD	3.77	3.87	6.85	3.41	1.49	1.52	1.78	1.07	1.50	1.40	0.87	0.56		

Table 4 evaluates three operators: MDF, USM, and USL. As with the findings in Tables 1–3, the transfer function (TF) consistently outperforms all other operators across different scenarios, providing more stable and reliable results. In particular, when the combination of $u = MDF_3 \times 3$ and $v = USM_2.0$ is used, TF achieves an impressive MPDE of 3.36%. Despite a slightly larger MDF kernel ($u = MDF_5 \times 5$) while maintaining $v = USM_2.0$, TF still performs well, resulting in a 5.03% MPDE. However, it is worth noting that the outcomes are less favorable when using the combination of $u = MDF_5 \times 5$ and $v = USL_{1.2}$, which leads to inferior results compared to other manipulation types. Additionally, the classification errors are exceptionally high for the C2 and C5 classes, indicating

that these classes pose significant challenges for the evaluated operators and manipulation combinations. Overall, Table 4 reaffirms the superiority of the transfer function in achieving superior performance and stability across diverse scenarios. Furthermore, it underscores the importance of selecting appropriate operator combinations to optimize classification results effectively.

U	U MDF_ 3×3					$MDF_5 \times 5$			$MDF_5 \times 5$				USM_2.0			
V	USM_2.0				USM	USM_2.0			USL_1.2				USL_1.5			
Class	T1	T2	T3	TF	T1	T2	T3	TF	T1	T2	T3	TF	T1	T2	T3	TF
C1	2.83	1.77	1.77	2.31	3.29	1.69	4.11	1.92	0.11	0.16	0.19	0.05	1.75	1.51	1.13	2.13
C2	7.07	4.82	4.82	5.19	9.00	6.75	12.53	8.49	24.83	38.22	14.05	19.13	5.33	5.52	8.05	2.47
C3	4.75	6.17	6.17	2.34	4.15	5.66	2.93	2.56	0.26	0.37	0.17	0.06	21.13	9.79	9.99	5.17
C4	1.10	1.08	1.08	0.20	0.19	0.47	0.17	0.11	0.09	0.14	0.31	0.09	1.25	6.83	10.95	7.72
C5	9.01	11.13	11.13	6.73	14.66	19.24	11.80	12.07	13.25	6.71	39.11	14.26	0.38	0.15	0.12	0.01
MPDE	4.95	4.99	4.99	3.36	6.26	6.76	6.31	5.03	7.71	9.12	10.77	6.72	5.97	4.76	6.05	3.50
STD	3.18	4.02	4.02	2.59	5.66	7.45	5.54	5.04	11.12	16.51	16.94	9.27	8.68	3.94	5.07	2.99

Table 4. Operator series analysis of median filtering, unsharp masking, and upscaling.

The JPEG format is frequently used as the default format in a real-world camera setting, as the graphic quality is still decent afterward compression. As a result, three stages are taken into account while detecting the operator series in JEPG images. The image is compressed with the QF1 factor in the first step. In step 2, the operator series is applied to the compressed images. Step 3 involves using JPEG compression with the QF2 quality factor. In Section 2.2, a comprehensive description of JPEG compression is provided. Compared to uncompressed images, the proposed CNN's performance can suffer from compressed images. However, performance is acceptable considering the modest image size (64×64) and low compression quality aspects.

The outcomes of compressed images are presented in Table 5. Numerous compression quality variables are considered in the actual scenario, QF1 = QF2, QF1 < QF2, and QF1 > QF2. The difference between quality factors QF1 and QF2 is variable ranging from 5 to 20. The operator's artifacts are diminished during compression. However, the average percentage of detection error is less than 9% in most cases. In the first case, QF1 = 75 and QF2 = 85, and in the second case, QF1 = 85 and QF2 = 75; the two cases with *u* = GAB_1.0 and *v* = MDF_5 × 5 are taken into consideration. The percentage of detection error in the first case is 2.32%, as shown in Table 5, while it is 3.89% in the second. The percentage of detection error of TF is less than 5%, even when QF1 = 90 and QF2 = 70, and *u* = GAB_1.0 and *v* = MDF_5 × 5. Several other results are also considered in Table 6, where the results of only TF are considered.

Table 5. Operator series detection on compressed images.

u		GAI	3_1.0			GAB_1.0				GAB_1.0			
v	$MDF_5 \times 5$				$MDF_5 \times 5$				$MDF_5 \times 5$				
	QF1 = 75, QF2 = 85				Q	PF1 = 85,	QF2 = 2	75	QF1 = 90, QF2 = 70				
	T1	T2	T3	TF	T1	T2	T3	TF	T1	T2	T3	TF	
C1	2.94	2.36	1.46	0.96	3.43	2.65	2.34	2.25	5.95	5.01	2.93	3.37	
C2	7.76	5.77	4.70	4.02	9.60	6.94	4.95	5.72	5.41	4.55	8.81	6.31	
C3	5.75	8.18	6.87	5.51	8.10	11.36	11.57	8.65	15.35	14.97	13.53	9.98	
C4	1.54	1.43	1.34	0.96	2.23	3.89	4.01	2.27	3.42	6.02	4.63	3.00	
C5	0.39	0.71	0.29	0.13	1.16	1.59	1.60	0.57	0.19	0.37	0.26	0.07	
MPDE	3.67	3.69	2.93	2.32	4.90	5.28	4.89	3.89	6.07	6.18	6.03	4.55	
STD	3.04	3.17	2.75	2.32	3.73	3.94	3.96	3.26	5.66	5.36	5.22	3.75	

u	v	Compression	C1	C2	C3	C4	C5	MPDE
GAB_1.0	USL_1.5	QF1 = 75, QF2 = 85	0.28	7.99	0.31	3.15	10.38	4.42
GAB_1.0	USL_1.5	QF1 = 85, QF2 = 85	0.91	12.02	0.71	3.49	12.74	5.97
GAB_0.8	$MDF_3 \times 3$	QF1 = 70, QF2 = 90	0.06	1.51	1.15	3.71	1.95	1.68
$MDF_5 \times 5$	USL_1.5	QF1 = 75, QF2 = 85	0.49	16.85	0.25	0.69	14.05	6.47
$MDF_5 \times 5$	USL_1.5	QF1 = 85, QF2 = 75	0.27	20.83	1.04	1.84	16.96	8.19
USM_3.0	USL_1.5	QF1 = 80, QF2 = 90	0.35	0.24	2.14	4.57	2.91	2.04
USM_3.0	USL_1.5	QF1 = 75, QF2 = 85	0.50	0.20	1.93	6.10	6.46	3.04

Table 6. Results of TF for operator series detection on compressed images.

In both the training and testing phases of the aforementioned experimental analysis, the same parameter settings are used for the operators. The operators might be the same but have different parameters in the real world. Experiments are conducted to evaluate the proposed method's robustness to the different operator specification requirements. Gaussian blurring standard deviations 0.7, 0.8, 0.9, and 1.0 are considered for training. The training set consists of sixty thousand images, of which fifteen thousand are processed with the Gaussian blurring parameters on sixty thousand images. The forty thousand photographs are utilized for testing with 300 Gaussian blurring parameters, with a parameter variety of 0.701 to 0.900, for a total of forty thousand images. In order to detect difficulty with the five-class classification of the two-operator series, 300,000 pictures are utilized for training and 200,000 for testing. In different specifications, the proposed CNN model likewise performs quite well. For operators u = GAU and v = UP in Table 7, two scenarios of uncompressed and compressed images with QF1 = 80 and QF2 = 90 are shown.

			U	ncompress	ed	QF1 = 80, QF2 = 90				
Class	T1	T2	T3	TF	T1	T2	T3	TF		
C1	0.51	0.55	0.32	0.16	0.58	0.31	0.62	0.09		
C2	4.14	7.55	2.33	1.27	5.35	5.80	6.23	2.43		
C3	0.60	0.54	4.16	1.63	2.25	8.35	3.05	1.84		
C4	2.26	1.38	0.23	0.34	5.52	3.26	4.49	3.73		
C5	5.68	3.45	0.94	1.08	5.38	6.98	5.16	4.88		
MPDE	2.64	2.69	1.60	0.90	3.81	4.94	3.91	2.59		
STD	2.25	2.96	1.66	0.62	2.27	3.19	2.17	1.83		

Table 7. Operator series detection for different training and testing specifications.

The tests focused on detecting two-operator series, excluding single-operator detection, as shown in Table 8. The images underwent various operations, including Gaussian blurring, median filtering, unsharp masking, and upscaling, and unaltered images were categorized separately. The Mean Percentage Detection Error (MPDE) for uncompressed images of TF was found to be 1.35%, whereas for JPEG images with a QF of 85 of TF, the MPDE was 3.19%. These results indicate that the proposed scheme is effective in detecting two-operator series and well-suited for accurately identifying single-operator processed images.

Table 8. Detection error of single operator.

			Ur	ncompress	sed	QF1 = 80, QF2 = 90			
Class	T1	T2	T3	TF	T1	T2	T3	TF	
UNF	3.70	2.41	6.66	2.23	9.11	14.45	12.44	9.02	
GAB_0.7	0.23	0.17	0.20	0.13	6.17	5.57	11.17	5.26	
$MDF_5 \times 5$	0.11	0.08	0.05	0.06	0.67	0.43	0.54	0.56	
USM_2.0	4.15	5.37	3.39	2.78	11.17	9.69	7.77	7.03	
USL_1.2	1.96	0.03	0.07	0.03	9.46	13.08	11.72	6.82	
MPDE	2.03	1.61	2.07	1.05	7.32	8.64	8.73	5.74	
STD	1.89	2.33	2.93	1.35	4.12	5.73	4.92	3.19	

The proposed CNN is capable of classifying two operators and their series for uncompressed and JPEG small-size images. The in-depth investigation is covered. Now, the outcomes of the proposed method are contrasted with a few different cutting-edge methods. Unlike other conventional models, the CNN model [12] introduces a confined convolutional layer. Different size filters, including 7×7 , 5×5 , and 3×3 , are employed in the convolutional layer. According to our experimental results, small-dimension kernels are better suited. Bayar and Stamm [24] introduced a constrained convolution layer to picture residuals for better outcomes. After improvement, the results are better, but there is still a performance gap because there are fewer convolutional layers and larger filters. In the Bayar and Stamm methods [12,24], method [24] has better performance, which is the reason why method [24] is used for comparison purposes. CNN model with two streams was proposed by Liao et al. [25]. The two-stream model's findings were excellent. Another significant development in the research is the notion of operator series detection. The two-stream model is capable of identifying operators with unidentified requirements. Due to the numerous layers and specialized prior processing that must be used to identify, the computational cost is high. Cho et al. [26] improved the performance using the bottleneck approach. The scheme by Cho et al. did not need customized preprocessing. Due to the transposed convolution and bottleneck method, our proposed scheme is more trustworthy. The two benefits of the bottleneck technique are reducing the learning parameters and allowing for an increase in the network depth. Also, information fusion is performed to reduce the detection errors. The outcomes of several scenarios for uncompressed and compressed images are displayed in Figures 9 and 10, respectively.



Figure 9. Comparative analysis with [24-26] schemes for uncompressed images.



Figure 10. Comparative analysis with [24-26] schemes of compressed images.

In a comparative analysis, it has been found that the scheme proposed by Liao et al. [25] exhibits inferior performance compared to our proposed CNN architecture. Liao's CNN model includes multiple pooling layers, which, unfortunately, results in the loss of critical statistical data during the downsampling process. Additionally, their approach's usage of large kernel sizes has negatively impacted the overall performance. In contrast, the scheme introduced by Cho et al. [26] outperforms Liao et al.'s approach. In Cho et al.'s scheme, the use of pooling layers is omitted, which prevents the loss of crucial statistical information. This absence of pooling layers contributes to better data preservation, ultimately improving the results.

Our proposed CNN architecture considers these observations, aiming to address the limitations of previous approaches. Instead of employing pooling layers, we utilize a transposed convolution, which allows us to retain as many hereditary fingerprints as possible during the upsampling process. This feature is vital for maintaining essential information and ensuring the accuracy of the classification. Furthermore, the information fusion techniques are incorporated into our proposed CNN model to enhance detection performance further. Combining information from multiple layers improves the accuracy and reliability of operator sequence detection.

As a result of these enhancements and adaptations in our proposed CNN architecture, the method's performance has shown significant improvement when compared to both Liao et al.'s and Cho et al.'s schemes. Using transposed convolution, coupled with information fusion, is crucial in achieving more precise and reliable operator sequence detection. These advancements make our proposed CNN model a promising solution for addressing the challenges of image processing history analysis and hold potential for various real-world applications in the field.

In all other respects, the proposed CNN also performs well. The proposed technique has a lower average classification error, exclusive of particular preprocessing.

5. Conclusions

The widespread type of information representation is digital images. Recent technological advancements have made it simple for ignorant users to produce deceiving photos. However, several processes were carried out to create a deceptive image that appeared authentic. The misleading image's detection has been made easier by discovering the manipulation operations. In this paper, a new scheme has been proposed to maintain an image's authenticity. So far, most approaches have been proposed for single-operator detection. The order of operators and two operators have been identified using a few schemes. The image's consecutive two operators and matching order could be precisely detected using the proposed information fusion-based deep learning model. The bottleneck strategy was used to add more layers while maintaining the number of training parameters and reducing the detection error.

In contrast to earlier networks, the information loss and overfitting issues have been addressed using a global average pooling layer in the proposed scheme. Information fusion and transposed convolution have been performed to decrease classification errors. The proposed model successfully handled various conditions in the experimental investigation, including low-resolution images and JPEG compression.

The proposed method currently focuses on handling two operator sequences. However, it is essential to acknowledge that real-world scenarios may involve more than two sequences. Therefore, it is crucial to thoroughly verify the effectiveness and performance of the proposed method under such circumstances where there are multiple operator sequences. Additionally, while the proposed method shows promising results in detecting compressed images, there is still room for improvement in handling highly compressed images. Highly compressed photos can pose a challenge due to the significant loss of data and details, making them more difficult to detect accurately. Thus, further research and development are needed to enhance the method's capabilities in detecting and effectively handling such highly compressed images. **Author Contributions:** Conceptualization, S.A. and K.-H.J.; software, S.A.; validation, D.-J.C.; formal analysis, S.A.; investigation, D.-J.C.; resources, S.A.; data curation, K.-H.J.; writing—original draft preparation, S.A.; writing—review and editing, D.-J.C. and K.-H.J.; visualization, S.A.; supervision, K.-H.J.; project administration, K.-H.J.; funding acquisition, K.-H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (2019H1D3A1A01101687) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R111A3049788).

Data Availability Statement: The datasets used in this paper are publicly available and their links are provided in the reference section.

Acknowledgments: We thank the anonymous reviewers for their valuable suggestions that improved the quality of this article.

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

References

- 1. Xiao, Y.; Tian, H.; Cao, G.; Yang, D.; Li, H. Effective PRNU Extraction via Densely Connected Hierarchical Network. *Multimed. Tools Appl.* **2022**, *81*, 20443–20463. [CrossRef]
- Manisha; Li, C.-T.; Lin, X.; Kotegar, K.A. Beyond PRNU: Learning Robust Device-Specific Fingerprint for Source Camera Identification. Sensors 2022, 22, 7871. [CrossRef]
- 3. You, C.; Zheng, H.; Guo, Z.; Wang, T.; Wu, X. Tampering Detection and Localization Base on Sample Guidance and Individual Camera Device Convolutional Neural Network Features. *Expert Syst.* **2022**, *40*, e13102. [CrossRef]
- Agarwal, S.; Jung, K.-H. Median Filtering Forensics Based on Optimum Thresholding for Low-Resolution Compressed Images. Multimed. Tools Appl. 2022, 81, 7047–7062. [CrossRef]
- Zhang, J.; Liao, Y.; Zhu, X.; Wang, H.; Ding, J. A Deep Learning Approach in the Discrete Cosine Transform Domain to Median Filtering Forensics. *IEEE Signal Process. Lett.* 2020, 27, 276–280. [CrossRef]
- Wang, D.; Gao, T. An Efficient USM Sharpening Detection Method for Small-Size JPEG Image. J. Inf. Secur. Appl. 2020, 51, 102451. [CrossRef]
- Li, G.; Wang, Y.; Liu, Z.; Zhang, X.; Zeng, D. RGB-T Semantic Segmentation with Location, Activation, and Sharpening. *IEEE Trans. Circuits Syst. Video Technol.* 2022, 33, 1223–1235. [CrossRef]
- Li, Y.; Xia, M.; Liu, X.; Yang, G. Identification of Various Image Retargeting Techniques Using Hybrid Features. J. Inf. Secur. Appl. 2020, 51, 102459. [CrossRef]
- 9. Peng, A.; Wu, Y.; Kang, X. Revealing Traces of Image Resampling and Resampling Antiforensics. *Adv. Multimed.* 2017, 2017, 7130491. [CrossRef]
- Qiao, T.; Zhu, A.; Retraint, F. Exposing Image Resampling Forgery by Using Linear Parametric Model. *Multimed. Tools Appl.* 2018, 77, 1501–1523. [CrossRef]
- Bayar, B.; Stamm, M.C. A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, New York, NY, USA, 20 June 2016; ACM: New York, NY, USA, 2016; pp. 5–10.
- Bayar, B.; Stamm, M.C. Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection. *IEEE Trans. Inf. Forensics Secur.* 2018, 13, 2691–2706. [CrossRef]
- Boroumand, M.; Fridrich, J. Deep Learning for Detecting Processing History of Images. *Electron. Imaging* 2018, 30, 213-1–213-219. [CrossRef]
- 14. Li, H.; Luo, W.; Qiu, X.; Huang, J. Identification of Various Image Operations Using Residual-Based Features. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 31–45. [CrossRef]
- 15. Singhal, D.; Gupta, A.; Tripathi, A.; Kothari, R. CNN-Based Multiple Manipulation Detector Using Frequency Domain Features of Image Residuals. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–26. [CrossRef]
- 16. Xue, H.; Liu, H.; Li, J.; Li, H.; Luo, J. Sed-Net: Detecting Multi-Type Edits Of Images. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
- Barni, M.; Nowroozi, E.; Tondi, B.; Zhang, B. Effectiveness of Random Deep Feature Selection for Securing Image Manipulation Detectors Against Adversarial Examples. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2977–2981.
- Liu, X.; Liu, Y.; Chen, J.; Liu, X. PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization. *IEEE Trans. Circuits Syst. Video Technol.* 2022, 32, 7505–7517. [CrossRef]
- Wang, J.; Wu, Z.; Chen, J.; Han, X.; Shrivastava, A.; Lim, S.-N.; Jiang, Y.-G. ObjectFormer for Image Manipulation Detection and Localization. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 2354–2363.

- Agarwal, S.; Jung, K.-H. Photo Forgery Detection Using RGB Color Model Permutations. *Imaging Sci. J.* 2022, 70, 87–101. [CrossRef]
- 21. Stamm, M.C.; Chu, X.; Liu, K.J.R. Forensically Determining the Order of Signal Processing Operations. In Proceedings of the 2013 IEEE International Workshop on Information Forensics and Security, WIFS 2013, Guangzhou, China, 18–21 November 2013.
- Chu, X.; Chen, Y.; Liu, K.J.R. Detectability of the Order of Operations: An Information Theoretic Approach. *IEEE Trans. Inf. Forensics Secur.* 2016, 11, 823–836. [CrossRef]
- Comesaña, P. Detection and Information Theoretic Measures for Quantifying the Distinguishability between Multimedia Operator Chains. In Proceedings of the WIFS 2012—2012 IEEE International Workshop on Information Forensics and Security, Costa Adeje, Spain, 2–5 December 2012.
- 24. Bayar, B.; Stamm, M.C. Towards Order of Processing Operations Detection in JPEG-Compressed Images with Convolutional Neural Networks. *Electron. Imaging* **2018**, 2018, 211-1–211-219. [CrossRef]
- 25. Liao, X.; Li, K.; Zhu, X.; Liu, K.J.R. Robust Detection of Image Operator Chain with Two-Stream Convolutional Neural Network. *IEEE J. Sel. Top. Signal Process.* 2020, 14, 955–968. [CrossRef]
- Cho, S.-H.; Agarwal, S.; Koh, S.-J.; Jung, K.-H. Image Forensics Using Non-Reducing Convolutional Neural Network for Consecutive Dual Operators. *Appl. Sci.* 2022, 12, 7152. [CrossRef]
- Barni, M.; Costanzo, A.; Nowroozi, E.; Tondi, B. Cnn-Based Detection of Generic Contrast Adjustment with JPEG Post-Processing. In Proceedings of the International Conference on Image Processing, ICIP, Athens, Greece, 7–10 October 2018; pp. 3803–3807.
- Bas, P.; Filler, T.; Pevný, T. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In International Workshop on Information Hiding; Springer: Berlin/Heidelberg, Germany, 2011; pp. 59–70.
- Venkatanath, N.; Praneeth, D.; Maruthi Chandrasekhar, B.; Channappayya, S.S.; Medasani, S.S. Blind Image Quality Evaluation Using Perception Based Features. In Proceedings of the 2015 Twenty First National Conference on Communications (NCC), IEEE, Mumbai, India, 27 February–1 March 2015; pp. 1–6.
- Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Trans. Image Process.* 2012, 21, 4695–4708. [CrossRef]
- Zhang, R.; Zhu, F.; Liu, J.; Liu, G. Depth-Wise Separable Convolutions and Multi-Level Pooling for an Efficient Spatial CNN-Based Steganalysis. *IEEE Trans. Inf. Forensics Secur.* 2020, 15, 1138–1150. [CrossRef]
- Xu, G.; Wu, H.-Z.; Shi, Y.-Q. Structural Design of Convolutional Neural Networks for Steganalysis. *IEEE Signal Process. Lett.* 2016, 23, 708–712. [CrossRef]
- Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 32nd Int. Conf. Mach. Learn. ICML 2015, 1, 448–456.
- Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the ICML 2010—Proceedings, 27th International Conference on Machine Learning, Madison, WI, USA, 21–24 June 2010.
- Xu, G. Deep Convolutional Neural Network to Detect J-UNIWARD. In Proceedings of the IH and MMSec 2017—2017 ACM Workshop on Information Hiding and Multimedia Security, New York, NY, USA, 20–22 June 2017.
- Yedroudj, M.; Comby, F.; Chaumont, M. Yedrouj-Net: An Efficient CNN for Spatial Steganalysis. 2018 IEEE Int. Conf. Acoust. Speech Signal Process. 2018, 2018, 2092–2096. [CrossRef]
- Lin, M.; Chen, Q.; Yan, S. Network in Network. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014—Conference Track Proceedings, Banff, AB, Canada, 14–16 April 2014.
- Xavier Glorot, Y.B. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 31 March 2010; pp. 249–256.
- Schaefer, G.; Stich, M. UCID: An Uncompressed Color Image Database. In Proceedings of the Storage and Retrieval Methods and Applications for Multimedia 2004, SPIE, San Jose, CA, USA, 22 December 2003; Volume 5307, pp. 472–480.
- Abdulrahman, H.; Chaumont, M.; Montesinos, P.; Magnier, B. Color Image Steganalysis Based On Steerable Gaussian Filters Bank. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security—IH&MMSec '16, New York, NY, USA, 20–22 June 2016; ACM Press: New York, NY, USA, 2016; pp. 109–114.
- 41. Liu, Q.; Chen, Z. Improved Approaches with Calibrated Neighboring Joint Density to Steganalysis and Seam-Carved Forgery Detection in JPEG Images. *ACM Trans. Intell. Syst. Technol.* **2015**, *5*, 1–30. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.