applied sciences

MDPI

Article

A New Method to Detect Splicing Image Forgery Using Convolutional Neural Network

Khalid M. Hosny [1,*], Akram M. Mortda [2], Nabil A. Lashin [1] and Mostafa M. Fouda [3,*]

[1] Department of Information Technology, Zagazig University, Zagazig 44519, Egypt
[2] Department of Information Technology, Faculty of Information Technology and Computer Science, Sinai University, Arish 16020, Egypt
[3] Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID 83209, USA
* Correspondence: k_hosny@yahoo.com (K.M.H.); mfouda@ieee.org (M.M.F.)

Abstract: Recently, digital images have been considered the primary key for many applications, such as forensics, medical diagnosis, and social networks. Image forgery detection is considered one of the most complex digital image applications. More profoundly, image splicing was investigated as one of the common types of image forgery. As a result, we proposed a convolutional neural network (CNN) model for detecting splicing forged images in real-time and with high accuracy, with a small number of parameters as compared with the recently published approaches. The presented model is a lightweight model with only four convolutional layers and four max-pooling layers, which is suitable for most environments that have limitations in their resources. A detailed comparison was conducted between the proposed model and the other investigated models. The sensitivity and specificity of the proposed model over CASIA 1.0, CASIA 2.0, and CUISDE datasets are determined. The proposed model achieved an accuracy of 99.1% in detecting forgery on the CASIA 1.0 dataset, 99.3% in detecting forgery on the CASIA 2.0 dataset, and 100% in detecting forgery on the CUISDE dataset. The proposed model achieved high accuracy, with a small number of parameters. Therefore, specialists can use the proposed approach as an automated tool for real-time forged image detection.

Keywords: deep learning; image processing; lightweight model

Citation: Hosny, K.M.; Mortda, A.M.; Lashin, N.A.; Fouda, M.M. A New Method to Detect Splicing Image Forgery Using Convolutional Neural Network. *Appl. Sci.* **2023**, *13*, 1272. https://doi.org/10.3390/app13031272

Academic Editor: Andrea Prati

Received: 15 December 2022
Revised: 11 January 2023
Accepted: 15 January 2023
Published: 18 January 2023

1. Introduction

With the advancement of technology, digital images have become widely used in many fields, such as social networks [1], the military [2], computer-aided medical diagnosis systems [3], and evidence in court and forensics [4], and it has become very easy to obtain them. The focus of human interest in the current era is photography, which led to the huge growth of digital images. The high progress of technology was the reason behind the dramatic increase in the tools that have been used to manipulate digital images. Therefore, it is necessary to find more efficient approaches to detect forgery images.

There are two approaches to image forgery: active and passive. Active approaches include digital signatures in images and watermarks; passive approaches include copying, splicing, image morphing, image retouching, and image enhancement [5,6].

Image splicing is the most significant type of image forgery. Many methods for detecting image splicing were proposed in the image field. In general, we can classify these methods into two classes. First, extract the features using traditional methods such as Markov features in discrete cosine transform (DCT) and discrete wavelet transform (DWT), and extract features by using support vector machines (SVM) and the orthogonal moments. Second is the deep learning image splicing forgery detection (ISFD) technique, where different deep learning methods are configured. In Figure 1, A and B are the original images, and C is the splicing image.

Figure 1. Represents the splicing of original images for obtaining a spliced image forgery, where (**A**,**B**) are the original images, and (**C**) is the spliced image forgery.

1.1. Traditional Splicing Forgery Detection Approach

An efficient approach for image splicing detection based on Markov features in the DCT and DWT domain was suggested by [7]. Experimental results substantiate the high performance of their approach as compared with the others. An approach for image splicing based on inter-color channel information has been introduced [8]. This approach aims to detect the most appropriate chroma-like channel, but this approach is computationally complex. An improved Markov state selection approach, which matches coefficients to Markov states based on a well-performed functional model, is proposed by Bo Su et al. [9]. This work focuses on enhancing the Markov state selection method [10,11], and the obtained results reveal high performance compared to the previous version. However, the number of images used in the database is not sufficient to measure the best performance of the proposed approach. An enhanced version of the Run Length Run number algorithm for ISFD is introduced by Zahra Moghaddasi et al. [12], and the enhanced version used the principal component analysis (PCA) and kernel PCA, which is a blind technique for ISFD based on the Markov features for edge images in spatial domain and DCT coefficients. The efficiency of the previous merge has been proved by using PCA and SVM, suggested by El-Alfy et al. [13].

An efficient technique for the spliced blurred image can localize the spliced region proposed by Khosro Bahrami et al. [11]. This approach can be applied only to blurred images. A Markov-based approach in the DCT and contourlet transform domains introduced by Qingbo Zhang and Wei Lu [14]. The superiority of the proposed approach is that it can be extended in terms of gray and color image splicing detection. Chi-Man Pun et al. [15] exposed a novel approach for ISFD, using noise discrepancies in multiple scales as an indicator for ISFD. The proposed approach reveals high superiority when compared with existing state-of-the-art approaches. An efficient approach for color ISFD was suggested by Ce Li et al. [16]. The authors used Markov features in quaternion discrete cosine transform (QDCT), then exploited SVM to make a classification for the Markov feature vector. The experimental results reveal high superiority thanks to their approach with more than 92.38% accuracy compared with other recently published methods, but accuracy should be improved on that. An efficient algorithm based on the PCA algorithm and the K-means method has been introduced by Hui Zeng et al. [17]. The experimental results specified good results for ISFD when compared to the original and spliced regions. An approach for ISFD based on local binary pattern (LBP) and DCT for feature extraction; hence, SVM, has been used for detection and is proposed by Alahmadi et al. [10]. A novel method for ISFD based on a noise level function (NLF), the values of NLF reflect the relationship between noise variance and sharpness of image blocks, is introduced by Nan Zhu and Zhao Li [18]. The experimental results reveal the high superiority of the proposed method, but that approach cannot detect more areas of forgery.

An efficient method for ISFD based on several algorithms: roughness measure algorithm, PCA algorithm, and SVM algorithm, is suggested by Zahra et al. [19]. These algorithms together enhance the overall process of ISFD. An efficient algorithm based on the optimal threshold local ternary pattern has been introduced [20], and the proposed technique achieved an accuracy of up to 98.25%. Kunj et al. [21] proposed an approach for detecting and localizing ISF based on noise level estimation, with high accuracy revealed when experiments were performed on the CUISDE dataset. Local binary pattern (LBP) has been employed for ISFD [22], the LBP is used to compute the image texture features. Hence, machine learning algorithms have been used for classification.

Quaternion representation QR represents an efficient approach for representing color images. Zhang et al. [23] proposed an approach for ISFD depending on error level analysis (ELA) and local binary pattern (LBP). Chen et al. [24] introduced an improved quaternion representation (QR) approach based on pseudo-Zernike moments to resolve the redundancy problem. The proposed approach has been used for color ISFD.

1.2. Deep Learning-Based Splicing Forgery Detection Approach

An effective solution for ISFD based on a fully convolutional network (FCN) is presented by Salloum et al. [25]. The authors first introduced a single FCN (SFCN); after that, they used multi-task FSN (MFSN), and the experimental results have shown superiority in favor of SFCN and MFCN when compared with exciting splicing localization algorithms. A novel ISFD method has been introduced by Bin Xiao et al. [26]. The proposed approach depends on a coarse-to-refined convolutional neural network (C2RNet) and diluted adaptive clustering. Experimental results reveal the high superiority of the proposed approach over other existing approaches. However, the proposed detection method only focuses on one manipulated area in the image due to the limitation of the post-processing approach and cannot detect the distortion otherwise, which is an efficient blind ISFD technique. Additionally, they employed a deep learning architecture called ResNet-Conv suggested by Belal Ahmed et al. [27]. The suggested model has been trained and evaluated using a computer-generated image splicing dataset and found to be more efficient than other models. S. Nath and R. Naskar have suggested a blind ISFD technique [28]. The proposed approach used a deep convolutional residual network and a fully connected classifier network. Good results were obtained when the approach was tested using the CASIA v2.0 database. An efficient ISFD based on dual-channel U-Net, that is, DCU-Net is suggested by Hongwei et al. [29]. Experimental results reveal the robustness of the proposed approach. Multiple ISFD techniques were proposed by Kadam et al. [30]. The authors used Mask R-CNN with MobileNet VI as a backbone architecture. The proposed method was tested over ultra-modern datasets such as CASIA, Wild Web, MISD, and Columbia Gray. The results specified good superiority. However, his proposed model is not tested on a larger number of attacks, and there is no comparison of evaluation results with and without attacks. A lightweight architecture based on CNN for copy-move forgery detection is introduced by Hosny et al. [31]. The presented approach reveals superiority in terms of time and accuracy compared with other recently published methods. However, it does not have a high-efficiency rate when it comes to splicing image fraud.

The main contribution of this work is as follows:

- The proposed model achieved high accuracy with a small number of parameters as compared with the recently published approaches, which can be considered as power key for the proposed architecture. Moreover, the proposed model is suitable for environments that have limitations in memory space and CPUs.
- The presented CNN model has four convolutional layers, four max-pooling layers, one global average pooling layer, one fully connected layer, and 97,698 hyper-parameters shown in Table 1, so it is a lightweight CNN model.
- Three standard datasets were used that allowed us to provide accurate experiments, and these datasets are CASIA 1.0 [32], CASIA 2.0 [32], and CUISDE [33].
- Experiments were conducted on the dataset, and an analytical comparison was made between the proposed model's results and previously presented models (Alahmadi et al. [10], Kanwal et al. [20], Zhang et al. [22], Ding et al. [29], Itier et al. [34], Kadam et al. [30], Abd El-Latif et al. [35], Nath et al. [28], and Niyishaka et al. [22]). This comparison showed that the proposed model is efficient and accurate against the other investigated models.

Table 1. The proposed model activation shape, activation size, and hyperparameters.

Layers	Activation Shape	Activation Size	Number of Parameters
Input layer	(224,224,3)	163,968	0
Conv1	(222,222,16)	788,544	448
Max pool1	(111,111,16)	197,136	0
Conv2	(109,109,32)	380,192	4640
Max pool2	(54,54,32)	93,312	0
Conv3	(52,52,64)	173,056	18,496
Max pool3	(26,26,64)	43,264	0
Conv4	(24,24,128)	73,728	73,856
Max pool4	(12,12,128)	18,432	0
Global Average Pooling 2D	18,432	18,432	0
Fully Connected	18,432	18,432	0
Output layer	2	2	258
Total number of parameters			97,698

This study encompasses other parts as follows: the preliminaries of CNN have been discussed through Section 2. In Section 3, we discussed the proposed approach in detail. The experimental results are explained and discussed in Section 4. Finally, through Section 5, we exposed the conclusion.

2. Preliminaries

Understanding of a CNN

CNN stands for a convolution neural network. It is a class of deep learning consisting of multilayers. It has gained much popularity in the literature due to its ability to handle enormous amounts of data. Most of the advantages of convolutional neural networks are related to reducing the number of parameters in ANN, which encourages researchers and developers to use larger models to complete tasks previously impossible with standard ANNs Albawi et al. [36].

The CNN consists of a group of elements (layers). The basic elements of CNN are the convolutional layer Yamashita et al. [37], the pooling layer, and the fully connected layer. It is intended to automatically and adaptively learn the spatial hierarchy of features using the backpropagation technique shown in Figure 2.

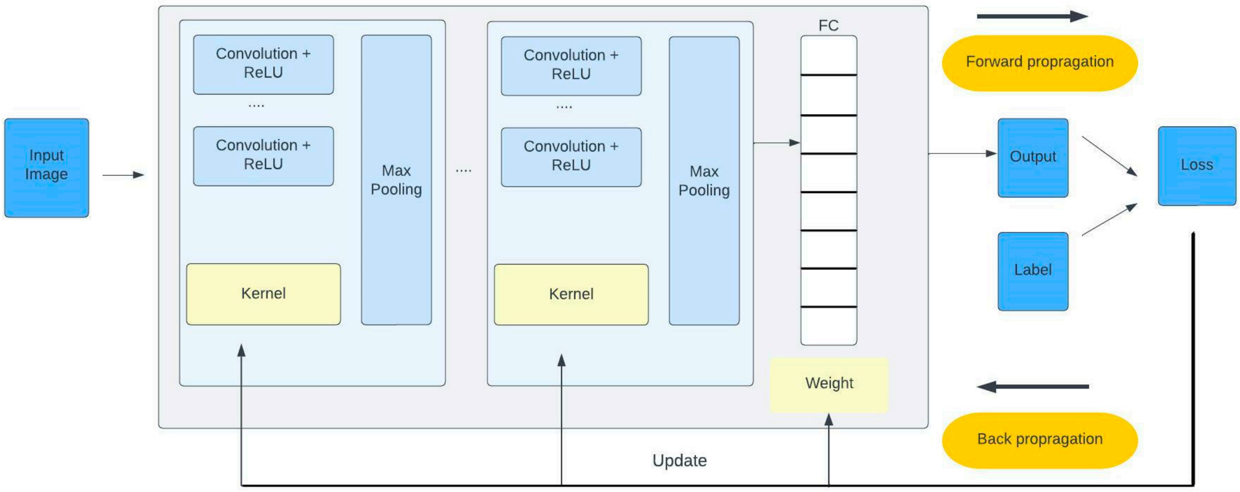

Figure 2. Shows the overall architectural of the convolutional neural network, which includes an input layer, multiple alternating convolution and max-pooling layers, one follow connected layer, and one classification layer.

3. Proposed Approach

This study introduces an efficient and accurate model. The proposed CNN model is shown in Figure 3. It deals with the image as a whole. The traditional approaches deal with the image as a block. Our approach consists of three stages.

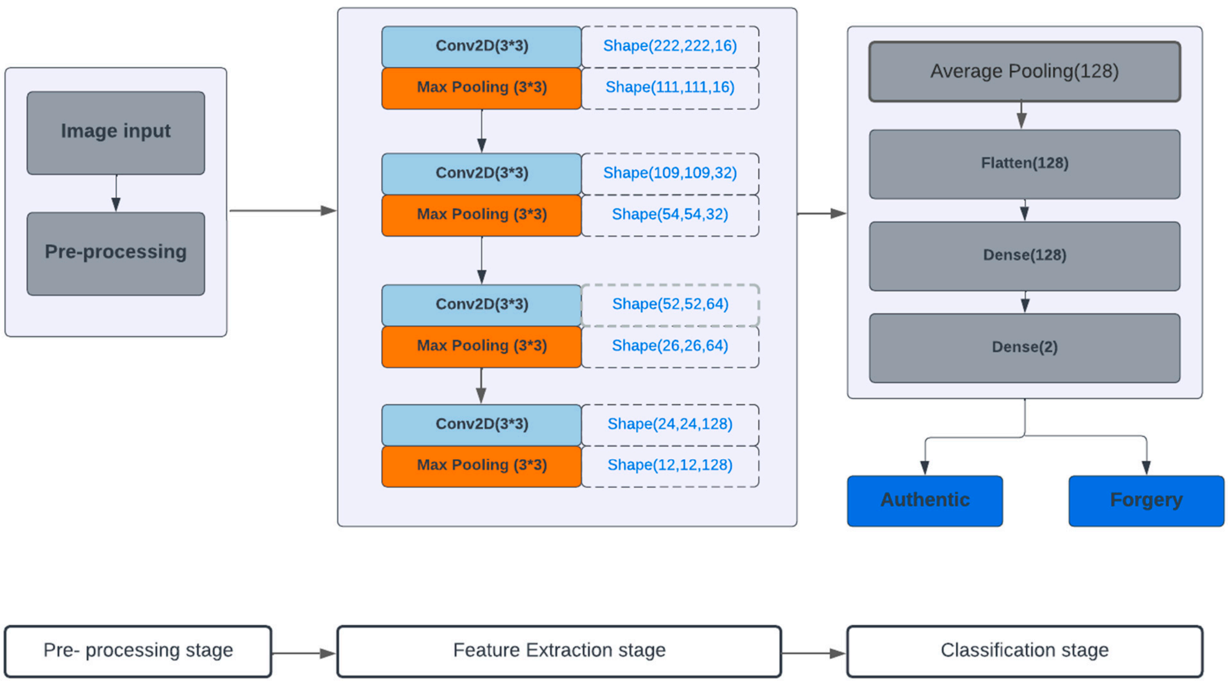

Figure 3. The structure of the proposed algorithm CNN layers.

- The first stage is preprocessing. In this stage, the image is resized to a suitable size to be inserted into the next stage without cutting any part of the entered image.
- The second stage is feature extraction. At this stage, there are four convolutional layers. Each convolutional layer is followed by the following: a max-pooling layer, one global average pooling layer, and one fully connected layer. The first convolutional layer has 16 feature maps, a filter size of (3,3), an input shape of (224,224), and an activation function (RELU). The first max-pooling layer has a pool size of (2,2). The second convolutional layer has 32 feature maps, a filter size of (3,3), a shape of (111,111), and

an activation function (RELU). The second max-pooling layer has a pool size of (2,2). The third convolutional layer has 64 feature maps, a filter size of (3,3), an input shape of (54,54), and an activation function (RELU). The third max-pooling layer has a pool size of (2,2). The fourth convolutional layer has 128 feature maps, a filter size of (3,3), an input shape of (26,26), and an activation function (RELU). The fourth max-pooling has a pool size of (2,2). These hyperparameters were tabulated in Table 1. The final stage is a dense layer called the classification stage, and it classifies data into two categories: authentic or forgery. The main role of the convolutional layer is to extract features. Each convolutional layer has its own feature maps based on its specified filter. In the first convolutional layer, feature map sizes were reduced, which is important for providing next-layer feature maps. This process is called max-pooling [36]. This map works as an input to the next convolutional layer.

- The third stage is the classification stage: the output of the last block of the convolutional part represents the input of the global average pooling layer of the classification part. The final pooled feature maps of the global average pooling layer are formulated as vectors and fed to the fully connected layer. Finally, we can detect whether the input image is a forgery or authentic.

4. Experimental Results

Through this section, we introduced, in detail, many experiments to test the efficiency of the proposed approach. The experiments have been implemented on a google collab server machine with the following specifications: GPU and RAM: 2.5 GB/12 GB in python 3, and using Keras with TensorFlow backend.

4.1. Datasets

The experiments have been completed over three benchmark datasets, namely: CASIA v1.0 [32], CASIA v2.0 [32], and CUISDE [33]. All datasets contain original and forgery color images shown in Table 2.

CUISDE [33] dataset consists of 363 images, 183 original images, and 180 images forgery. Its resolution is 568 $\times$ 757 to 768 $\times$ 1152. Its extensions are BMP or Tiff format.

CASIA 1.0 [32] dataset consists of 913 images, 451 original images, and 462 images forgery, and its resolution is 384 $\times$ 256 or 256 $\times$ 384. The images are in JPG format.

CASIA 2.0 [32] dataset consists of 12,613 images, 7491 original images, and 5122 images forgery. Its resolution is 900 $\times$ 600. Its extensions are BMP, TIFF, or JPG format. Figure 4 shows a sample of these datasets.

4.2. Evaluation Metrics

The following metrics are used to test the efficiency of the proposed model [35,38]:

$$accuracy = \frac{(\mathbf{T_N}+\mathbf{T_P})}{(\mathbf{T_P}+\mathbf{F_P}+\ \mathbf{T_N}+\ \mathbf{F_N})} * \mathbf{100} \tag{1}$$

$$precision = \frac{\mathbf{T_P}}{\mathbf{T_P}+\mathbf{F_P}} * \mathbf{100} \tag{2}$$

$$Recall = \frac{\mathbf{T_P}}{\mathbf{T_P}+\mathbf{F_N}} * \mathbf{100} \tag{3}$$

$$F\mathbf{1} - score = \frac{\mathbf{2} * (precision * precall)}{(precision + Recall)} \tag{4}$$

$$sensitivity = \frac{number\ of\ true\ positives(\mathbf{T_P})}{(True\ positives(\mathbf{T_P}) + False\ negatives(\mathbf{F_N}))} \tag{5}$$

$$specificity = \frac{number\ of\ true\ negatives(\mathbf{T_N})}{(True\ Negatives(\mathbf{T_N}) + False\ posotives(\mathbf{F_P}))} \tag{6}$$

Table 2. The details of the CASIA 1.0, CASIA 2.0, and CUISDE datasets.

<table>
<tr><th>Dataset</th><th colspan="4">Composition</th><th>Size of Image</th><th colspan="4">No. of Training Images</th><th colspan="4">No. of Validation Images</th><th colspan="4">No. of Testing Images</th><th>The Input Shape</th></tr>
<tr><td rowspan="2">CASIA 1.0 [32]</td><td colspan="4">913 images</td><td rowspan="2">384 × 256 pixels and 256 × 384 pixels</td><td colspan="4">457 images</td><td colspan="4">229 images</td><td colspan="4">227 images</td><td rowspan="2">224 × 244 pixels</td></tr>
<tr><td>tampered</td><td>462</td><td>original</td><td>451</td><td>tampered</td><td>231</td><td>original</td><td>226</td><td>tampered</td><td>116</td><td>original</td><td>113</td><td>tampered</td><td>115</td><td>original</td><td>112</td></tr>
<tr><td rowspan="2">CASIA 2.0 [32]</td><td colspan="4">12,613 images</td><td rowspan="2">900 × 600 pixels</td><td colspan="4">6308 images</td><td colspan="4">3154 images</td><td colspan="4">3152 images</td><td rowspan="2">224 × 244 pixels</td></tr>
<tr><td>tampered</td><td>5122</td><td>original</td><td>7491</td><td>tampered</td><td>2562</td><td>original</td><td>3746</td><td>tampered</td><td>1281</td><td>original</td><td>1873</td><td>tampered</td><td>1280</td><td>original</td><td>1872</td></tr>
<tr><td rowspan="2">CUISDE [33]</td><td colspan="4">363 images</td><td rowspan="2">757 × 568 to 1152 × 768 pixels</td><td colspan="4">183 images</td><td colspan="4">90 images</td><td colspan="4">90 images</td><td rowspan="2">224 × 244 pixels</td></tr>
<tr><td>tampered</td><td>180</td><td>original</td><td>183</td><td>tampered</td><td>90</td><td>original</td><td>93</td><td>tampered</td><td>45</td><td>original</td><td>45</td><td>tampered</td><td>45</td><td>original</td><td>45</td></tr>
</table>

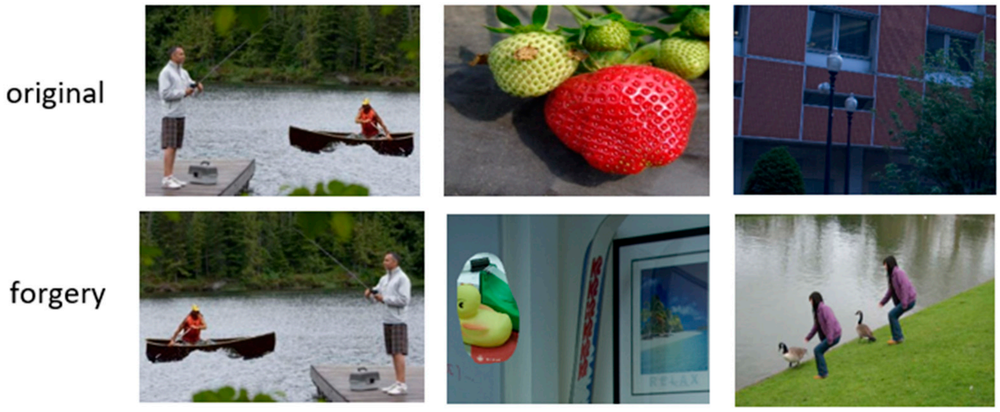

Figure 4. Samples of datasets CASIA v1.0, CASIA v2.0, and CUISDE.

4.3. Experimental Results

Our study has been tested over CASIA 1.0 [32], CASIA 2.0 [32], and CUISDE [33] datasets. The results obtained were evaluated against recently published methods (A. Alahmadi et al. [10], N. Kanwal et al. [20], Y. Zhang et al. [23], H. Ding et al. [29], V. Itier et al. [34], K. Kadam et al. [30], E. Abd El-Latif et al. [35], S. Nath et al. [28] and P. Niyishaka et al. [22]). Results of confusion matrices are specified in Table 3. The sensitivity and specificity of the proposed model over CASIA 1.0, CASIA 2.0, and CUISDE datasets are shown in Table 4. The feature map for a forgery image from a CASIA 1.0 dataset is shown in Figure 5.

4.4. The Results and Comparison over the CASIA 1.0, CASIA 2.0, and CUISDE Datasets

Through the present study, we computed the F1-score for the proposed model and compared it with the other recently published approaches over the CASIA 1.0 [32], CASIA v2.0 [32], and CUISDE [33] datasets. The obtained results are specified in Table 5.

Over CASIA 1.0, the proposed approach reveals high superiority in terms of the F1-score, which has achieved an F1-score value of 97.34% for A. Alahmadi et al. [10], 97.03% for E. Abd El-Latif et al. [35], 98.3% for N. Kanwal et al. [20], and 61.0% for K. Kadam et al. [30], and the proposed model achieves an F1-score of 99.14%.

Table 3. Confusion matrices of the proposed model over CASIA 1.0, CASIA 2.0, and CUISDE dataset.

Dataset	Classes	+	−	Total
CASIA 1.0	+	115	0	115
	−	2	110	112
	Total	117	110	227
CASIA 2.0	+	1850	22	1872
	−	0	1280	1280
	Total	1850	1302	3152
CUISDE	+	45	0	45
	−	0	45	45
	Total	45	45	90

The positive (+) sign stands for the original classes, while the negative (−) sign stands for the forgery classes. Blue color indices are the number of corrected detected images by the proposed approach.

Table 4. Sensitivity and specificity of the proposed model over CASIA 1.0, CASIA 2.0, and CUISDE dataset.

Dataset	Sensitivity %	Specificity %
CASIA 1.0	98.29	100
CASIA 2.0	100	98.31
CUISDE	100	100

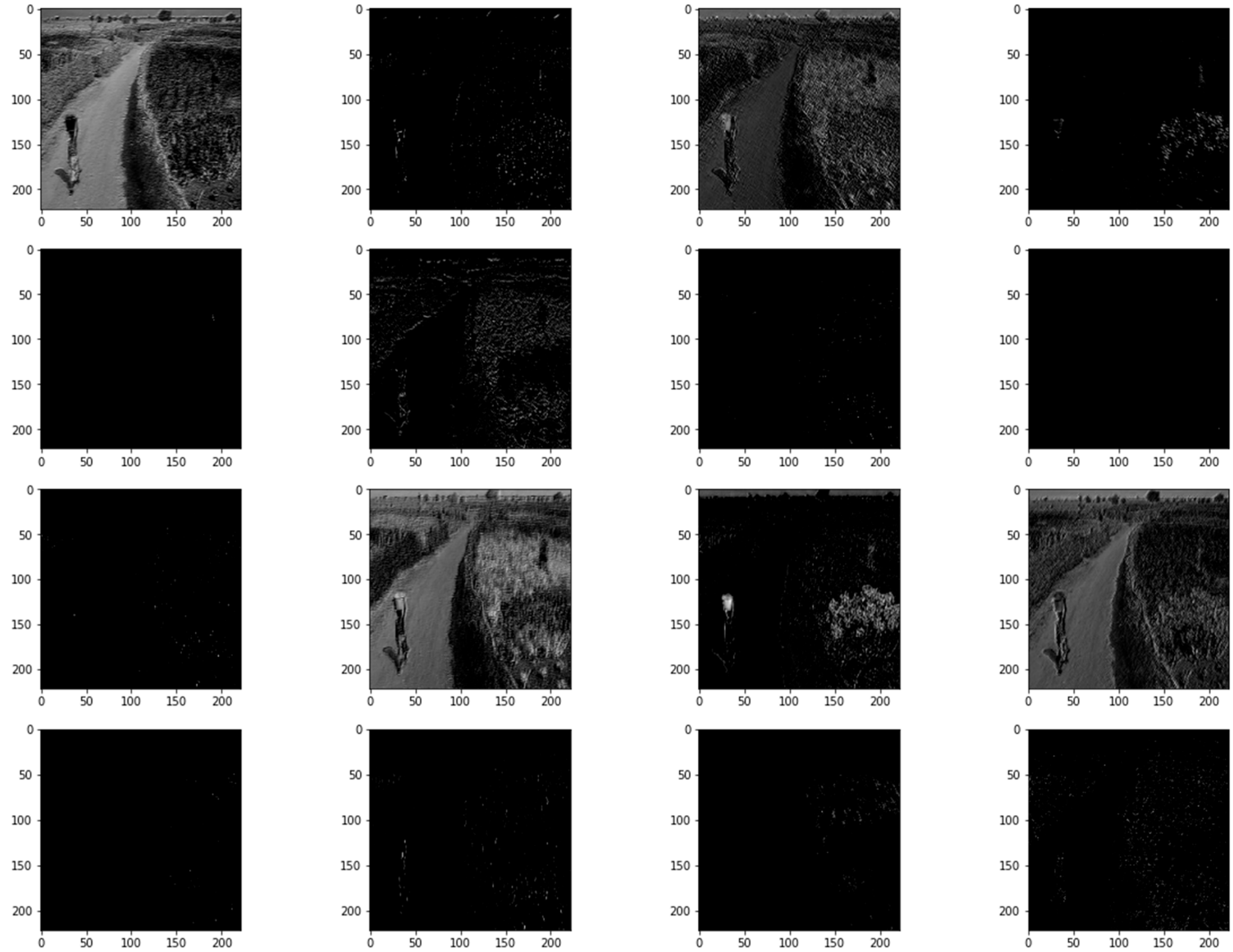

Figure 5. Feature map for a forgery image from a CASIA 1.0 dataset.

Table 5. A comparison of F1-score, precision, and recall in the case of the proposed model and other recently published approaches over the CASIA 1.0, CASIA 2.0, and CUISDE dataset.

Methods	CASIA 1.0			CASIA 2.0			CUISDE		
	Recall %	Precision %	F1-Score %	Recall %	Precision %	F1-Score %	Recall %	Precision %	F1-Score %
A. Alahmadi et al. [10]	98.2	96.75	97.34	96.84	98.45	97.64	97.07	98.3	97.68
E. Abd El-Latif et al. [35]	98.99	95.14	97.03	99.03	97.14	98.08	-	-	-
N. Kanwal et al. [20]	100	-	98.3	100	-	97.52	-	-	-
K. Kadam et al. [30]	66.0	67.0	61.0	-	-	-	66.0	67	66.0
H. Ding et al. [29]	-	-	-	88.93	89.12	86.67	91.76	99.81	94.98
S. Nath et al. [28]	-	-	-	94.15	96.69	95.4	-	-	-
P. Niyishaka et al. [22]	-	-	-	99	97	98	-	-	-
Y. Zhang et al. [23]	-	-	-	-	-	-	93.99	89.58	91.73
Proposed method	100	98.3	99.14	98.83	100	99.4	100	100	100

Over CASIA 2.0, our approach reveals high superiority in terms of the F1-score, which has achieved an F1-score value of 97.64% for A. Alahmadi et al. [10], 98.08% for E. Abd El-Latif et al. [35], 97.52% for N. Kanwal et al. [20], 86.67% for H. Ding et al. [29], 95.4% for S. Nath et al. [28], and 98% for P. Niyishaka et al. [22], and the proposed model achieves an F1-score of 99.4%.

Over the CUISDE, the presented approach reveals high superiority in terms of the F1-score, which has achieved an F1-score value of 97.68% for A. Alahmadi et al. [10], 66.0% for K. Kadam et al. [30], 94.98% for H. Ding et al. [29], and 91.73% for Y. Zhang et al. [23], and the proposed model achieves an F1-score of 100%.

Additionally, the time of our model was compared to those in these recently published papers, as shown in Table 6, which achieved a speed of 156 s for Alahmadi et al. [10], 143 s for Kanwal et al. [20], and 280 s for Kadam et al. [30], and the proposed model achieves a speed of 15.7 s over the CASIA 1.0 dataset. When tested on the CASIA 2.0 dataset, the proposed model took the shortest time of 15.7 s compared with 326 s for Alahmadi et al. [10] and 234 s for Kanwal et al. [20]. When tested on the proposed model on the CUISDE dataset, it took 7.54 s compared with 126 s for Alahmadi et al. [10], 193 s for Kanwal et al. [20], and 295.2 s for Kadam et al. [30]. The results of Table 6 have been depicted in Figure 6.

Table 6. A comparison of speed recognition (time) in the case of the proposed model and other recently published approaches over the CASIA 1.0, CASIA 1.0, and CUISDE dataset.

	Speed Recognition (Time)		
	CASIA 1.0	**CASIA 2.0**	**CUISDE**
A. Alahmadi et al. [10]	156	326	126
N. Kanwal et al. [20]	143	234	193
K. Kadam et al. [30]	280	-	295.2
Proposed method	15.7	220	7.54

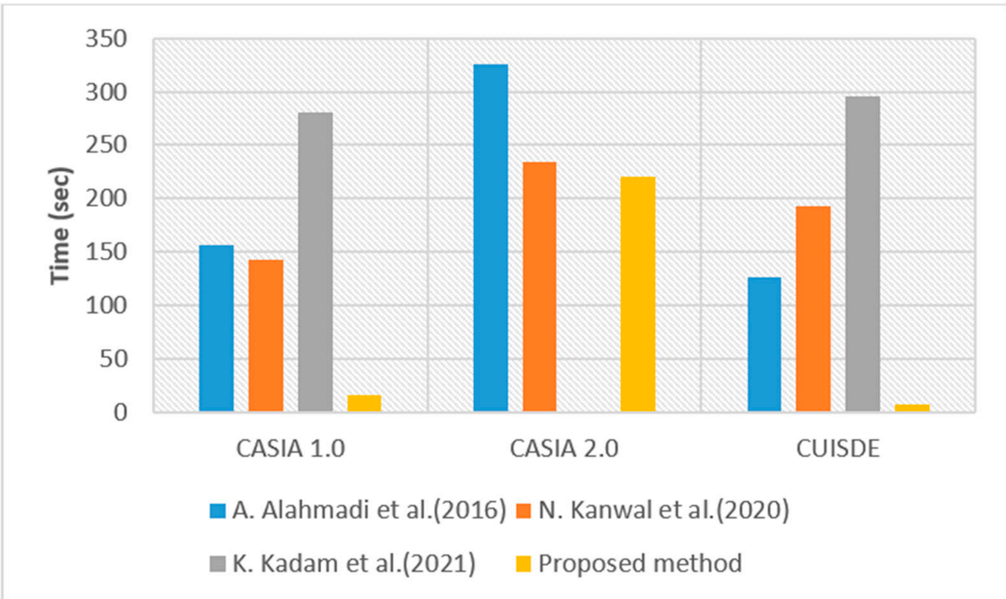

Figure 6. Compares the proposed method and the various approaches in terms of speed recognition (time) on the CASIA 1.0, CASIA 2.0, and CUISDE datasets [11,21,31].

The proposed model achieved high accuracy with a small number of parameters when compared to recently published approaches, as shown in Table 7.

In the CASIA 1.0 dataset, the proposed model achieved 97.0% accuracy for A. Alahmadi et al. [10] in the number of parameter "16,458,966", 98.25% accuracy for N. Kanwal et al. [20] in the number of parameter "18,534,965", 64.0% accuracy for K. Kadam et al. [30] in the number of parameter "23,812,574" and 99.1% accuracy for the proposed model in the number of parameter 97,698.

Table 7. A comparison of accuracy and number of parameters in the case of the proposed model and other recently published approaches over the CASIA 1.0, CASIA 1.0, and CUISDE dataset.

	CASIA 1.0		CASIA 2.0		CUISDE	
	Accuracy %	**Parameter**	**Accuracy %**	**Parameter**	**Accuracy %**	**Parameter**
A. Alahmadi et al. [10]	97.0	16,458,966	97.5	16,458,966	97.77	16,458,966
E. Abd El-Latif et al. [35]	94.55	-	96.36	-	-	-
N. Kanwal et al. [20]	98.25	18,534,965	97.59	18,534,965	96.66	18,534,965
K. Kadam et al. [30]	64.0	23,812,574	-	-	64.0	23,812,574
H. Ding et al. [29]	-	-	97.93		97.27	-
S. Nath et al. [28]	-	-	96.45	-	-	-
P. Niyishaka et al. [22]	-	-	94.59	2,542,144	-	-
Y. Zhang et al. [23]	-	-	-	-	91.46	-
V. Itier et al. [34]	-	-	-	-	98.13	-
Proposed method	99.1	97,698	99.3	97,698	100	97,698

In the CASIA 2.0 dataset, the proposed model achieved 97.5% accuracy for A. Alahmadi et al. [10] in the number of parameter "16,458,966", 97.59% accuracy for N. Kanwal et al. [20] in the number of parameter "18,534,965", 94.59% accuracy for P. Niyishaka et al. [22] in the number of parameter "2,542,144" and 99.3% accuracy for the proposed model in the number of parameter 97,698.

In the CUISDE dataset, the proposed model achieved 97.77% accuracy for A. Alahmadi et al. [10] in the number of parameter "16,458,966", 96.66% accuracy for N. Kanwal et al. [20] in the number of parameter "18,534,965", 64.0% accuracy for K. Kadam et al. [30] in the number of parameter "23,812,574" and 100% accuracy for the proposed model in the number of parameter 97,698. The results of Table 7 have been depicted in Figure 7.

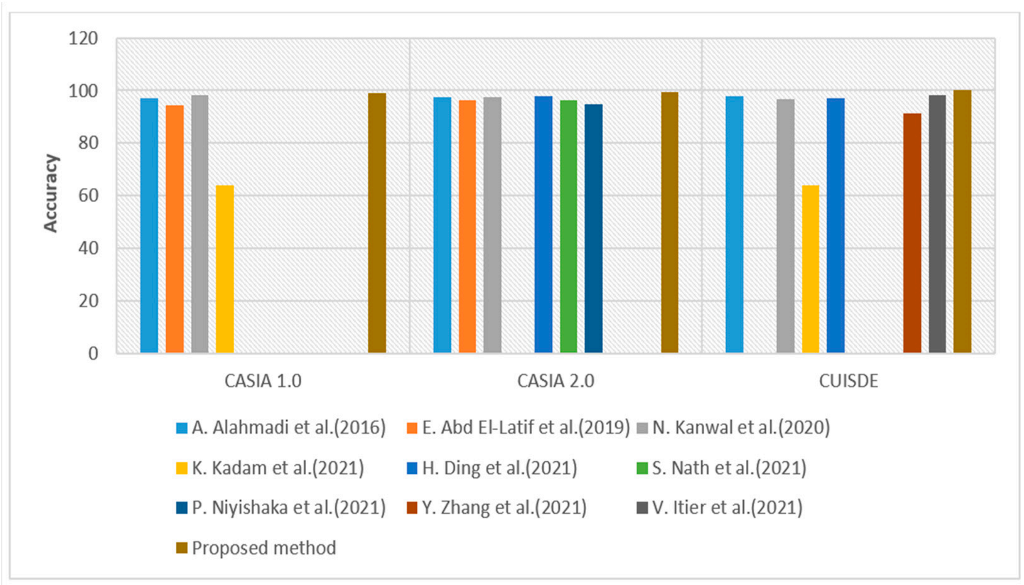

Figure 7. Compares the proposed method and the various approaches in terms of accuracy on the CASIA 1.0, CASIA 2.0, and CUISDE datasets [11,21,23,24,29–31,35,36].

5. Conclusions

This study introduces an efficient and lightweight approach for ISFD. We create a lightweight CNN model that gives high accuracy and compares the recent other used methods such Markov features in DCT and DWT domain, PCA, SVM, and C2RNet. Good results were conducted with appropriate convolutional layers and max-pooling layers. These results revealed that the proposed model is efficient and accurate against the other discovered models. Our experiments were achieved based on benchmark datasets: CASIA 1.0, CASIA 2.0 and CUISDE. The proposed model reached an F1-score of 99.14%, 99.4%, and 100% in CASIA 1.0, CASIA 2.0, and CUISDE, respectively. On the other hand, the presented model achieved an accuracy of 99.1%, 99.3%, and 100% for CASIA 1.0, CASIA 2.0, and CUISDE, respectively. Moreover, our model obtained this accuracy in the number of parameters (97,698) for CASIA 1.0, CASIA 2.0, and CUISDE. Overall, the proposed model can be a strong tool for detecting image splicing forgery in the real world.

The proposed model is distinguished from previous works in that it achieves high accuracy and uses fewer parameters compared to previous works. Reducing the number of parameters enables us to implement this model in an environment with limited capabilities for memory and processors.

6. Future Work

The proposed approach worked only with the image splicing forgery problem; however, there are no experiments on other problems such as medical images or other types of forgery to prove its effectiveness in dealing with images in general.

Author Contributions: Methodology, K.M.H., A.M.M., N.A.L. and M.M.F.; software, A.M.M.; investigation, K.M.H., A.M.M., N.A.L. and M.M.F.; writing—original draft, K.M.H., A.M.M., N.A.L. and M.M.F.; supervision, K.M.H. and N.A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is available on request.

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

References

1. Marcon, F.; Pasquini, C.; Boato, G. Detection of Manipulated Face Videos over Social Networks: A Large-Scale Study. *J. Imaging* **2021**, *7*, 193. [CrossRef]
2. Bi, X.; Zhang, Z.; Xiao, B. Reality Transform Adversarial Generators for Image Splicing Forgery Detection and Localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 14274–14283. [CrossRef]
3. Eltoukhy, M.M.; Elhoseny, M.; Hosny, K.M.; Singh, A.K. Computer aided detection of mammographic mass using exact Gaussian–Hermite moments. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *247*, 1–9. [CrossRef]
4. Ross, A.; Banerjee, S.; Chowdhury, A. Security in smart cities: A brief review of digital forensic schemes for biometric data. *Pattern Recognit. Lett.* **2020**, *138*, 346–354. [CrossRef]
5. Velmurugan, S.; Subashini, T.; Prashanth, M. Dissecting the literature for studying various approaches to copy move forgery detection. *Int. J. Adv. Sci.* **2020**, *29*, 6416–6438.
6. Kadam, K.D.; Ahirrao, S.; Kotecha, K. Efficient Approach towards Detection and Identification of Copy Move and Image Splicing Forgeries Using Mask R-CNN with MobileNet V1. *Comput. Intell. Neurosci.* **2022**, *2022*, 6845326. [CrossRef]
7. He, Z.; Lu, W.; Sun, W.; Huang, J. Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recognit.* **2012**, *45*, 4292–4299. [CrossRef]
8. Zhao, X.; Li, S.; Wang, J.L.; Yang, K. Optimal Chroma-like channel design for passive color image splicing detection. *EURASIP J. Adv. Signal Process.* **2012**, *2012*, 240. [CrossRef]
9. Su, B.; Yuan, Q.; Wang, S.; Zhao, C.; Li, S. Enhanced state selection Markov model for image splicing detection. *EURASIP J. Wirel. Commun. Netw.* **2014**, *2014*, 7. [CrossRef]
10. Alahmadi, A.; Hussain, M.; Aboalsamh, H.; Muhammad, G.; Bebis, G.; Mathkour, H. Passive detection of image forgery using DCT and local binary pattern. *Signal Image Video Process.* **2016**, *11*, 81–88. [CrossRef]
11. Sunitha, K.; Krishna, A. Efficient Keypoint based Copy Move Forgery Detection Method using Hybrid Feature Extraction. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; pp. 670–675. [CrossRef]
12. Moghaddasi, Z.; Jalab, H.A.; Noor, R.M.; Aghabozorgi, S. Improving RLRN Image Splicing Detection with the Use of PCA and Kernel PCA. *Sci. World J.* **2014**, *2014*, 606570. [CrossRef]
13. Muhammad, E.; Qureshi, A. Combining spatial and DCT based Markov features for enhanced blind detection of image splicing. *Pattern Anal. Appl.* **2015**, *18*, 713–723.

14. Zhang, Q.; Lu, W.; Weng, J. Joint image splicing detection in DCT and Contourlet transform domain. *J. Vis. Commun. Image Represent.* **2016**, *40*, 449–458. [CrossRef]
15. Pun, C.-M.; Liu, B.; Yuan, X.-C. Multi-scale noise estimation for image splicing forgery detection. *J. Vis. Commun. Image Represent.* **2016**, *38*, 195–206. [CrossRef]
16. Li, C.; Ma, Q.; Xiao, L.; Li, M.; Zhang, A. Image splicing detection based on Markov features in QDCT domain. *Neurocomputing* **2017**, *228*, 29–36. [CrossRef]
17. Zeng, H.; Zhan, Y.; Kang, X.; Lin, X. Image splicing localization using PCA-based noise level estimation. *Multimedia Tools Appl.* **2016**, *76*, 4783–4799. [CrossRef]
18. Zhu, N.; Li, Z. Blind image splicing detection via noise level function. *Signal Process. Image Commun.* **2018**, *68*, 181–192. [CrossRef]
19. Moghaddasi, Z.; Jalab, H.A.; Noor, R. Image splicing forgery detection based on low-dimensional singular value decomposition of discrete cosine transform coefficients. *Neural Comput. Appl.* **2018**, *31*, 7867–7877. [CrossRef]
20. Kanwal, N.; Girdhar, A.; Kaur, L.; Bhullar, J.S. Digital image splicing detection technique using optimal threshold based local ternary pattern. *Multimedia Tools Appl.* **2020**, *79*, 12829–12846. [CrossRef]
21. Revi, K.R.; Wilscy, M.; Antony, R. Portrait photography splicing detection using ensemble of convolutional neural networks. *J. Intell. Fuzzy Syst.* **2021**, *41*, 5347–5357. [CrossRef]
22. Niyishaka, P.; Bhagvati, C. Image splicing detection technique based on Illumination-Reflectance model and LBP. *Multimed. Tools Appl.* **2021**, *80*, 2161–2175. [CrossRef]
23. Zhang, Y.; Shi, T. Image Splicing Detection Scheme Based on Error Level Analysis and Local Binary Pattern. *Netw. Intell.* **2021**, *6*, 303–312.
24. Chen, B.; Qi, X.; Sun, X.; Shi, Y.-Q. Quaternion pseudo-Zernike moments combining both of RGB information and depth information for color image splicing detection. *J. Vis. Commun. Image Represent.* **2017**, *49*, 283–290. [CrossRef]
25. Salloum, R.; Ren, Y.; Kuo, C.-C.J. Image Splicing Localization using a Multi-task Fully Convolutional Network (MFCN). *J. Vis. Commun. Image Represent.* **2018**, *51*, 201–209. [CrossRef]
26. 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.* **2019**, *511*, 172–191. [CrossRef]
27. Ahmed, B.; Gulliver, T.A.; AlZahir, S. Image splicing detection using mask-RCNN. *Signal Image Video Process.* **2020**, *14*, 1035–1042. [CrossRef]
28. Nath, S.; Naskar, R. Automated image splicing detection using deep CNN-learned features and ANN-based classifier. *Signal Image Video Process.* **2021**, *15*, 1601–1608. [CrossRef]
29. Ding, H.; Chen, L.; Tao, Q.; Fu, Z.; Dong, L.; Cui, X. DCU-Net: A dual-channel U-shaped network for image splicing forgery detection. *Neural Comput. Appl.* **2021**, *1710*, 1–17. [CrossRef]
30. Kadam, K.D.; Ahirrao, S.; Kotecha, K.; Sahu, S. Detection and Localization of Multiple Image Splicing Using MobileNet V1. *IEEE Access* **2021**, *9*, 162499–162519. [CrossRef]
31. Hosny, K.M.; Mortda, A.M.; Fouda, M.M.; Lashin, N.A. An Efficient CNN Model to Detect Copy-Move Image Forgery. *IEEE Access* **2022**, *10*, 48622–48632. [CrossRef]
32. 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.
33. Hsu, Y.F.; Chang, S.F. Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 549–552.
34. Itier, V.; Strauss, O.; Morel, L.; Puech, W. Color noise correlation-based splicing detection for image forensics. *Multimed. Tools Appl.* **2021**, *80*, 13215–13233. [CrossRef]
35. El-Latif, E.I.A.; Taha, A.; Zayed, H.H. A Passive Approach for Detecting Image Splicing using Deep Learning and Haar Wavelet Transform. *Int. J. Comput. Netw. Inf. Secur.* **2019**, *11*, 28–35. [CrossRef]
36. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6. [CrossRef]
37. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef]
38. Musallam, A.S.; Sherif, A.S.; Hussein, M.K. A New Convolutional Neural Network Architecture for Automatic Detection of Brain Tumors in Magnetic Resonance Imaging Images. *IEEE Access* **2022**, *10*, 2775–2782. [CrossRef]