Deep Learning Model for Detecting Copy-Move Attack in Images: Testing and Verification

Chapter 9

Deep Learning Model for Detecting Copy-Move Attack in Images: Testing and Verification

Aleksandra Pauls [1]
[1] Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, 60131 Ancona, Italy
Email: a.pauls@pm.univpm.it
https://orcid.org/0000-0002-4780-6277

Luca Romeo [1,2]
[1] Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, 60131 Ancona, Italy
[2] Department Economics and Law, University of Macerata, Via Crescimbeni, 30/32, 62100 Macerata, Italy
Email: luca.romeo@unimc.it
https://orcid.org/0000-0003-1707-0147

Riccardo Rosati [1]
[1] Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, 60131 Ancona, Italy
Email: r.rosati@pm.univpm.it
https://orcid.org/0000-0003-3288-638X

Primo Zingaretti [1]
[1] Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, 60131 Ancona, Italy
Email: p.zingaretti@staff.univpm.it
https://orcid.org/0000-0002-5709-2159

Emanuele Frontoni [3]
[3] Department of Political Sciences, Communication and International Relations, University of Macerata, Via Crescimbeni, 30/32, 62100 Macerata, Italy
Email: emanuele.frontoni@unimc.it
https://orcid.org/0000-0002-8893-9244

Oleksandr Kuznetsov [4*]
[4] Department of Theoretical and Applied Sciences, eCampus University, Via Isimbardi 10, Novedrate (CO), 22060, Italy
Email: oleksandr.kuznetsov@uniecampus.it
https://orcid.org/0000-0003-2331-6326

***Corresponding Author:** Oleksandr Kuznetsov (oleksandr.kuznetsov@uniecampus.it)

Abstract

The modern level of digital technology leads to the widespread use of multimedia content: photo and video images, audio, texts, presentations, etc. This trend is significantly enhanced with the development of Internet technologies. Today, every gadget connected to the Internet continuously creates, processes, and sends huge amounts of multimedia information: in social networks; news channels; advertising mailings; messengers and much more. Obviously, this greatly simplifies communication between people and provides convenient, fast, and reliable information services. However, every new technology can be used for damaging purposes. For example, the Internet is currently overflowing with fake photo and video content. Fake multimedia has an extremely negative aspect: it devalues intellectual property and copyright; it compromises objective journalism; it causes reputational and material damage and much more. All this makes to develop and constantly improve new technologies for detecting fakes, verify them repeatedly and test them in various Internet applications. This article explores a well-known type of image spoofing based on copy-move attack. This simple attack can be implemented quickly even in automatic mode, but it is extremely difficult to detect fake images in a huge stream of multimedia data. We consider a deep learning model using convolutional neural networks and perform numerous tests on different datasets. We show that this approach can indeed be used to detect some fake images. However, to build a universal protection mechanism, it is necessary to significantly extend the datasets and take into account the peculiarities of copy-move attacks.

Keywords: Image spoofing; Copy-move attack; Deep learning models; Convolutional neural networks; Testing and verification.

9.1. Introduction

With the development of modern technology and digital imaging tools, photos have become an essential part of our lives. They are used in various fields: journalism, law, military, medicine, culture, news sources, social networks, and many others [1], [2]. Photographs of military and political events, scanned references and documents, hospital documents, X-ray images, photos from exhibitions and presentations, and more. In the age of digitalization, real-life events become a set of pixels, in the form of photo images.

The Internet and social media have also become a growing field. Due to the wide-spread use of digital devices such as cameras and smartphones, the amount of graphic content circulating on the Internet is constantly increasing every day. On the one hand, this availability of photo creation carries a positive message, as it is now much easier to capture real life. However, on the other hand, advanced software capabilities on smartphones and computers make it easy, even for an inexperienced user, to create a fake image and misinform society [3], [4]. It is also worth noting that it is sometimes physically impossible to distinguish a fake from the original with the human eye.

It is possible to obtain a fake image using various methods, such as copy-move, deleting objects, stitching images, scaling, morphing, retouching, and many others [2], [3]. Copy-move implies that a copy of a certain area of an image is created and moved to a new location [5].

The stitching method implies that the final image con-sists of parts of other images [6]. Other spoofing attacks use different graphical primi-tives: scaling allows resizing objects; morphing is created by merging shapes to obtain a new shape; retouching allows changing color, contrast, brightness, lighting, hiding visible defects and imperfections in order to improve or modify the original.

Thus, detecting fakes in images is an important task because successful data falsi-fication can mislead others, e.g., devalue intellectual property and copyright; com-promise objective journalism; cause reputational and material damage, and entail other negative scenarios. The solution to this kind of problem can be the deep learn-ing. The advantage of neural networks is that they independently identify the fea-tures of images, because at the moment there is no effective universal algorithm for detecting fake images that can take into account all the features of the image.

This paper discusses copy-move attacks and deep learning models to detect them. This new direction of research to protect against such attacks was proposed in [7]. According to data published in [7], the convolutional neural network (CNN) model trained on datasets MICC-F220, MICC-F2000, MICC-F600 [8] gives the best results on successful copy-move attack detection. The purpose of our paper is to build a deep learning model to improve and further generalize these results on new unseen dataset.

9.2. Contributions of the Chapter

This chapter makes several key contributions to the field of copy-move forgery detection:

1. Comprehensive Analysis: We provide a thorough review and synthesis of state-of-the-art techniques in copy-move forgery detection, encompassing both traditional and deep learning-based approaches. This analysis offers valuable insights into the strengths and limitations of current methods.

2. Novel CNN Architecture: We introduce a new CNN architecture specifically tailored for copy-move forgery detection. This model incorporates innovative feature fusion techniques and attention mechanisms designed to improve detection accuracy and robustness.

3. Extensive Experimental Validation: Our approach is rigorously evaluated on multiple benchmark datasets, including challenging cases that simulate real-world forgery scenarios. This comprehensive testing provides a clear picture of the model's performance across various conditions.

4. Robustness Analysis: We conduct a detailed exploration of the model's performance across various image transformations and post-processing operations. This analysis offers insights into the model's resilience and identifies areas for future improvement.

5. Interpretability Enhancements: We investigate methods to improve the interpretability of our deep learning model's decisions. This work aims to bridge the gap between the black-box nature of deep learning models and the need for explainable forensic evidence.

6. Comparative Study: We provide a thorough comparison of our proposed method against existing state-of-the-art techniques, offering a clear perspective on the advancements made by our approach.

9.3. Organization of the Chapter

The remainder of this chapter is organized as follows:

Section 2: Literature Review - This section provides a comprehensive review of related work in copy-move forgery detection. We trace the evolution of detection techniques from early handcrafted feature approaches to recent deep learning-based methods.

Section 3: Methodology - Here, we detail our proposed methodology, including the architecture of our deep learning model and the rationale behind its design. We describe the novel components of our approach and how they address the challenges identified in current methods.

Section 4: Experimental Setup and Results - This section presents our experimental framework, including dataset descriptions, implementation details, and evaluation metrics. We then provide a comprehensive presentation of our results, including comparisons with existing methods.

Section 5: Discussion - In this section, we offer a thorough analysis of our findings. We discuss the implications of our results, the strengths and limitations of our approach, and potential areas for future research.

Section 6: Conclusion - The chapter concludes with a summary of our contributions and their significance to the field of copy-move forgery detection. We also outline promising directions for future work in this rapidly evolving area of digital image forensics.

Through this structured approach, we aim to provide a comprehensive exploration of our novel method for copy-move forgery detection, contextualizing our work within the broader landscape of digital image forensics and offering valuable insights for both researchers and practitioners in the field.

9.4. Related works

Copy-move forgery detection has been an active area of research in digital image forensics for over a decade. Early seminal work in this field emerged from the Images and Communication Lab at the University of Florence. Amerini et al. (2010, 2011) [9], [10] proposed an influential methodology based on Scale Invariant Feature Transform (SIFT) features to detect and localize copy-move forgeries. Their approach demonstrated robustness to various geometric transformations applied during the forgery process.

Building on this foundation, several researchers extended and refined SIFT-based techniques. Caldelli et al. (2012) [11] investigated the effectiveness of local warping attacks against SIFT-based detection, highlighting potential vulnerabilities. Amerini et al. (2013) [12] introduced a robust clustering approach using J-Linkage to improve forgery localization, particularly for challenging cases involving distant copied regions. The same year, Amerini et al. (2013) [13] explored counter-forensic techniques, proposing methods to classify and selectively remove SIFT keypoints to evade detection.

While SIFT remained a dominant approach, other researchers began exploring complementary techniques. Hashmi et al. (2014) [14] combined wavelet transforms with SIFT features, leveraging multi-resolution analysis to enhance detection performance. Prajapati et al. (2015) [6] integrated the RANSAC algorithm with SIFT matching to improve the accuracy of tampered region localization.

As the field matured, several comprehensive surveys emerged, synthesizing the growing body of literature. Warif et al. (2016) [15] provided an extensive overview of copy-move forgery detection techniques, categorizing approaches based on block-based and keypoint-based feature extraction methods. More recently, Nabi et al. (2022) [16] conducted a broader survey covering both image and video forgery detection, situating copy-move techniques within the larger landscape of multimedia forensics.

The advent of deep learning techniques has opened new avenues for copy-move forgery detection, marking a significant shift from hand-crafted feature extraction to learned representations. Elaskily et al. (2020) [7] proposed a convolutional neural network (CNN) architecture specifically designed for this task, demonstrating impressive performance across multiple benchmark datasets. Building on this foundation, Zhang et al. (2022) [17] introduced a Feature Compensation Network (FCNet) that addresses the challenge of detecting manipulations with arbitrary parameters. Their approach incorporates feature enhancement, sensitivity estimation, and adaptive average pooling to improve detection accuracy in complex scenarios.

Recent years have seen a surge in innovative approaches to copy-move forgery detection. Lin et al. (2024) [18] proposed a novel scheme based on Regional Density Center (RDC) clustering and Refined Length Homogeneity Filtering (RLHF). Their method employs scale normalization and contrast threshold adjustment to obtain adequate keypoints in challenging image areas, demonstrating improved effectiveness and robustness on public datasets. Shehin and Sankar (2024) [19] developed an algorithm utilizing Discrete Cosine Transform and eigenvalues to detect and localize copy-move duplications, showing particular resilience to post-processing rotation attacks. Wang et al. (2024) [20] introduced the Strong Robust Copy-Move Forgery Detection Network (DRNet), which employs a layer-by-layer decoupling refinement approach to extract semantically irrelevant shallow information, achieving state-of-the-art performance on multiple datasets.

As digital manipulation tools continue to evolve, so too do forensic techniques. Zhong et al. (2024) [21] proposed a Hierarchical Coarse-to-Fine framework for video copy-move forgery detection, addressing challenges in both intra-frame and inter-frame forgeries. Their approach demonstrates improved efficiency and accuracy, even under various adverse conditions. These recent advancements reflect the field's ongoing efforts to improve detection accuracy, computational efficiency, and robustness to various post-processing operations and anti-forensic techniques. As the complexity of digital forgeries increases, research in this domain remains critical for ensuring the integrity and authenticity of digital media across various applications, from social media to legal evidence.

The most interesting in our opinion is the work [7], in which the authors proposed to use deep learning methods for automatic detection of copy-move forgeries. In this work a specially developed CNN was trained on the most common datasets. The results published in [7] look

very encouraging. For example, the authors report that 100% accuracy was achieved for all the datasets studied.

The purpose of our paper is to independently verify the results from [7] as well as the conclusions and deductions made by the authors. In addition, we are analyzing the architecture of CNN and the conclusions drawn as a result may be a reason to improve the model and for developing new CNN architectures for copy-move attack detection.

9.5. Methodology

The deep learning model given in [7] is based on the multilayer CNN architecture, which consists in alternating convolution layers and pooling layers. In convolution layers, each image fragment is multiplied step by step by a special matrix (convolu-tion kernel), the obtained result is summarized and written in the same position of the output image. Thus, at each convolution layers there is a transition from specific image features to more abstract details. The abstraction increases from layer to layer, culminating in the isolation of high-level abstract concepts. Intuitively, this process can be described as "pattern recognition", and in the process of learning, the network adjusts itself to highlight salient features while filtering out unimportant details of the image.

A simplified architecture of CNN is shown in Fig. 9.1.

Thus, CNN is based on the convolution operation. When the network is trained, the image is processed in fragments. Each fragment is sequentially multiplied by a con-volution kernel (or filter), which is shifted across the image. The values are then summed, and the result is written to the corresponding position of the output image (Fig. 1). However, the use of such convolution leads to the extraction of too many selected features. To reduce these features, layers of subsampling (pooling) are used. As a result, a full-coherent layer is formed, where the probability distribution for dif-ferent classes is performed based on the obtained data.

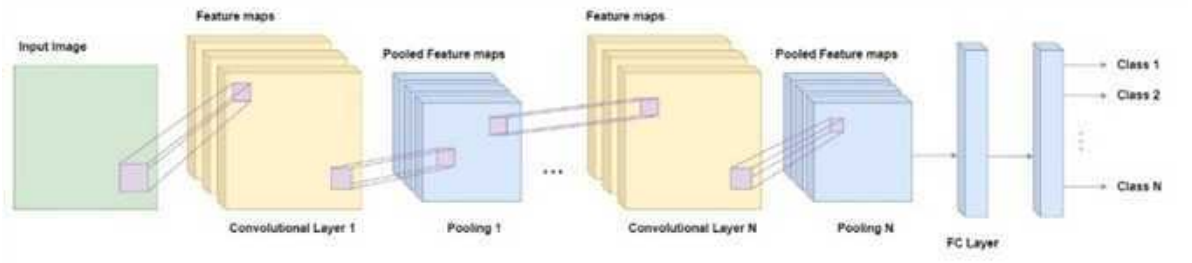


Figure 9.1: A simplified architecture of CNN

The advantages of convolutional neural networks include the possibility to reduce the number of trained parameters and increase the training speed, compared with the fully-connected (FC) network, the possibility of parallel computing and implementa-tion of network training algorithms on graphics processors (GPUs), as well as stability to shifting the object position in the input data [28], [29], [30].

9.5.1. The Copy-Move Forgery Detection (CMFD) network

The Copy-Move Forgery Detection (CMFD) algorithm is proposed in [7]. It is based on CNN architecture.

The CNN architecture from [7] is schematically presented in Fig. 9.2. It contains 6 convolution and MaxPooling layers, as well as the final merging operation using Global Average Pooling (GAP) [7]:

- CNV 1 (No. of samples, 224, 224, 16);
- Pooling 1 (No. of samples, 112, 112, 16);
- CNV 2 (No. of samples, 110, 110, 32);
- Pooling 2 (No. of samples, 55, 55, 32);
- CNV 3 (No. of samples, 53, 53, 64);
- Pooling 3 (No. of samples, 26, 26, 64);
- CNV 4 (No. of samples, 24, 24, 128);
- Pooling 4 (No. of samples, 12, 12, 128);
- CNV 5 (No. of samples, 10, 10, 256);
- Pooling 5 (No. of samples, 5, 5, 256);
- CNV 6 (No. of samples, 3, 3, 512);
- Pooling 6 (No. of samples, 1, 1, 512);
- Global Average Layer (No. of samples, 512);
- Dense (No. of samples, 2).

Traditionally, GAP is used to replace FC layers in classic CNNs. Instead of adding FC layers on top of object maps, the average value of each object map is used. The resulting vector is transferred directly to the softmax activation function. In this model GAP and Dense layers are used as a FC layer [7].

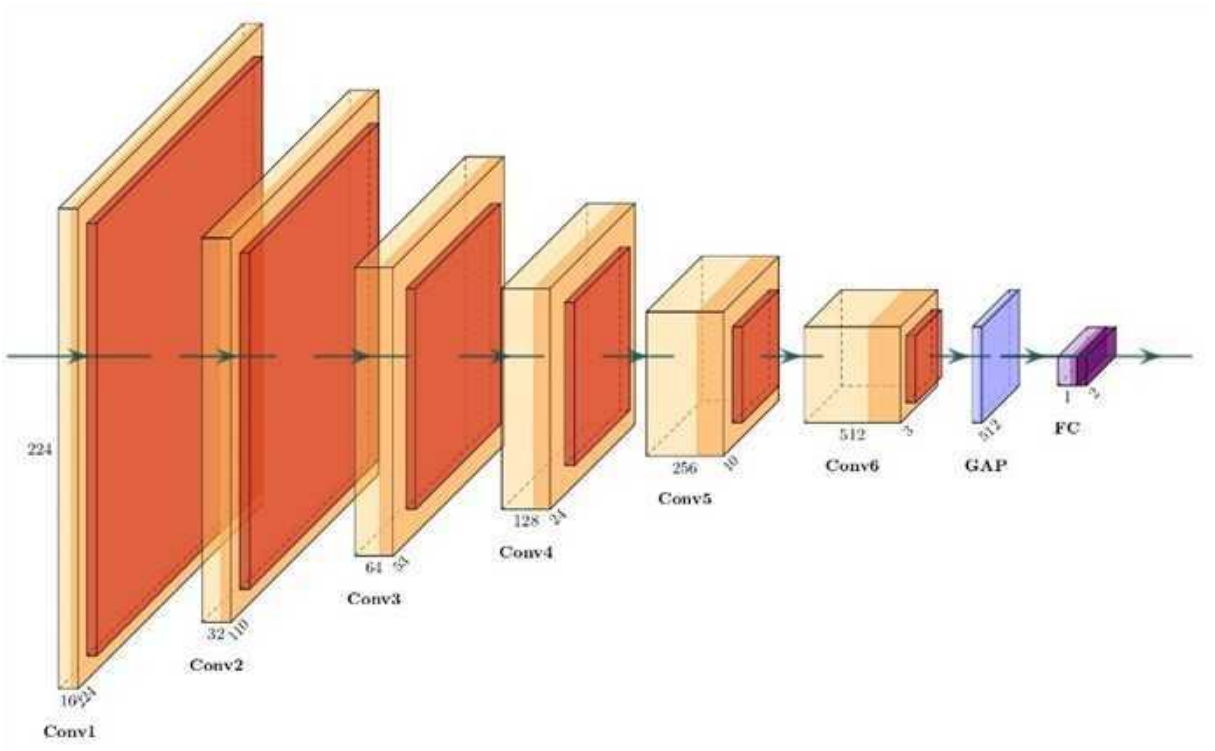


Figure 9.2: The architecture of the CNN model from [7]

Convolutional layers (CNV) are designed to extract features from the image through their filters. A different number of filters are used:

- 16 filters are used for the CNV 1;

- 32 filters are used for the CNV 2;
- 64 filters are used for the CNV 3;
- 128 filters are used for the CNV 4;
- 256 filters are used for the CNV 5;
- 512 filters are used for the CNV 6.

The final FC layer has a softmax activation function and is used to classify each class. This activation function is usually used for multiclass classification problems since it distributes the probability among several categories. However, in [7] this acti-vation function is used for binary classification:

- first class - the original image (without copy-move attack tampering);
- second class - fake image (detected fake copy-move attack).

Adaptive Moment Estimation (Adam) was used to train the neural network in [7]. It is an optimization method algorithm for gradient descent, which is effective when dealing with a large problem involving a large number of data or parameters. In its essence, Adam is a combination of the "gradient descent with momentum" and "Root mean square prop" (RMSP) algorithm [31]. A cross-entropy (or logarithmic loss function – LogLoss) was used as a loss function to quantify the difference between the probability distributions [7].

Cross-entropy is defined as [32]:

$$H(p,q) = \sum_{x} p(x) \log q(x),$$

where p – is the distribution of true responses, a q – the distribution of model predic-tion probabilities.

For the case of binary classification (as in our case), the formula for calculating cross-entropy will look like this:

$$LogLoss = -\log(p), \quad -\log(1-p), \tag{9.1}$$

where – the binary indicator (0 or 1) of true answers.

The base of the logarithm specifies the unit of measurement. Typically, a binary basis is used, and cross-entropy measures the average number of bits needed to rec-ognize an event from a feature set.

The article [7] did not specify what values were used for packet length and learn-ing rate. Therefore, we tested these parameters for different values:

- Batch size – 8, 16, 32 and 64;
- Learning rate – 0.01, 0.001 and 0.0001.

Our final implementation used: batch size – 16, learning rate – 0.001. We have vali-dated these hyper-parameters in a separate validation set.

As an experiment, we also implemented model improvement through cross-validation k-fold. The data were not separated into a training and validation sample. Instead, the entire data set was divided into folds, each of which became a test sample in turn.

9.5.2. Binary Classification Performance Metrics

The problem of copy-move attack detection belongs to the classical binary classifica-tion problem [33], [34], [35]. It consists in matching each element to one of two groups (clas-ses) based on some rule or feature. In other words, we need to match some image to one of the two groups:

- Class 0 for original images ("negative," no fakes);
- Class 1 for spoofed images ("positive," there is a spoof).

The classification problem can be solved correctly (when an element is correctly matched to its class) or incorrectly. We have 4 possible outcomes for binary classifi-cation (see Fig. 3). Fig. 3 shows the confusion matrix, also known as the error matrix [36]. The color behind Fig. 3 highlights the erroneous outcomes. Fig. 3 are labeled:

- Total population. It is a set of similar items or events of interest for a particular question or experiment. In our case, it is the total number of images as the sum of the number of images containing a fake (from class 1) and the number of images not containing a fake (from class 0);
- Predicted condition. These are positive and negative predicted states.

Table 9.1. Possible outcomes of binary classification

		Predicted condition	
	Total population $P + N$	Positive (PP) (belong to class 1)	Negative (PN) (belong to class 0)
Actual state	Positive (P) (belong to class 1)	True Positive (TP)	False Negative (FN)
	Negative (N) (belong to class 0)	False Positive (FP)	True Negative (TN)

When real states and predicted states overlap, we get 4 possible binary classification results:

- True Positive (TP) – when P and PP intersect (correctly classified fake image);
- True Negative (TN) – when N and PN intersect (the original image is correctly classified);
- False Positive (FP) – when N and PP intersect (misclassified the original image as a fake). This outcome is also known as a "Type I error" or "false alarm";
- False Negative (FN) – when P and PN intersect (mistakenly classified the fake image as original). This outcome is also known as "type II error" or "missed target".

Errors of the first and second kind are mutually symmetric, that is, if we swap the hypotheses about "positive" and "negative" (invert the notation "0" and "1" for clas-ses of elements), then errors of the first kind will turn into errors of the second kind and vice versa.

For each possible case, let us define the metric of binary classification efficiency as the frequency of the corresponding events:

- True Positive Rate (TPR) outcomes (this metric is also called sensitivity), which refers to the probability of a positive state, provided that it is actually positive, i.e.

$$TPR = \frac{TP}{P} = 1 - FNR, \tag{9.2}$$

- False Negative Rate (FNR) outcomes. This is the proportion of positive results that produce negative results, i.e.

$$FNR = \frac{FN}{P} = 1 - TPR, \tag{9.3}$$

- False Positive Rate (FPR) outcomes – it is the proportion of all negative results that produce positive classification results, i.e.

$$FPR = \frac{FP}{N} = 1 - TNR, \tag{9.4}$$

- True Negative Rate (TNR) outcomes refers to the probability of a negative test result, if it is indeed negative, i.e.

$$TNR = \frac{TN}{N} = 1 - FPR, \tag{9.5}$$

In addition to metrics (9.1) - (9.4), [7] also uses an accuracy. This is a statistical indicator of how well the binary classification test correctly identifies or excludes a condition. In other words, accuracy is the share of correct predictions (both true positives and true negatives) among the total number of cases considered.

$$\text{Accuracy} = \frac{TP + TN}{P + N}, \tag{9.6}$$

In our work, we also evaluated the measure as a harmonic mean of accuracy and sensitivity:

$$F_1 = \frac{2TP}{2TP + FP + FN}, \tag{9.7}$$

The highest possible value is 1.0, which indicates perfect accuracy and sensitivity, and the lowest possible value is 0 if the accuracy or sensitivity is zero.
Thus, we will use logarithmic loss function (9.1) as the main indicators of binary classification efficiency, as well as metrics (9.2) - (9.7), which characterize sensitivity, accuracy and other properties of the model.

9.5.3. Datasets

In the article [7] several datasets were used to train and test the deep learning model. In particular, sets MICC-F220, MICC-F2000, MICC-F600, given on the site of the research group Images and Communication Lab of the Media Integration and Communication Center (MICC) [8] were used:

In the article [7] several datasets were used to train and test the deep learning model. In particular, sets MICC-F220, MICC-F2000, MICC-F600, given on the site of the research group Images and Communication Lab of the Media Integration and Communication Center (MICC) [8] were used:

- The MICC-F220 data set consists of 220 images: 110 original images and 110 fake images. The images range in size from 722x480 to 800x600 pixels;
- The MICC-F600 dataset contains 760 photographic images, of which 440 original and 160 fake images were used. The dataset also contains 160 images of the ground-truth masks, where areas of duplicate parts are highlighted, but they were not used in the experiments due to their unsuitability for the task. The size of the images varies from 800x533 to 2014x3039 pixels;
- The MICC-F2000 dataset contains 2000 images: 1300 original and 700 fake images. Resolution: 1536x2048 or 2048x1536 pixels depending on image orientation.

Examples of original and fake images are shown in Fig. 9.3, 9.4, 9.5.

Figure 9.3: Examples of original and fake images from MICC-F220 dataset

Figure 9.4: Examples of original and fake images from MICC-F600 dataset


Figure 9.5: Examples of original and fake images from MICC-F2000 dataset

The main ways to produce fake photo images are to use copy-move attacks combining scaling and rotating the corresponding areas. For MICC-F220 and MICC-F2000 the shape of the regions is rectangular, for MICC-F600 the shape of the regions is streamlined. In addition, the authors of [7] used new sets obtained by combining and merging the MICC-F220, MICC-F2000, and MICC-F600 sets. All images were pretreated by resizing the image to 224x224 pixels, without cropping. After that they were normalized.

9.6. Experimental Setup and Results

Our implementation of the deep learning model from [7] is available on the open source GitHub https://github.com/raurica/Image-Forgery.git, which facilitates repro-duction and verification of our results. When implementing the model, we used Google Colaboratory platform (Python 3 server gas pedal based on Google Compute Engine, 12.68 GB RAM, 107.72 GB disk).

Tables 9.2 – 9.5 show the results of model training (state of the art model [7]) on different data sets for 15, 25, 35, 50, 75 and 100 epochs.

Table 9.2. Model training results on the MICC-F220 dataset (state of the art model [7])

	15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
(1) V_loss	1.0182	0.5336	0.6452	0.8371	0.6447	0.8515
(1) T_loss	0.2422	0.1265	0.1126	0.1257	0.0881	0.0511
(2) TPR	1.0	1.0	1.0	1.0	1.0	1.0
(3) FNR	0.0	0.0	0.0	0.0	0.0	0.0
(4) FPR	0.4545	0.1818	0.1818	0.1818	0.1818	0.1818
(5) TNR	0.5455	0.8182	0.8182	0.8182	0.8182	0.8182
(6) Acc	0.7727	0.9091	0.9091	0.9091	0.9091	0.9091
(7) F_1	0.815	0.917	0.917	0.917	0.917	0.917
T1	47ms/step	46ms/step	44ms/step	45ms/step	54ms/step	52ms/step
T2	4.516 sec	11.343 sec	9.21 sec	21.532 sec	21.918 sec	41.967 sec

Table 9.3. Model training results on the MICC-F600 dataset (state of the art model [7])

	15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
(1) V_loss	0.6149	0.7624	2.2725	3.0485	3.2168	4.2186
(1) T_loss	0.2695	0.1957	0.0077	0.0023	0.0064	0,0009
(2) TPR	0.875	0.625	0.875	0.875	0.875	0.875

(3) FNR	0.125	0.375	0.125	0.125	0.125	0.125
(4) FPR	0.2955	0.1364	0.1591	0.1818	0.25	0.2045
(5) TNR	0.7045	0.8636	0.8409	0.8182	0.75	0.7955
(6) Acc	0.75	0.8	0.85	0.8333	0.7833	0.8167
(7) F_1	0.651	0.625	0.757	0.737	0.683	0.718
T1	32ms/step	26ms/step	23ms/step	40ms/step	27ms/step	31ms/step
T2	10.29 sec	21.655 sec	18.8 sec	42.509 sec	38.537 sec	83.052 sec

Table 9.4. Model training results on the MICC-F2000 dataset (state of the art model [7])

	15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
(1) V_loss	0.6541	0.1394	0.1761	0.1772	0.1809	0.1329
(1) T_loss	0.6464	0.0524	0.0424	0.0365	0.0111	0.0351
(2) TPR	0.0	1.0	1.0	0.9583	0.9444	0.9722
(3) FNR	1.0	0.0	0.0	0.0417	0.0556	0.0278
(4) FPR	0.0	0.0938	0.0938	0.0469	0.0234	0.0312
(5) TNR	1.0	0.9062	0.9062	0.9531	0.9766	0.9688
(6) Acc	0.64	0.94	0.94	0.955	0.965	0.97
(7) F_1	0.0	0.923	0.923	0.939	0.951	0.959
T1	20ms/step	20ms/step	20ms/step	20ms/step	26ms/step	27ms/step
T2	25.818 sec	43.408 sec	60.935 sec	84.869 sec	143.96 sec	203.78 sec

Table 9.5. Model training results on the merged dataset from MICC-F220, MICC-F600 and MICC-F2000 (state of the art model [7])

	15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
(1) V_loss	0.2967	0.2549	0.2427	0.3140	0.4926	0.4529
(1) T_loss	0.1089	0.0723	0.0865	0.0434	0.0332	0.0198
(2) TPR	0.9505	0.9703	0.9208	0.9406	0.9406	0.9307
(3) FNR	0.0495	0.0297	0.0792	0.0594	0.0594	0.0693
(4) FPR	0.1159	0.0797	0.0725	0.1014	0.1087	0.0942
(5) TNR	0.8841	0.9203	0.9275	0.8986	0.8913	0.9058
(6) Acc	0.9121	0.9414	0.9456	0.9163	0.9121	0.9163
(7) F_1	0.901	0.933	0.912	0.905	0.9	0.904
T1	21ms/step	21ms/step	27ms/step	21ms/step	27ms/step	28ms/step
T2	33.704 sec	65.602 sec	83.991 sec	102.279 sec	203.964 sec	264.012 sec

For each dataset, the results were calculated using formulas (9.1) - (9.7). We give cross-entropy values (9.1) for both training and validation stages.

By analogy with [7] we also measured the testing time (T1), as the average time for checking images for k rounds of the testing operation, as well as the total testing time (T2). The values of this indicator are given in the last lines of Tables 9.2-9.5.

For clarity, Figs. 9.6-9.8 are plots of accuracy and loss function on the training and validation set for each dataset.

Tables 9.5, 9.7, 9.8 compare our results with the results from [7].

The learning procedure was stopped in the maximum number of epochs. The dataset was split by using train_test_split function. In this way we divide the dataset into random train and test subsets. where the test subset was 20%. Also, we tested different values of batch size (8, 16, 32, 64) and learning rate (0.01, 0.001, 0.0001) and then, chose the best ones from them. It used following hyper-parameters: batch size – 16, learning rate – 0.001.

In addition, in Tables 9.9, 9.10, 9.11, by analogy with the paper [7] we also give the re-sults of comparisons by metrics (9.2) - (9.5), these results are related to the test set.

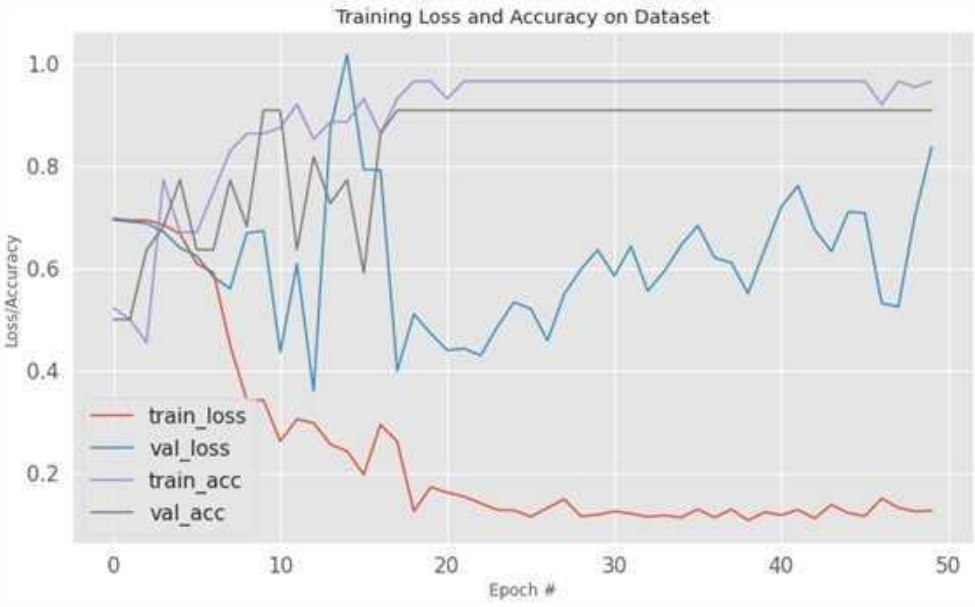


Figure 9.6: Accuracy and loss plots on the MICC-F220 dataset (50 epochs)

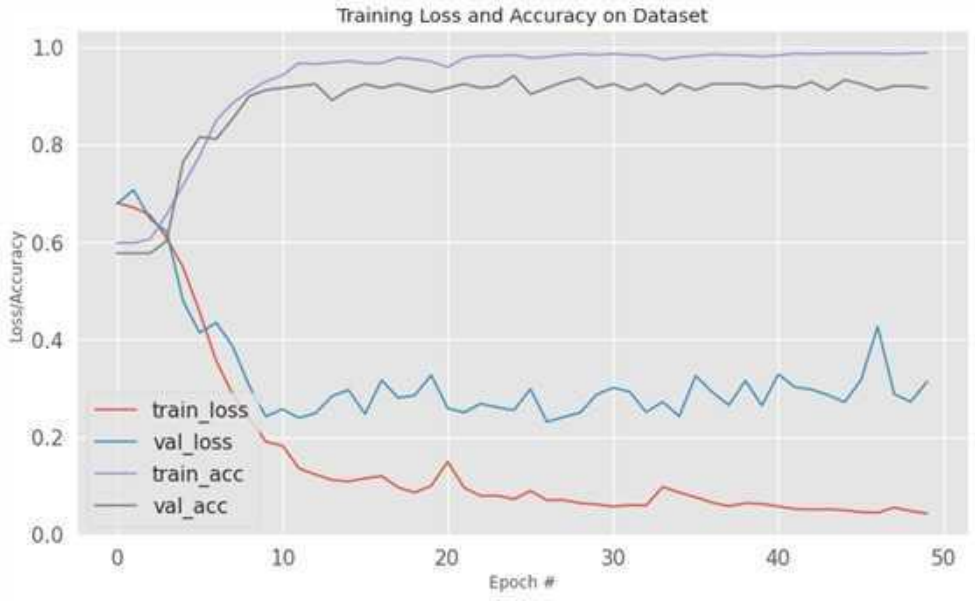


Figure 9.7: Accuracy and loss plots on the MICC-F2000 dataset (50 epochs)

Figure 9.8: Accuracy and loss plots on the merged dataset from MICC-F220, MICC-F600 and MICC-F2000 (50 epochs)

Table 9.6. Model training results on the MICC-F220 dataset

		15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
Our work	(1) V_loss	1.0182	0.5336	0.6452	0.8371	0.6447	0.8515
	(6) Acc	0.7727	0.9091	0.9091	0.9091	0.9091	0.9091
[7]	(1) V_loss	7.82	3.85	2.38	0	0	0
	(6) Acc	0.9218	0.9615	0.9762	1.00	1.00	1.00

Table 9.7. Model training results on the MICC-F600 dataset

		15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
Our work	(1) V_loss	0.6149	0.7624	2.2725	3.0485	3.2168	4.2186
	(6) Acc	0.75	0.8	0.85	0.8333	0.7833	0.8167
[7]	(1) V_loss	7.84	5.88	3.92	0	0	0
	(6) Acc	0.9215	0.9412	0.9608	1.00	1.00	1.00

Table 9.8. Model training results on the merged dataset from MICC-F220, MICC-F600, and MICC-F2000

		15 Epochs	**25 Epochs**	**35 Epochs**	**50 Epochs**	**75 Epochs**	**100 Epochs**
Our work	(1) V_loss	0.2967	0.2549	0.2427	0.3140	0.4926	0.4529

	(6) Acc	0.9121	0.9414	0.9456	0.9163	0.9121	0.9163
[7]	(1) V_loss	6.43	5.00	2.14	1.43	0	0
	(6) Acc	0.9357	0.9500	0.9786	0.9857	1.00	1.00

Table 9.9. Model training results on the MICC-F220 dataset

	[10]	[12]	[7]	**Our work**
(2) TPR	1.0	1.0	1.00	1.0
(3) FNR	0	0	0	0.0
(4) FPR	0	0.06	0	0.1818
(5) TNR	1.0	0.94	1.00	0.8182

Table 9.10. Model training results on the MICC-F600 dataset

	[10]	[12]	[7]	**Our work**
(2) TPR	0.692	0.816	1.00	0.875
(3) FNR	0.308	0.184	0	0.125
(4) FPR	0.125	0.0727	0	0.1818
(5) TNR	0.875	0.9273	1.00	0.8182

Table 9.11. Model training results on the MICC-F2000 dataset

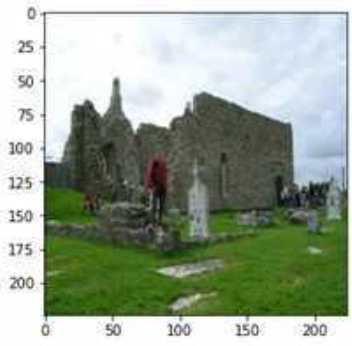
	[10]	[12]	[7]	**Our work**
(2) TPR	0.9342	0.9486	1.00	0.9583
(3) FNR	0.0658	0.0514	0	0.0417
(4) FPR	0.1161	0.0915	0	0.0469
(5) TNR	0.8839	0.9085	1.00	0.9531

It should be recognized that the general trends and patterns hold, the CNN from [7] is truly capable of correctly classifying images from the MICC-F220, MICC-F600, and MICC-F2000 datasets.

Fig. 9.9 shows examples of image classification from the validation sample (the model provides the probability that the image belongs to a particular class).

As we can see from the test results and the above examples, the deep learning model from [7] does correctly classify most of the images from the validation sam-ple. However, if we use images not from the validation sample for testing, the results will be significantly different. As an example, we generated several examples of pairs over to “original-fake" images. For each example we obtained classification results (see Fig. 9.9).

As we can see from the above examples, the deep learning model from [7] may not correctly detect images that were not included in the validation sample at all.



Original image (MICC-F220), Class 0. Obtained probabilities: for class 0 - 0.999, for class 1 - 0.001

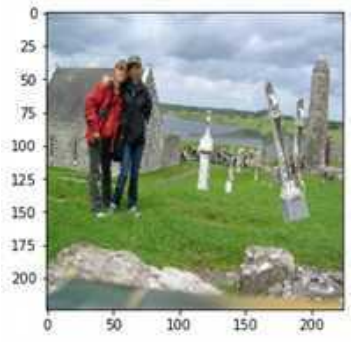


Image fake (MICC-F220), class 1. Obtained probabilities: for class 0 - 0.004, for class 1 - 0.996

Original image (MICC-F600), class 0. Obtained probabilities: for class 0 - 0.981, for class 1 - 0.019

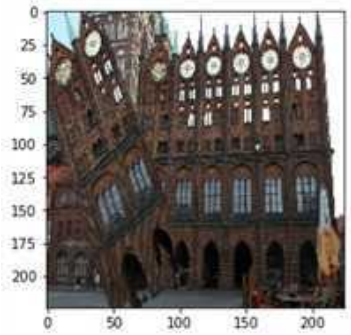
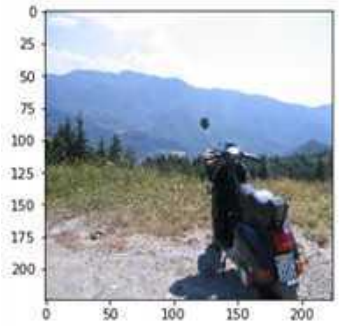


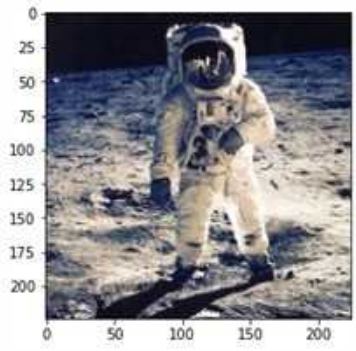
Image fake (MICC-F600), Class 1. Obtained probabilities: for class 0 - 0.001, for class 1 - 0.999



Original image (MICC-F2000), Class 0. Obtained probabilities: for class 0 - 0.992, for class 1 - 0.008

Image fake (MICC-F2000), class 1. Obtained probabilities: for class 0 - 0.017, for class 1 - 0.983

Figure 9.9: Examples of image classification from a validation set



Original image, class 0. Obtained probabilities: for class 0 - 0.999, for class 1 - 0.001

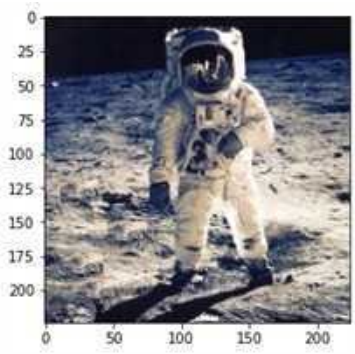
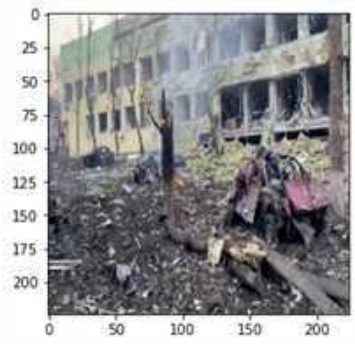


Image fake, class 1. Obtained probabilities: for class 0 - 0.999, for class 1 - 0.001



Original image, class 0. Obtained probabilities: for class 0 - 0.383, for class 1 - 0.616

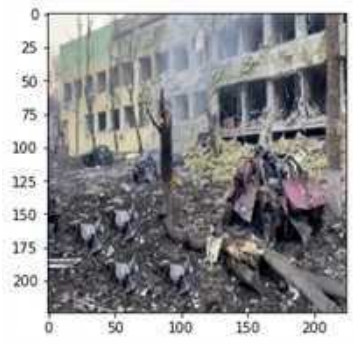
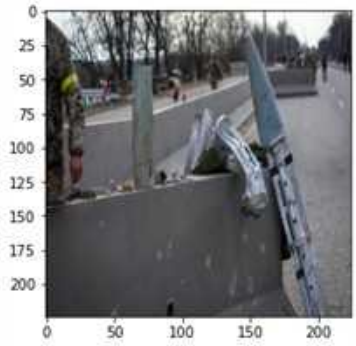


Image fake, class 1. Obtained probabilities: for class 0 - 0.413, for class 1 - 0.587



Original image, class 0. Obtained probabilities: for class 0 - 0.431, for class 1 - 0.569

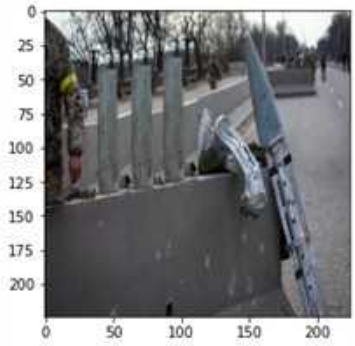


Image fake, class 1. Obtained probabilities: for class 0 - 0.763, for class 1 - 0.237

Figure 9.10: Examples of image classification from a validation set

9.7. Discussion

The results of this study provide valuable insights into the performance and limitations of deep learning models for detecting copy-move forgery in images. While the convolutional neural network approach proposed by Elaskily et al. [7] showed promise on the MICC datasets, our

independent implementation and testing revealed several key issues that warrant further investigation.

First, the accuracy metrics we obtained, though reasonably high, fell short of the near-perfect results reported in the original study. For example, on the MICC-F220 dataset, we achieved 90.91% accuracy after 50 epochs, compared to the reported 100%. This discrepancy could stem from differences in implementation details or hyperparameter settings not fully specified in [7]. More importantly, it highlights the challenge of reproducing state-of-the-art results in deep learning research - a growing concern in the field [37].

Second, our analysis of the loss function trajectories uncovered clear signs of overfitting, particularly on the smaller datasets. The divergence between training and validation loss as training progressed indicates the model was essentially memorizing training examples rather than learning generalizable features. This overfitting likely explains the model's poor performance on novel images outside the validation set, as demonstrated in our additional tests (Fig. 9.10).

The model's inability to reliably detect forgeries in images dissimilar to its training data is perhaps the most significant limitation revealed by our study. While achieving 90%+ accuracy on held-out data from the same distribution is impressive, it falls short of the robustness needed for real-world forensic applications. Adversaries could likely circumvent detection by using image manipulation techniques that differ from those in the training set.

These findings underscore the importance of diverse, large-scale datasets for developing practically useful forgery detection models. The MICC datasets, while valuable, appear insufficient to capture the full spectrum of possible copy-move forgeries. Future work should focus on curating more comprehensive datasets that encompass a wider range of image types, forgery techniques, and post-processing operations.

Additionally, architectural innovations may be necessary to improve the model's generalization capabilities. Techniques like transfer learning from models pre-trained on large natural image datasets could potentially help the network learn more robust, semantic features less prone to overfitting. Exploring alternative neural network architectures designed for anomaly detection, rather than binary classification, may also prove fruitful.

In conclusion, while deep learning shows promise for copy-move forgery detection, significant challenges remain in developing models that generalize beyond their training distribution. Our results suggest that claims of near-perfect accuracy on standard datasets should be interpreted cautiously, as they may not translate to real-world performance. Advancing the field will require not only algorithmic innovations, but also more rigorous evaluation protocols that assess models on diverse, previously unseen forgeries.

9.8. Conclusions

Based on the results obtained, we can conclude that, from a practical point of view, the approach to solving the problem of classification of original and fake images requires improvement. We confirm the main results and regularities obtained by the authors [7] using the MICC-F220, MICC-F600, and MICC-F2000 datasets. Although our results for metrics

(9.2)-(9.6) are a bit worse than what was declared in [7], it should however be recognized that the deep learning model considered does allow us to detect some fake images.

For a detailed analysis of the results, we should pay attention to the significant difference in the values of (9.1). The logarithmic loss function in our experiments is significantly higher than the results from [7]. In Tables 9.1-9.7 we specifically show the cross-entropy values calculated for the training and validation samples. While the cross-entropy in the training sample decreases (as it should), the values of the loss function in the validation sample may increase. The cross-entropy dependencies demonstrate the effect of overfitting, where the model has essentially memorized examples from the training sample and is unable to extract the key features necessary for correct prediction. In support of this thesis, we gave several examples (see Fig. 9.10) where third-party images (not from the validation sample) are not correctly detected.

Thus, the use of deep learning models to detect copy-move attacks requires further justification. To build a universal protection mechanism, it is necessary to significantly extend the datasets and consider the peculiarities of the attack.

References

[1] R. Ridzoň and D. Levický, "Multimedia security and multimedia content protection," in *2009 International Symposium ELMAR*, Sep. 2009, pp. 105–108.

[2] W. Zhu, X. Wang, and W. Gao, "Multimedia Intelligence: When Multimedia Meets Artificial Intelligence," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1823–1835, Jul. 2020, doi: 10.1109/TMM.2020.2969791.

[3] I. Amerini, A. Anagnostopoulos, L. Maiano, and L. R. Celsi, "Deep Learning for Multimedia Forensics," *FNT in Computer Graphics and Vision*, vol. 12, no. 4, pp. 309–457, 2021, doi: 10.1561/0600000096.

[4] V. T. Covello, "Social Media and the Changing Landscape for Risk, High Concern, and Crisis Communication," in *Communicating in Risk, Crisis, and High Stress Situations: Evidence-Based Strategies and Practice*, IEEE, 2022, pp. 385–410. doi: 10.1002/9781119081753.ch13.

[5] T. Mahmood, T. Nawaz, A. Irtaza, R. Ashraf, M. Shah, and M. T. Mahmood, "Copy-Move Forgery Detection Technique for Forensic Analysis in Digital Images," *Mathematical Problems in Engineering*, vol. 2016, p. e8713202, May 2016, doi: 10.1155/2016/8713202.

[6] B. M. Prajapati, N. P. Desai, and E. Dept, "FORENSIC ANALYSIS OF DIGITAL IMAGE TAMPERING," *International Journal For Technological Research In Engineering*, vol. 2, no. 10, p. 5, Jun. 2015.

[7] M. A. Elaskily *et al.*, "A novel deep learning framework for copy-moveforgery detection in images," *Multimed Tools Appl*, vol. 79, no. 27, pp. 19167–19192, Jul. 2020, doi: 10.1007/s11042-020-08751-7.

[8] "ICL– Images and Communication Lab. Department of Information Engineering (DINFO) of the University of Florence," Image and Communication Laboratory. Accessed: Jun. 22, 2022. [Online]. Available: http://lci.micc.unifi.it/labd/team/

[9] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, “Geometric tampering estimation by means of a SIFT-based forensic analysis,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2010, pp. 1702–1705. doi: 10.1109/ICASSP.2010.5495485.

[10] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, “A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011, doi: 10.1109/TIFS.2011.2129512.

[11] R. Caldelli, I. Amerini, L. Ballan, G. Serra, M. Barni, and A. Costanzo, “On the effectiveness of local warping against SIFT-based copy-move detection,” in *2012 5th International Symposium on Communications, Control and Signal Processing*, 2012, pp. 1–5. doi: 10.1109/ISCCSP.2012.6217846.

[12] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra, “Copy-move forgery detection and localization by means of robust clustering with J-Linkage,” *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 659–669, Jul. 2013, doi: 10.1016/j.image.2013.03.006.

[13] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, “Counter-forensics of SIFT-based copy-move detection by means of keypoint classification,” *J Image Video Proc*, vol. 2013, no. 1, p. 18, Apr. 2013, doi: 10.1186/1687-5281-2013-18.

[14] M. F. Hashmi, A. Hambarde, V. Anand, and A. Keskar, “Passive Detection of Copy-Move Forgery using Wavelet Transforms and SIFT Features,” *Journal of Information Assurance and Security(JIAS) ISSN 1554-1010*, vol. 9, pp. 197–204, Aug. 2014.

[15] N. B. A. Warif *et al.*, “Copy-move forgery detection: Survey, challenges and future directions,” *Journal of Network and Computer Applications*, vol. 75, pp. 259–278, Nov. 2016, doi: 10.1016/j.jnca.2016.09.008.

[16] S. T. Nabi, M. Kumar, P. Singh, N. Aggarwal, and K. Kumar, “A comprehensive survey of image and video forgery techniques: variants, challenges, and future directions,” *Multimedia Systems*, vol. 28, no. 3, pp. 939–992, Jun. 2022, doi: 10.1007/s00530-021-00873-8.

[17] Y. Zhang, Y. Yan, and G. Feng, “Feature compensation network based on non-uniform quantization of channels for digital image global manipulation forensics,” *Signal Processing: Image Communication*, vol. 107, p. 116795, Sep. 2022, doi: 10.1016/j.image.2022.116795.

[18] C. Lin, Y. Wu, K. Huang, H. Yang, Y. Deng, and Y. Wen, “Copy-move forgery detection using Regional Density Center clustering,” *Journal of Visual Communication and Image Representation*, vol. 103, p. 104221, Aug. 2024, doi: 10.1016/j.jvcir.2024.104221.

[19] A. U. Shehin and D. Sankar, “Copy Move Forgery detection and localisation robust to rotation using block based Discrete Cosine Transform and eigenvalues,” *Journal of Visual Communication and Image Representation*, vol. 99, p. 104075, Mar. 2024, doi: 10.1016/j.jvcir.2024.104075.

[20] J. Wang *et al.*, “Strong robust copy-move forgery detection network based on layer-by-layer decoupling refinement,” *Information Processing & Management*, vol. 61, no. 3, p. 103685, May 2024, doi: 10.1016/j.ipm.2024.103685.

[21] J.-L. Zhong, Y.-F. Gan, and J.-X. Yang, “Efficient detection of intra/inter-frame video copy-move forgery: A hierarchical coarse-to-fine method,” *Journal of Information Security and Applications*, vol. 85, p. 103863, Sep. 2024, doi: 10.1016/j.jisa.2024.103863.

[22] W. Ye, Q. Zeng, Y. Peng, Y. Liu, and C.-C. Chang, “A two-stage detection method of copy-move forgery based on parallel feature fusion,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, p. 30, Apr. 2022, doi: 10.1186/s13638-022-02112-8.

[23] I. Amerini, R. Caldelli, A. D. Bimbo, A. D. Fuccia, L. Saravo, and A. P. Rizzo, “Copy-move forgery detection from printed images,” in *Media Watermarking, Security, and Forensics 2014*, SPIE, Feb. 2014, pp. 336–345. doi: 10.1117/12.2039509.

[24] R. Caldelli, I. Amerini, and A. Costanzo, “SIFT match removal and keypoint preservation through dominant orientation shift,” in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2062–2066. doi: 10.1109/EUSIPCO.2015.7362747.

[25] N. B. A. Warif, Mohd. Y. I. Idris, A. W. A. Wahab, N.-S. N. Ismail, and R. Salleh, “A comprehensive evaluation procedure for copy-move forgery detection methods: results from a systematic review,” *Multimed Tools Appl*, vol. 81, no. 11, pp. 15171–15203, May 2022, doi: 10.1007/s11042-022-12010-2.

[26] M. M. A. Alhaidery, A. H. Taherinia, and H. I. Shahadi, “A robust detection and localization technique for copy-move forgery in digital images,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 1, pp. 449–461, Jan. 2023, doi: 10.1016/j.jksuci.2022.12.014.

[27] M. A. Elaskily, H. K. Aslan, O. A. Elshakankiry, O. S. Faragallah, F. E. A. El-Samie, and M. M. Dessouky, “Comparative study of copy-move forgery detection techniques,” in *2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)*, 2017, pp. 193–203. doi: 10.1109/ACCS-PEIT.2017.8303041.

[28] A. C.-C. Liu and O. M. K. Law, “Convolution Optimization,” in *Artificial Intelligence Hardware Design: Challenges and Solutions*, IEEE, 2021, pp. 85–126. doi: 10.1002/9781119810483.ch5.

[29] “Deep Learning,” in *Nonlinear Filters*, John Wiley & Sons, Ltd, 2022, pp. 113–140. doi: 10.1002/9781119078166.ch8.

[30] J. I. Agbinya, “11 Convolutional Neural Networks,” in *Applied Data Analytics – Principles and Applications*, River Publishers, 2019, pp. 185–204. Accessed: Jun. 25, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9227126

[31] “Intuition of Adam Optimizer,” GeeksforGeeks. Accessed: Jun. 25, 2022. [Online]. Available: https://www.geeksforgeeks.org/intuition-of-adam-optimizer/

[32] J. Brownlee, *Probability for Machine Learning: Discover How To Harness Uncertainty With Python*. Machine Learning Mastery, 2019.

[33] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2020, doi: 10.1016/j.aci.2018.08.003.

[34] J. Keshet and S. Bengio, "From Binary Classification to Categorial Prediction," in *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, Wiley, 2009, pp. 27–49. doi: 10.1002/9780470742044.ch3.

[35] J. Opitz and S. Burst, "Macro F1 and Macro F1," Feb. 08, 2021, *arXiv*: arXiv:1911.03347. doi: 10.48550/arXiv.1911.03347.

[36] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, Oct. 1997, doi: 10.1016/S0034-4257(97)00083-7.

[37] O. Kuznetsov, E. Frontoni, L. Romeo, and R. Rosati, "Enhancing copy-move forgery detection through a novel CNN architecture and comprehensive dataset analysis," *Multimed Tools Appl*, Jan. 2024, doi: 10.1007/s11042-023-17964-5.