

Review

Not peer-reviewed version

Image Inpainting forgery detection: a review

[Adrian-Alin Barglazan](#)^{*}, [Remus Brad](#)^{*}, [Constantin Constantinescu](#)^{*}

Posted Date: 13 October 2023

doi: 10.20944/preprints202310.0870.v1

Keywords: image inpainting object removal detection forensic forgery



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Review

Image Inpainting Forgery Detection: A Review

Adrian-Alin Barglazan ^{1,*}, Remus Brad ^{1,*} and Constantin Constantinescu ^{1,*}

¹ Faculty of Engineering, Computer Science, "Lucian Blaga" University of Sibiu, 550024 Sibiu, Romania

* Correspondence: adrian.barglazan@ulbsibiu.ro, remus.brad@ulbsibiu.ro, ctin.constantinescu@ulbsibiu.ro .

Abstract: In recent years, significant advancements in the field of machine learning have influenced the domain of image restoration. While these technological advancements present prospects for improving the quality of images, they also present difficulties, particularly the proliferation of manipulated or counterfeit multimedia information on the internet. The objective of this paper is to provide a comprehensive review of existing inpainting algorithms and forgery detections, with a specific emphasis on techniques that are designed for the purpose of removing objects from digital images. In this study, we will examine various techniques encompassing conventional texture synthesis methods, as well as those based on neural networks. Furthermore, we will explore the artifacts associated with the identification of modified photos and present the artifacts frequently introduced by the inpainting procedure and assess the state-of-the-art technology for detecting such modifications. Lastly, we shall look at the available datasets and how the methods compare with each other. Having covered all of the above, the final outcome of this study is to provide a comprehensive perspective on the abilities and constraints to detect images for which an inpainting object removal method was applied.

Keywords: image inpainting object removal detection forensic forgery

1. Introduction

With the improvements and innovations in the last few years in the fields of image inpainting techniques and machine learning, an increasing number of altered and fake media content has invaded the internet. In the current paper we will do a thorough review of the current inpainting mechanism and ascertain current state-of-the-art techniques to detect these alterations.

Nowadays, in our modern society, we rely greatly on technology. This can be seen as an advantage but also as a drawback. The evolution of technology has impacted the lives of each one of us. We are just a click away from an almost infinite amount of information that can be accessed at any time. Most of the time people rely almost completely on the information that they find online and form their opinions based on those facts, but unfortunately, this is not always a safe approach [5]. The authenticity of the information that can be found online can sometimes be distorted or even false. That is the reason why its accuracy needs to be always checked. We tend to believe that false information can be transmitted only through textual content, but this is not entirely the case. Nowadays images and videos are also a tool for transmitting information. We use them daily, and we became accustomed to believing everything we see to be the truth. The most powerful example, in this case, can be the images and videos that we see and upload on social networks. This example shows that it is equally important to check the authenticity of images as it is to check the trustworthiness of a written text. All the reasons stated above imply the fact that there is a great need to detect forgeries in images and videos [7].

The science area focusing on tampered image and video detection is called media forgery detection. The area is quite vast and has an increasing interest as described in recent bibliometric studies done by [1], [2], [3] [Figure 1 Number of forgery detections per paper in recent years]. The forgery detection methods can be divided into two categories: active and passive. For the active methods, the main focus is embedding some metadata onto the images at the time of creation and it can be later used to validate the authenticity of the image. On the other hand, passive methods which

are sometimes called also blind methods, do not offer so much specific information, thus one has to rely entirely on the possible artifacts introduced in the tampering process.

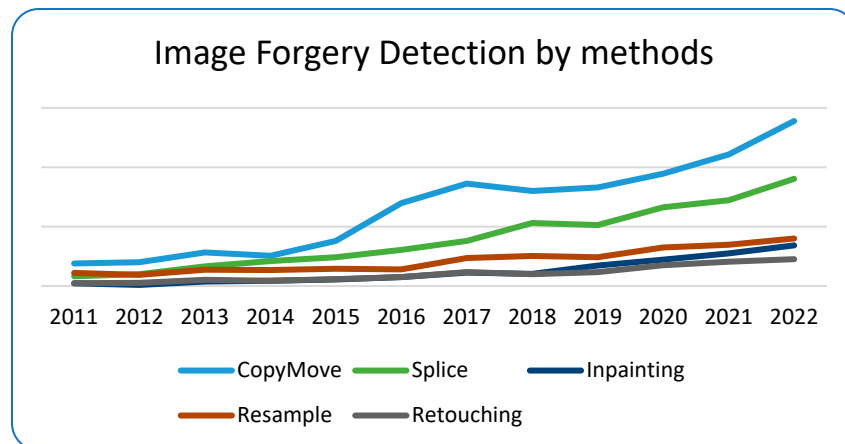


Figure 1. Trends in forgery detection during the last years.

If one would look at the tampering process according to the father of digital image forensics, Hany Farid) as mentioned in [4], the forgery detection can be performed through:

- Active methods: briefly, the main idea here is to incorporate various information that can be validated later on, in the moment of image acquisition.
- Passive methods: here the area is quite big. Some of these methods focus on the peculiarities of image capturing, camera identification, noise detection, image inconsistencies, or on some specific type of traces which are usually introduced by the forgery mechanism – for e.g. for a copy-paste forgery (combining information multiple images) – some traces like inconsistent coloring, noising, blur, etc., might be noticed.

A more detailed schema based on the above-mentioned categories can be seen in Figure 2 on the comprehensive work done by Pawel Korus in his research paper [5]. Based on our analysis, we structured a bit different the categorization. The first difference is related to camera traces, where we are grouping all steps / artifacts that might influence the outcome of the resulted image. We considered this categorization important because camera traces can be used to determine the forged area. As it can be noticed in the later chapters, authors have tried initially to focus only on one type of artifact, however, recent studies suggest that the best approach would be an ensembled method. Compared to the initial categorization done by Korus in his review, in this case, the image Copy-Move is further sub divided into several sub-categories. Let us consider that all types of image operations which are done on a source image, without blending information from other images, would fall under this category. Therefore, the new Copy-Move category contains items like Object Removal, Resample, Blurring, Gamma correction, etc. This is very important, because basically the Copy-Move forgery is redefined as an operation done on image, solely based on the statistical data of the image itself; for e.g. when we inpaint an image, we fill in data based on the overall statistical analysis of that particular image.

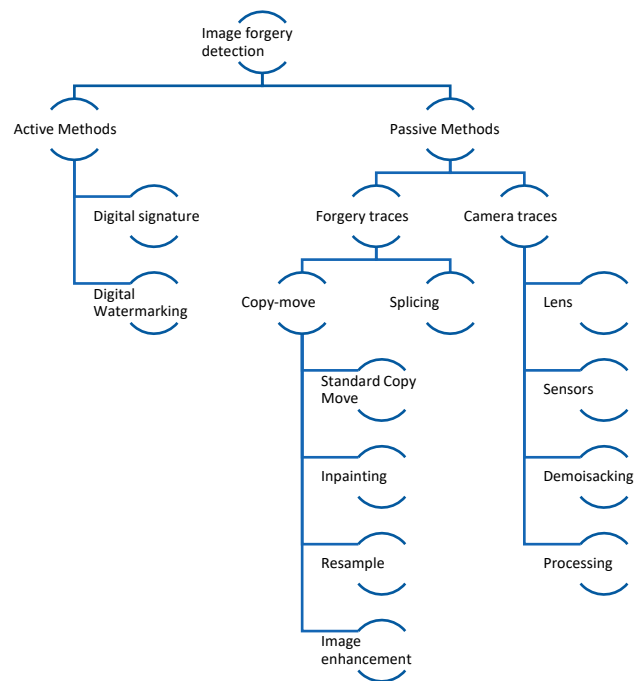


Figure 2. Image forgery detection overview.

Looking from the “attacking” point of view, for the passive methods, we can end up with another classification from the traces that forgery methods might introduce:

- Copy-paste methods: in this case the picture is altered by copying parts of the original image into the same image. Of course, things like re-sample & rescaling can be included; usually both resampling and rescaling at their core are not methods of altering but are rather used as a step to apply copy-paste or splicing methods.
- Splicing: the forged media is obtained by combining several images into one; e.g., taking the picture of someone and adding it within another one;

The current paper conducts a deep evaluation of the current methods for detecting inpainting/object removal inside images and videos. The material is divided into the following parts: the first part gives a full review of the current state-of-the-art methods in inpainting methods, with a deep focus on object removal. Going further, we plan to analyze all the pros and cons of each method, review the dataset, see the way they behave in real-world scenarios and how they compare to others in terms of quality. After this thorough review, we will shift our focus to describe the detection of inpainting problems as a general matter. After that, we shall analyze the forgery detection methods. We will start with some older variants, their main ideas, and assess their performance. We will continue the journey by analysing also other relevant forgery detections mechanisms and investigate if they can apply successfully to the object removal tasks. For each relevant method we will do a comprehensive analysis on the pro and cons and analyze also how they work outside the tested datasets. Furthermore, additional analysis will be performed on available datasets used for evaluation of these methods and the issues they raise. Lastly, all the relevant findings are briefly summarized and the relevant areas of improvement are also diligently addressed.

2. Inpainting methods

Image inpainting is sometimes called an inverse problem and usually these types of problems are ill-posed. In the process of inpainting, especially in large areas, all three conditions specified by Hadamard are infringed – thus the problem is a so-called inversed problem). Generally, the problem of inpainting consists in finding the best approximation to fill in the region inside the source image and comparing it with the ground truth. All the algorithms which tackled this problem begin with the assumption that there must be some correlation between the pixels presented inside the image,

either from a statistical or from a geometrical perspective. This work differs from the one in [6], by its deep focus on inpainting methods exclusively.

Once the mathematical concepts were formulated, the possible solutions which have been proposed can be identified. An image can be defined as a collection of points, each point having a set of values belonging to that point. Starting with a mathematical approach, first the image model is defined and then the inpainting problem will follow.

$$I: x \rightarrow I(x) - \text{image consists of an array of pixels where } x: R^n \rightarrow R^m$$

- each pixel is defined as follows: n represents the coordinate system-usually 2
- m represents the color space representation of that pixel (usually 3 for an RGB space)

The inpainting problem consists of the following:

D – degradation operator applied on initial image I

$$D(I) = K \cup U - \text{we called } U \text{ the unknown part and } K \text{ the known part}$$

$$R(I) = K \cup r(U)$$

R will be denoted the reconstructed image and $r(U, K)$ the reconstruction of U area based on the K . The aim at image inpainting is to reconstruct the U area as best by comparing with the original image I , in other words, to minimize the differences between the original image I and the reconstructed image called R . Thus, having formulated the mathematical concepts for image inpainting, we shall use as starting point the previous reviews done in [7], [8] and most recently in [9]. All the above authors categorize the inpainting methods as follows:

- Diffusion based or sometimes called Partial Differential Equations based
- Exemplar based or patch based as it refers in some other papers
- Machine Learning based, usually we shall address all machine learning algorithms inside this category although [9] splits the machine learning based into several categories based on the model construction

2.1. Diffusion based methods

The term diffusion (from a chemistry point of view) is the action in which items inside a region of higher concentration tend to move to a lower concentration area. From a mathematical point of view - let's define it as: let $\Omega \subset R^2$ denotes the entire image domain f . The basic idea then is to propagate information from the border of the missing region into it, in such a way that the border of the missing region is no longer visible to the human eye. The border of missing region is going to be called ∂D ; Figure below illustrates the inpainting steps.

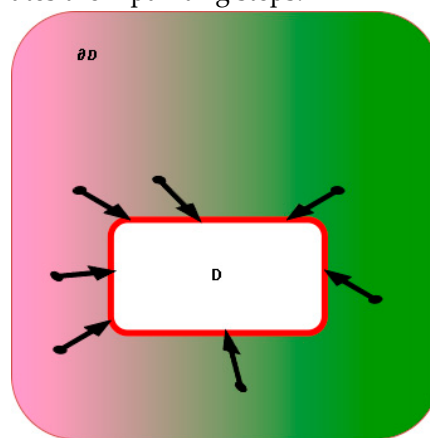


Figure 3. Process of inpainting-based on PDE method.

Several authors [10], [9] have suggested a more detailed approach of splitting the inpainting diffusion class. They have suggested to further divide into the following sub-categories like: isotropic, anisotropic, total variation, PDE based. For the simplicity of this paper, we intended to organize all

these methods under one big umbrella, because the starting points are the main ideas observed by [11], in which the inpainting process is inspired from the “real” inpainting of canvas and consists of the following steps:

- Global image properties enforce how to fill in the missing area
- The layout of the area δD is continued into D (all edges are preserved)
- The area D is split into regions and each region is filled with the color matching δD (the color information is preserved from the bounding area δD into the rest of the D area)
- Texture is added

The first step in almost all the inpainting algorithms is to apply some sort of regularization. It can be either isotropic, with some rather poor results, anisotropic, or any other type of regularization. This is done in order to ensure that image noise is removed, and thus it shall not interfere in the computation of the structural data needed in the next step.

In order to apply diffusion, the structural and statistical data of the low level image must be identified. Based on this data, if on an edge on the δD area, we must conserve the edge identified and if δD area belongs to a consistent area, we can then easily replicate the same pixel information from the border. In order to retrieve image geometry, one can use isophotes - curve on surface connecting points of same values. For this one needs to compute first the gradient on each point on the margin area and then to compute the direction as a normal one to the discretized gradient vector.

Having performed these steps, the initial algorithm from [11] is just a succession of anisotropic filtering, followed by inpainting and then this repeated several times. From a mathematical point of view, based on [11], the intention is to achieve the following:

$$\begin{aligned}
 U(x) &= \text{unknown area} - \text{an array of pixels where } x: R^n \rightarrow R^m \\
 U_k(x) &= \text{is the } k \text{ variant obtained.} \\
 U_k(x) &= U_{k-1}(x) + \alpha * \delta L^k(x) * N^k(x) \\
 L^k(x) &= \text{smoothness estimator [11]} \\
 N^k(x) &= \text{isophotes directions}
 \end{aligned}$$

In the original implementations the authors made some assumptions choosing α to be 0.1 and as for the smoothness estimator they have used the Laplacian. An interesting part is the choice of the N vector. In their original paper it is suggested that this vector has to be computed each time and is based on the current rotated gradient of the current block size to be inpainted. The problem with this vector is that it has to be computed at each iteration because with each iteration new information arises at the area to be reconstructed. Additionally, they also add an anisotropic diffusion step at each several steps, intending not to lose too much sharpness. From a forensic point of view, this is a very important step, because it does not tend to keep the same level of blur between the original area and the reconstructed area. Later on, the authors in [12] proposed an improved version of their initial algorithm. The idea was inspired from the mathematical equations of fluid dynamics, specifically the Navier-Stokes equations, which describe the motion of fluid. The proposal was to use the continuity and momentum equations of fluid dynamics to propagate information from known areas of the image or video towards the missing or corrupted areas. This was more or less an improved version of higher PDE version presented initially. As a follow up of his original work, Bertalmio proposed in [13] the use of 3rd order PDE, which are a better continuation of edges.

The algorithm starts by defining an initial velocity field that guides the propagation of information. This velocity field is then iteratively updated with the known image used as boundary conditions by the use of the Navier-Stokes equations. The result is obtained by advection of the original image along the final velocity field. The algorithm seems to perform slightly better than the initial paper in situations where the missing data is large, or the structure of the image is complex. The Navier-Stokes inpainting algorithm is a method for completing missing or corrupted parts of images or videos that uses the equations of fluid dynamics to propagate information from known areas to missing or corrupted areas. The final result is obtained by advecting the original image or video along the final velocity field.

At the same time, Chan & Shen developed similar algorithms [14], [15] in which they postulated the use of the local curvature of an image to guide the reconstruction of missing or obscured parts.

Using Euler's Elastica model, they can predict for what the missing parts of the image might look like. Both Euler's Elastica and PDE-based inpainting are effective methods for image inpainting and have their own advantages and disadvantages. Euler's Elastica is particularly well-suited for images that contain thin, flexible objects, while PDE-based inpainting is well-suited for images that are smooth and locally consistent. Depending on the specific characteristics of the image and the desired outcome, one method may be more appropriate than the other.

Based on the work described above, many methods continue in the same direction, trying to map real physical processes into the inpainting process (diffusion, fluid dynamics, osmosis). For e.g. in [16] the authors proposed curvature-preserving PDE. Their tensor PDE, is used for regularizing images, keeping into account the curvatures of specific integral curves. In this way, they estimate better the shape of the inpainting data, thus reducing the blurring effect on the resulted image. Another variant with very good results and implemented in the computer vision library (OpenCv) is presented in the paper [17]. The authors present a fast marching technique that estimates the missing pixels in one pass using weighted means of known calculated pixels. This is suboptimal algorithm compared to other inpainting algorithms, but it gains strength in its speediness compared to the example of Bertalmio, in which several iterations were needed and the result was affected by the number of iterations.

In the recent year the focus for diffusion based inpainting has moved towards more and more complex PDE forms. For e.g. in [18] using high order variational models is suggested, like low curvature image simplifiers or Cahn-Hilliard equation. Another recent paper that goes into the same direction is [19], which basically integrates the geometric features of image, namely the Gauss curvature. Still, even these methods introduce the blurring artifact also found in initial papers [11], [12]. In order to surpass these challenges in the current models, with second-order diffusion-based models that are prone to staircase effects and connectivity issues and fourth-order models that tend to exhibit speckle artifacts, a newer set of models has to be developed. The authors Sridevi & Srinivas Kumar proposed several robust image inpainting models that employ fractional-order nonlinear diffusion, steered by difference curvature in papers [20], [21], [22]. In their most recent paper [23], a fractional-order variational model is added to mitigate noise and blur effectively. In essence, a variation of DFT is used to consider pixel values from the whole image, not only by relying strictly on the neighboring pixels.

In an attempt to summarize the diffusion inpainting methods, it is found that they usually rely on 2nd or higher order partial derivatives, or via Total variation of energy, in order to be able to "guess" the missing area. One of the major drawbacks of these methods is that in some way, either locally or globally, some sort of anisotropic diffusion is introduced with a blurring effect, which in turn will affect the entire image. Due to this blurring effect nature, in theory image inpainting via PDE can be detected via some sort of inconsistency in the blurring effect of various regions.

2.2. Exemplar based methods

Approximately at the same a newer approach based on texture synthesis started to gain more momentum. The main inspiration came from [24] where A. A. Efros and T. K. Leung introduced a non-parametric method for texture synthesis, where the algorithm generates new texture images by sampling and matching pixels from a given input texture, based on their neighborhood pixels, thus effectively synthesizing textures that closely resemble the input sample. This approach was notable for its simplicity and ability to produce high-quality results, making it a foundational work in the area of texture synthesis. . The primary goal of these approaches was to enhance the reconstruction of the image section area which is missing. However, the challenges brought by texture synthesis are slightly different from those presented by classical image inpainting. The fundamental objective of texture synthesis is to generate a larger texture that closely resembles a given sample in terms of visual appearance. This challenge is also commonly referred to as sample-based texture synthesis. A considerable amount of research has been conducted in the field of texture synthesis, employing strategies such as local region growing or holistic optimization. Probably one of the main papers that gained a lot of attention was the work of [25]. In this paper, Criminisi presented a novel algorithm

for the removal of large objects from digital images. The technique is known as Exemplar-Based Image Inpainting. The method is based on the idea of priority computation for the fill front, and best exemplar selection for texture synthesis. Given a target region Ω to be inpainted, the algorithm determines the fill order based on the priority function $P(p)$, defined for each pixel p on the fill front $\partial\Omega$. $P(p) = C(p) * D(p)$ Where, $C(p)$ is the confidence term, an indication of the amount of reliable information around pixel p . $D(p)$ is the data term, a measure of the strength of isophotes hitting the front at p . The algorithm proceeds in a greedy manner, filling in the region of highest priority first with the best match from the source region Φ . This is identified using the Sum of Squared Differences (SSD) between patches. The novel aspect of the method is that it combines the structure propagation and texture synthesis into one framework, aiming to preserve the structure of the image, while simultaneously considering the texture. It's been demonstrated to outperform traditional texture synthesis methods in many complex scenes and it has been influential in the field of image processing.

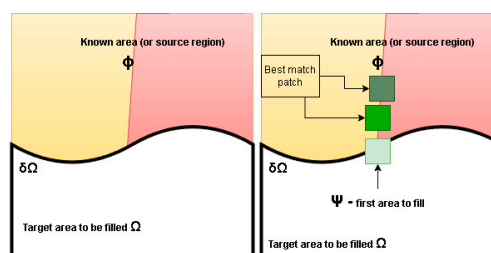


Figure 4. Criminisi algorithm [25].

Based on these seminal works of Leung and Criminisi, the area started to be more and more researched and various places were investigated for further improvements: order of patch processing, faster way of computing the best patch, applying various operations on the best patch found in order not to disturb higher statistical data inside the image, multiscale and overcome global constraints, and even finding the best way of dealing with distances between patches.

Several methods are available for calculating the resemblance between patches of images. The most commonly employed metrics can be grouped into two categories: pixel-based metrics, with gauge similarity based on the difference or cross-correlation among pixel color values, and statistics-based metrics, which estimate the similarity between the probability distributions of pixel color values in patches. The first category includes metrics such as the sum of squared differences (SSD), the L_p norm, and normalized cross-correlation. The second category features statistics-based metrics like the Bhattacharyya distance, normalized mutual information (NMI), and Kullback-Leibler divergence. The SSD is the most frequently used metric when searching for similar patches. One key aspect in the designing of an inpainting forensic tool is that SSD tends to favor uniform regions, meaning it prefers copying pixels from those areas. To address this bias, a weighted Bhattacharyya distance, denoted as $d(SSD, BC)$, has been proposed. Nonetheless, when two patches have identical distributions, their Bhattacharyya distance (d_{BC}) is zero, implying that the weighted Bhattacharyya distance is also zero, even if one patch is a geometrically modified version of the other.

Another area of improvement was one how fast/optimal the best patch is found. For this reason, some methods start by identifying the K-nearest neighbors (K-NNs) within the recognizable sections of the image. One simple approach to solving the nearest neighbor search problem is by calculating the distance from the target patch to all potential patches, regarding each patch as a point in multi-dimensional space. More efficient and approximate nearest neighbor search strategies are available, which structure the potential candidates using certain space-segmenting data structures like k-dimensional trees (kd-trees) or vantage point trees (vp-trees), guided by their spread in the search space. The nearest neighbor search can be performed effectively by using the properties of these trees to quickly discard vast sections of the search space, leaving only a minor portion of candidates for verification. The matching process based on kd-trees is one of the most used methods for identifying the nearest patch. However, the number of nodes examined expands exponentially with the

dimension of the space. As a result, when the dimension is big, the search speed slows. A variety of nearest neighbor search algorithms are evaluated in a separate study [26] to determine their effectiveness in locating similar patches within images. A big improvement in this area was incorporated into Photoshop tool in recent years. The idea is based on the Patch Match algorithm [27]. Approximate nearest neighbor (ANN) search methods that are tree-based treat each query individually. PatchMatch, a randomized patch search algorithm introduced, takes advantage of the relationships between queries to facilitate collaborative searching. This method operates on the presumption that images maintain coherency. That is to say, once a similar pair of patches in two images is identified, their adjacent patches (those offset by a few pixels) are likely also similar. Consequently, the match result of a specific patch can be transferred to proximate queries, providing an advantageous initial guess that can then be updated with randomly selected candidates. PatchMatch is a speedy algorithm used for calculating dense, approximate nearest neighbor correspondences between patches in two image areas, with these correspondences collectively referred to as the nearest neighbor field (NNF). The algorithm initiates the search for the NNF as follows. The NNF is initially assigned either random values or prior information, with random guesses likely offering only a few beneficial guesses. The NNF is then continually fine-tuned by alternating between two operations known as propagation and random search, carried out on the patch level. The propagation step updates a patch offset using known offsets from its causal neighborhood, leveraging image coherency. During even iterations, offsets are propagated from the top and left patches, while during odd iterations, they are propagated from the right and bottom patches. The second operation carries out a local random search to establish initial patch matches, which are then disseminated by iterating a limited number of times. Although the algorithm is significantly faster than kd-trees, it offers less accuracy. It can get stuck in a local optimum due to the limited distance of propagation.

In the recent years the methods have become more and more complex and tried to exploit various artifacts inside the image and analyzing more in depth the structure near the area to be inpainted. Other approaches like [28], utilize a patch-based approach that searches for well-matched patches in the texture component using a Markov random field (MRF). Jin and Ye [29] proposed an alternative patch-based method that incorporates an annihilation property filter and a low rank structured matrix. Their approach aims to remove an object from an image by selecting the target object and restricting the search process to the surrounding background. Additionally, Kawai [30] presented an approach for object removal in images by employing a target object selection technique and confining the search area to the background. Authors have also explored patch-based methods for recovering corrupted blocks in images using two-stage low rank approximation [31] and gradient-based low rank approximation [32]. Another sub-area of focus for some authors was to represent the information by “translating” first the image into another format, so called sparse representation like DCT, DFT, DWT etc.. Here we just want to mention a few interesting research papers [33], [34]. They obtained good quality, while the area to be inpainting is rather uniform, but if the area is at edge of various different texture, the methods introduce some pretty ugly artifacts that make the methods unusable.

Some various authors like [7][8]–[10], [35], [36] have suggested that another classification of the inpainting procedure can be done. Either they suggest to add a sub-division based on sparse representation of images (like authors have suggest in [33], [37]–[39]) and then later try to apply existing algorithms on this representation, or a so called mixed / fusion mode – in which authors try to incorporate ideas from both world: from diffusion based and from texture synthesis (patch copying). In this latter category we can name a few interesting ideas, like the one Bertalmio explored in his [40] study, in which they combined PDE based solution together with patch synthesis and coherence map. The resulting energy function is a combination of the 3 metrics. As similar idea to Bertalmio’s above mentioned work, is the research of Aujol J, Ldjal S and Masnou S in their [41] article, in which they try to use exemplar based methods to reconstruct local features like edges. Another inquest in the same idea was the work of Wallace Casaca, Maurilio Boaventura , Marcos Proença de Almeida and Luis Gustavo Nonato on [42] in which they combine anisotropic diffusion

with transport equation to produce better results. Their suggested approach of using a cartoon-driven filling sequence has proven to be highly effective for image inpainting using as metrics both PSNR and speed.

If we were to summarize the exemplar (or patch based) inpainting methods, they try to do 3 steps:

- find the best order to fill the missing area
- find the best patch that approximates that area
- try to apply if needed some processing on the copied patch in order to ensure that both local and global characteristics are maintained

Now if we look at the artifacts these methods introduce, we can easily categorize them into two groups: methods that simply “copy-paste” unconnected / unrelated regions (patches) into the missing area and methods that do some enhancement / adaptation of the patch values. The first category is straightforward to determine via a forensic algorithm: we rely solely on the fact that for a given region (usually several times greater than the patch used for inpainting) is comprised of patches that have “references” in other areas. The main problems are that how to determine the correct patch size to be able to determine “copied”, speed, and last but not least – how to correctly eliminate false positives (especially when the patch size is smaller and the window step is small as well). The second category of inpainting patch based methods – that do not simply copy the source patch to destination, is a little harder to detect. The above algorithm, where we search for similar (identical) patches can no longer be applied, we must introduce as well some heuristic, and probably evaluate some parameters on how much the patches resemble. Last, the problem of false positive with these approaches, increases exponentially because we are no longer finding identical patches, but rather nearly identical patches, and on images with smooth large texture, we might end up with a lot of false positives data.

2.3. Machine learning based methods

Starting with approximately 2013, Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) have emerged as state-of-the-art methods for image inpainting. Usually, these methods are employed as feature extraction tools through convolution, enabling the capture of abstract representations. The combination of CNNs with adversarial training, as proposed by Goodfellow in his 2014 paper, has generated some impressive results in inpainting tasks, achieving perceptual similarity to the original image. The integration of CNNs with GANs has proven advantageous, as CNNs provide an encoder for high-dimensional abstraction extraction, while GANs enhance image sharpness and color synthesis. In the area of machine learning based, the main method, which is attributed to Deepak Pathak in his paper [43], suggested the use of a system composed of an encoder and a decoder. The encoder will focus on retaining data information (extracting it), and the decoder responsibility will be to generate features based on the encoders learned data. Starting with this approach, several methods have been suggested, like FCN initially, where we have two neural networks with skip connections between them. An improved version of this FCN version, was the U-Net architecture, which resembles FCN, but uses summation as a skip connection mechanism, while in U-NET the concatenation is employed. The advantage of using concatenation is that it can retain more detailed data. One of the existing problem on the inpainting problem was on how to generate the missing area in highly texturized area. To address this challenge, some authors proposed different mechanisms to exploit both global and local texture information. Another point of variation between inpainting methods was the convolutions used. Here we would like to focus more, because from the perspective of detection, the convolution applied at the decoder level (deconvolution or how some authors call it - transposed convolutional layer), is the one responsible for introducing various type of artifacts. From the analysis the most used convolutions are: simple (or standard) convolution - it is good for reconstruction especially rectangular based shapes and gated convolution - the main idea is to be able to fill in irregular holes, by using a mask which is updated after each convolution. For a detailed review on the machine learning based technique, we recommend [9] or [44] which does a very good overview of the current state of the art methods. Another very recent and good overview on machine learning inpainting methods is the

research done in [38]. Below we present an updated version (only new additions) to the summary presented in the above papers, with focus on the improved versions compared with the above review papers, and by analyzing also the possible artifacts that are introduced by them. The first category of research with relevant good results are [45], [46], [47]. Mainly the methods rely on Fourier convolutions and perceptual loss. Their results are pretty impressive on the CelebA ([48]) and places ([49]) dataset. An improvement on the Lama model was presented by the authors in [50] on the learned perceptual image patch similarity (LPIPS) metric. Their method starts from a noisy image and applies denoising filling in the missing data based on the known area. In [51] the authors have suggested another approach, they apply already established classical method of inpainting from OpenCV (Telea and NS methods) and use a CNN model to learn the reconstruction of these features. As a backbone, they use a VGG16 model and as features, the authors used three distinct traits: image size, RGB information and brightness levels. The results are straightforward when the area to be inpainted is uniform (for e.g. the middle of the sky), but when the region to be generated is at the border of several highly texturized areas, the methods does not yield proper results. Recently in [52] they suggest that the task of inpainting should be divided into two separate stages. In the first stage, they use two separate DRGAN modules – one to generate roughly the content of the inpainted area, and one to roughly generate the edges of the missing area – actually they generate a label image where 1 is the edge and 0 represents the background. This information is crucial in the second stage of the algorithm, where the authors used a fine-grained network to generate more coarse pixel information based on the label edges and the roughly already generated data. Again they use a DRGAN, deep residual generative adversarial network, architecture for this part of the method. Analyzing the results and the comparison with some state of the art methods, it seems the proposed method is able to reconstruct highly texturized areas, but has some limitations in – what the authors call “overly complex” regions. Authors in [53] have diverged from the mainstream usage of transformers and have incorporated a discrete wavelet transformer along the convolutional layers. Still for upsampling the authors use the standard transpose convolution which generates checkboard artifacts. Another approach with very good results is the work [54] in which the authors combine autoencoders and transformers on a Resnet architecture. The idea of using transformers, is that they are able to better represent details and thus able to reconstruct the missing area, but still the authors use the same type of architecture (resnet) which employs the same type of upsampler.

From a detection point of view, these methods are becoming more and more challenging due to their ability to propagate patches that are indistinguishable from the rest of image. Also, due to their nature to complete large areas, they are able to reconstruct the entire image characteristics. As attack vector various methods have been proposed, but mainly they are focusing on the artifacts introduced by the various up sampling steps.

3. Inpainting forgery detection mechanism

In the following sub-chapter, we plan to present image and video forgery detection mechanisms, with a deep focus on detecting only forgeries generated via an inpainting process (either manually or automatically). Regarding the overall forgery detection mechanism, there are existing reviews in this area, just to mention a few which are more relevant and have gone into very great detail: [3] [5], [55], [56], [57]. Their focus is mainly on generical forgery determination methods, while in the following paragraphs, we intent to focus closer to the proposed methods for inpainting determination. The following chapter will be organized as follows: we shall present state-of-the-art inpainting detections using – we call them classical methods, and then later we focus on deep learning methods. This approach will be both on images and videos.

We will start with a small review of diffusion-based inpainting detection. Due to the nature of these methods, which usually fill a small area, there are very few attempts to try to identify images that went through diffusion-based inpainting. In [58] authors realized that in diffusion-based inpainting, the Laplacian is acting differently on untouched vs touched regions. Based on this observation, they propose to use two discriminating intra and inter-channel variances. Of course, they need to add some extra post-processing to eliminate false positives, especially on images that

went under or over-exposure. In the context of Machine Learning-based there are only two papers/examples/attempts that we are aware of [59], [60]. One of the papers presents a feature pyramid network for the detection of diffusion-based inpainting, while the other suggests using the same observations, that the Laplacians are usually affected by inpainting, thus a feature extractor + ensemble classifier should be able to detect those regions. Another recent research [61] suggests using Local Binary Pattern mechanism to detect only diffusion-based inpainting

Before going deep into inpainting forgery detection, let's revise the copy-move forgery detection mechanisms. In the 2012 article [62], the authors presented a framework for determining copy-move blocks from images (CMFD). Their focus was on reliability and exploring various mechanisms for determining duplicate areas. Their proposed algorithm consists of the following main steps:

- Feature extraction (either via block-based or using some variants of key points detection like SURF/SIFT). From their analysis, it seems that Zernike moments feature extraction gives the best overall results. Also the algorithm is less influenced when the copied area is either shrunk and/or rotated. Additionally, the algorithm (feature extraction mechanism), seems to work on various attacking vectors, like resizing, jpeg compression, blurring, etc.
- Matching – here they've suggested a variety of methods – kNN, brute force, etc. Based on the analysis of the authors in [62], kNN gave the best results
- Filtering – to ensure the “estimated” blocks do not violate some other constraints (like the distance between them etc.)

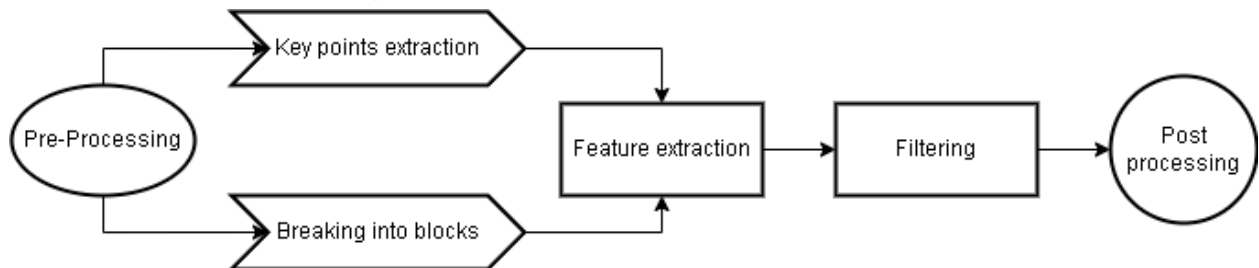


Figure 5. General steps to copy paste forgery detection as presented in P. Korus paper.

3.1. Image Inpainting detection

If we were to analyze the above-proposed framework, we can see that it works well on the copy-move scenario because it assumes that the copied area is at least bigger than the copy-paste algorithm patch size used in detection. But due to their nature, the inpainting algorithms do not copy a fixed size block, but rather, at each step they try to decide what is the best patch to fill in the missing information. The following will introduce several elements that form the above framework, and other variants that were added to it, unusable for these types of forgeries. To summarize the various inpainting techniques, the following was observed:

- Diffusion-based/PDE/Variational based – they are not suitable for filling large areas, and usually they do not copy patches, but rather propagate smaller information into the area to be reconstructed, thus they do not copy patches, but rather fill the area with information diffused from the known region. So applying a block-based detection will yield no results, as there are no copy regions, but rather diffused areas. Still, some traces can be analyzed, but they are more in the area of inconsistencies in blurring, CFA, and other camera/lens properties.
- Patch-based – at first they seem rather well suited to the above-mentioned framework. They work rather well if the region forged contains a lot of texture, but fail in case the object removed was rather large or surrounded by a continuous area (like sky, water, grass, etc). But at a closer look, the method may give unreliable results, due to the inpainting procedures: usually, patch-based methods reconsider the filling area at each step, and the currently selected patch may vary in location from the previously selected patch. Thus, if for the forgery method, we select a larger area that contains several inpainting patches we will not be able to properly determine a similar area. On the other hand, if for the forgery method, we select a smaller patch, we might need to confront 2 aspects: one will be the speed of the method and the other will be the necessity to add some other mechanism to remove false positives.

3.1.1. Classical Image forensic techniques

Considering the above constraints, various authors proposed some areas to circumvent these problems. One of the first proposed methods to detect object removal was [63] – Detection of Exemplar Based Inpainting - DEBI. The authors proposed a zero connectivity algorithm and added a fuzzy logic to exclude areas likely to be false positives. They make the following observations from the Criminisi paper [25]: because of the way the object removal algorithm works, it will end up introducing some patches which are going to be disconnected. Their algorithm proposes to scan the entire image (or rather a region of interest), compare patches and if they are found similar, pass them through a fuzzy logic semi-trapezoidal membership function. When comparing patches to determine if they are similar – they proposed the following:

- Compute all patches from the image (or from ROI). They are called the suspicious patches
- For each patch in the suspicious patches apply the following algorithm:
 - Compute all the other image patches and compare each one of them to the suspicious patch:
 - Create a difference between the two patches
 - Binarize the difference matrix
 - Find the longest connectivity (either 4way or 8way) inside the binarized matrix
 - Compare the obtained value with the maximum longest connectivity obtained for the suspicious patch
- In the end, apply fuzzy logic to exclude some of the false positive cases.

The algorithm works well for some test scenarios, but it has some serious drawbacks, especially if the targeted image is not altered via the Criminisi inpainting algorithm, but rather with a newer variant of patch-based (or ML based). First of all, the computation effort is very high. If we were not to specify an ROI (and usually this is the case), then for a given image of size $M \times N$ and a block size B , the algorithm has to compute $(M - B + 1) * (N - B + 1)$ patches. Then they have to take each patch and compare it to the other patches, thus the time grows exponentially. Also, the fuzzy logic works rather well if the ROI area is given, but in case the entire image is ROI, then it fails. Also, the algorithm signals a lot of false positives for larger areas.

Several years later, in [64] they enhanced the original paper from 2008 with some extra steps. The first major change the authors did is to enhance the search mechanism. They are proposing a two-step search. First, they are using a weight-transformation of the blocks. The main idea is to be able to increase the performance of the block comparison by first grouping similar blocks into a structure similar to a dictionary, where several blocks that resemble are categorized in the same “key” of the dictionary. Later the comparison of blocks will be done only per key, that being said, they will take all blocks which belong to the same key and try to see what is the best match. The authors are comparing several mechanisms for various methods of block grouping like equal numbers, even numbers, odd numbers, prime numbers, and a weight transformation proposed by them.

Another relevant improvement in this search mechanism was later proposed by authors in [65], where they are using only the central pixel as the key, rather than computing a value by applying the weight transformation. After that, they use the information to search for similar blocks inside the grouped blocks. They are using again for block comparison the Zero Connectivity algorithm proposed in the [63]. After this operation, they are proposing a vector filtering mechanism to eliminate a lot of false positives. Shortly they are implying that if two blocks are similar, their distance should be short as possible in a normal situation. If the distance is greater (the authors in all papers do not present the values used) than a threshold they are marking that block as potentially forged. The next step in their research is to decrease even further the false positive rates. For this, they’ve noticed that, usually Criminisi’s algorithm copies patches from different areas. Based on this observation, they proposed to use a multi-region correlation. Shortly blocks identified in previous steps are grouped (if they are connected), thus obtaining some bigger regions. For each region, it is identified where are the reference blocks to it. If there is a self-relationship group or a pair-relationship group then these area is marked as not forged. So if we were to summarize this last step, the region created by blocks that are connected is marked as forged, as long as there are referenced blocks from

other regions (the authors proposed a value of 3 – thus if the region is referencing at least 3 other regions than that blocked is marked as forged). Again, as a result, the method overall seems to produce good results for Crimini object removal cases, but it still has problems with diffusion-based inpainting or with enhanced example-based object removal methods. Another two potential improvements can be addressed in the area of speed, it still takes a lot of time to calculate the hash value of each block, and secondly the fact that they apply zero connectivity mechanisms is good for areas at the inpainted region, but to apply some mechanisms somewhere in the middle of the inpainted region might seem as a redundant step. Last, but not least, from all the analyzed pictures it seems it produces good results when the inpainted area (removed object) is either in a fully textured region or the object removed is at the boundary between two texturized regions.

Another improvement was suggested in [66] in which the authors suggested a jump patch approach increases the overall results. Later in [65] authors suggested several improvements to the original 2013 paper [64]. The first difference compared to the original paper is that they improve the two searches by changing the weight approach with a simpler and faster approach. They take as “key” only the central pixel value of each block. Secondly then computing the differences matrix, they suggest replacing the AND with an OR between the color components, in this way the recent paper makes a stricter comparison of blocks. This algorithm seems an important improvement versus the previous ones, but it still suffers in different areas:

- They have tested only on recent Crimini’s variation paper, not on state-of-the-art (at that time) methods for inpainting (and especially for object removal) – they’ve used [67].
- The computation effort is still very high. Again, applying the GZL in the middle-forged area seems a little too exhaustive and probably will not affect the overall results.

The same authors, several years later proposed in [68] to extend the above framework with a machine learning-based approach. If the above frameworks do not generate any false positives, images are fed into an ensemble classifier. For feature extractions, the authors rely on the fact that there has to be a Generalized Gaussian Distribution between the DCT coefficients of various blocks. Other authors tried similar approaches like the authors proposed in the CMFD framework. For e.g. in [69] suggested using Gabor magnitude as feature extractions to extract the features. The rest of the methods resemble the CMFD proposed framework (like block comparing, sorting, and detection). The interesting thing regarding this paper is the claims that the method is robust against inpainting methods, although they didn’t mention what method they applied for object removal. Another interesting method was proposed [70]. Here the authors have analyzed the impact of sparsity-based inpainting on the color correlations. They’ve noticed that a modified canonical correlation analysis might be able to properly detect these artifacts. Again, we must emphasize that the method works only on sparsity-based inpainting. An interesting direction was in analyzing the reflectance of the forged and non-forged areas. The authors in [71] have identified that some inpainting methods leave these traces and they suggest that this approach should be combined with the CMFD framework. Below you can see a summary table of the methods described.

Table 1. Classical inpaiting forgery detection.

Reference article	Year	Observations
[63]	2008	The first found method which tackles inpainting methods. They test against Crimini dataset. Method relies on detecting similar patches and applies a fuzzy logic for similar patches.
[64]	2013	They continue the work on [63] and add several mechanism to exclude a lot of false positives
[65]	2015	The authors come with 2 proposals – first do not compute the block differences – only compare the central pixel (this improves performance and the accuracy is not that much affected). Secondly they proposed an improved method comparing to [64] of filtering – eliminating false postives.
[66]	2013	Similar method as the on in [64]. For better results / faster computation – they suggest a jump patch best approach.

[68]	2018	The same authors which proposed [65], come with an additional step which consists of an ensemble learning. They rely on Generalized Gaussian Distribution between the DCT coefficients of various blocks
[69]	2015	The authors took the CMFD framework proposed in [62] but use as feature extractions the Gabor magnitude
[70]	2018	The authors took the CMFD framework proposed in [62] but use as feature extractions the color correlations between patches
[71]	2020	The focus was on analyzing the reflectance of the forged and non-forged areas

3.1.2. Machine learning-based methods

Before we start on a deep review of deep learning methods for image inpainting, we would like to summarize all the methods classical methods that constitute the basis for the deep learning approach and their limitations. The first focus was on analyzing the physical traces. Here some of the base ideas are searching for blur inconsistencies, chromatic aberrations inconsistencies, or radial lens distortions (for a more in-depth review we strongly suggest the work of [5] and [44]. Usually, the methods imply splitting the image into patches and searching for inconsistencies in one of the three areas mentioned earlier. The work does not target a specific type of forgery because it relies on the assumption that whatever type of forgery method was applied (copy-paste, resampling, inpainting, copy-move), they all disturb the overall balance on the above properties. Mention here a few relevant papers – which are going to be the basis for machine learning methods:

- For blur inconsistencies, one of the most cited paper is [72]. They rely on the assumption that if the targeted original image contains some blur, by combining parts from other images the blur will be inconsistent. They are proposing a method to analyze the gradients and detect inconsistencies among them. Of course, the method does not give good results in case the target digital image does not contain some blur artifacts.
- Some other researchers focused on other camera properties like lens traces. The authors in [73], postulated that in some cases it is possible to detect especially copy-move forgeries by analyzing the lens discrepancies at block level. Their method basically detects edges and extracts distorted lines and use this in classical block based approach to analyze discrepancies within the image. The problem with this approach is that if the targeted area is either to big – or too small the results yielded are not very satisfactory, also there is another problem with low resolution images, because they tend to yield false positive results.
- A very good camera-specific parameter that was heavily studied was the noise generated at image acquisition. Several authors have proposed different mechanisms to detect inconsistencies at block noise levels. Some authors even went in the direction that based on noise patterns they are able to uniquely identify the camera model. To name a few of the most cited works [74][75]. For e.g. in[75], the authors suggested computing noise for non-overlapping blocks and then unifying regions which have similar noise – thus partitioning the image into areas of the same noise intensity. The authors suggested using wavelet + a median filter approach on the grayscale image to compute the noise. Of course, the main limitations of these methods vary from false positives to the impossibility of the methods to detecting if noise level degradation is very small (a lot of anti-forgery mechanisms can exploit this method).
- Color filter array methods or demosaicking methods(CFA) – rely on the observation that most cameras capture only one color per pixel and then use some algorithms to interpolate these values. The forgery detection mechanism based on CFA – basically detects inconsistencies at block levels between the patterns generated by CFA. One of the most cited work is [76], in which the authors are proposing to use a small block (up 2x2) to detect inconsistencies in the CFA pattern. They are extracting the green channel from the image and calculate a prediction error and analyze the variance of the errors in order to mark the non-overlapping block as forged or not. The method yield good results as the original image does not suffer some post-processing operations like color normalization.

One of the first machine learning-based methods was the work of Linchuan Shen, Gaobo Yang, Leida Li, and Xingming Sun [77]. Here they propose a support vector machine classifier composed of the following features: local binary pattern features, gray-level co-occurrence matrix features, and gradient features (actually they suggest to use fourteen features extracted from patches). One key aspect to mention in their research is the robustness of this method especially on post-processing operations (like jpeg compression, scaling, noise, etc). Another variant of ML-based detection is the work [78] in which they employ a standard CNN model to be able to detect tampered regions or the work [79] which suggests using four Resnet modules to extract features from the image blocks or a later one which employs a hybrid model of LSTM and CNN [80]. Similar to the above was also the work in [81] in which they use a simple CNN and extract general features from inpainted vs non-inpainted areas. The main problem with the above methods is that they either test the network on Criminisi's old paper, or they randomly select a center area on real images and apply the latest image inpainting methods, neither with the potential to be used in real-life scenarios.

In the last years, the focus for detecting inpainting images is to apply higher and more complex machine learning models, with some strong feature extract mechanisms. For example in [82] [83] or [84], the authors suggested that the noise variance is disturbing the inpainted area, thus by applying three Resnet feature blocks in a multi-scale network, they obtained very good results. They use for testing the latest state-of-the-art deep inpainting methods, but they random masks for applying the inpainting methods, thus not a very realistic approach. Another fusion-based approach was proposed in [85]. Their work was actually inspired by [79], [86][87], and they suggest to use of three enhancements blocks: Steganalysis Rich Model – to enhance noise inconsistency, Pre-Filtering to enhance discrepancy in high-frequency components, and a Bayar filter to enable the adaptive acquisition of low-level prediction residual features. After these blocks, the authors used a search block to detect the areas based on the three enhancement blocks. The last step is a decision block – because usually there is an inconsistency in how pixels are classified. If we take a closer look at the results, the authors obtained very good results compared to previous algorithms. An interesting approach is also on how they generate the data – they incorporate various datasets of pictures and use several inpainting methods to generate forged data. The problem still with their approach is that the mask data used for the inpainting process is generated randomly, not in a realistic approach. In the same category as IID-NET methods, others have suggested [88] incorporating more enhancement blocks to make the detection more reliable and for a more general range of forgeries, and they used several datasets some of which present not very realistic tampering (more on the datasets on the next chapter).

Another more complex architecture was added in [89]. The authors based their network architecture on the work of [90]. Again, like in previous machine learning approaches, the authors try to create a method to suit all types of forgery like splicing, copy-move, or inpainting. The novelty in [89] was to add a top-down and a bottom-up path encompassed by another color correlation module called the Spatio-Channel Correlation module. Basically in the top-down area, they are trying to detect features at different scales, while in the bottom-up part, they start from the mask to find and strengthen the correlation between forged and not forged areas. They claim to obtain top SOT compared with other systems like [91], [92]. Again, it seems the authors opted to create their own data set to tackle various forgery types like splicing, copy-move, and object removal. An interesting thing was that they didn't compare their work with another claimed SOTA [85]. Also, the work based on noise inconsistencies [83] appears to surpass these approaches in terms of results, but not in terms of speed and model complexity (the authors in [89] claimed to use a small network model with approximately 3 M parameters only).

To summarize the above-presented machine learning based, we can see two trends regarding detection types: algorithms that focus solely on detecting inpainting (object removal) forgeries and methods that try to tackle all kinds of possible forgeries. From the dataset, point of view we see that most of the users either manually generate datasets by applying some random masks or use pretty old datasets. From the results point of view, we can see that there are several areas that offer good

results, the most promising ones seem to be in the detection of a mismatch in either noise and or color information presented. In the below table, you should see a summary of the presented data above.

Table 2. Machine learning based inpainting forgery detection.

Reference article	Year	Observations
[77]	2017	The main idea was to use a SVM classifier composed of the following features: local binary pattern features, gray-level co-occurrence matrix features, and gradient features (actually they suggest to use fourteen features extracted from patches).
[78]	2018	Standard CNN model on which they trained original / altered patches.
[79]	2019	Same as principle idea as [78] but they choose a Resnet model
[80]	2020	The authors employed a combination of Resnet + LSTM to better portray the differences between altered vs non-altered regions. All the above methods were mainly tested against Criminisi’s initial paper, thus not having to “compete” with latest image inpainting methods at that time.
[81]	2021	A tweaked version of a VGG model architecture
[83]	2022	A CNN model with focus on detecting noise inconsistencies
[84]	2022	A U-NET VGG model which adds an enhancement block of 5 filters (4 SRM + Laplacian) to be able to better detect inpainted areas.
[85]	2022	The authors suggest to use of three enhancements blocks: Steganalysis Rich Model – to enhance noise inconsistency, Pre-Filtering to enhance discrepancy in high-frequency components, and a Bayar filter to enable the adaptive acquisition of low-level prediction residual features similar to the Mantranet architecture.

3.2. Video Object removal detection

In the next part, we will present a rather small researched area [93], which is forgery detection inside videos. This subject contains little research compared to image forgery detection because usually people say that forgeries inside videos can be characterized/detected in one of the two options (or maybe a combination of them):

- Decode the video frames and apply already established image algorithms on the frames of the videos .
- Apply at the encoded video a mechanism to detect if frames have been tampered with (removed)

We had focus mainly on the first items. One of the first video forgery detection with a focus on inpainting was the [94] which focused on the detection of ghost artifacts introduced usually in the process of object removal. Later in [95] the authors are extracting frames from videos and applying the already started CMFD procedure with block-based similar to the work done in [63]. In [96], authors utilize an analysis of spatial-temporally coherence to identify areas with unusual coherence. Later in [97] enhance the previous idea with a Local Binary pattern to better extract / detect forged areas from videos. In the latest years, method involving some deep learning methods have been suggested. In [98] the authors are extending the work [99], by adding a multivariate process to enhance the detection phase and increasing the searching speedness. In [100] suggested using several Resnet networks (one for each type of forgery) plus the last one for following the forged area. As a basis for detecting inpainting, they are using a filter similar as in [87]. Similar ideas as in image copy-paste methods, have been investigated in also in [101]. The authors suggested that using Histogram of Oriented Gradients as a mechanism for feature description and detection, is going to yield good results. The method indeed works best in case an object has been replicated. Also another good feature of HOG is the fact that is robust against various types of attacks. Conversely, in situations where there is a depiction containing comparable entities, the algorithm may erroneously classify it as counterfeit. Another element to consider is that if an object is removed from a scene and the object is surrounded by a uniform region, the method will be unable to detect it.

4. Image inpainting datasets Datasets

Based on our previous deep analyses, we noticed that the majority of the proposed methods are employing a self-made dataset. This is causing a lot of effort while trying to compare with a previous SOTA to check if the newly proposed methods indeed add improvements. Classical inpainting detection mostly used the initial work of Criminisi to be able to detect (with some small exceptions). On the other hand, deep learning methods have tried to use newer variants for inpainting/object removal. The problem with this approach is that each machine learning forgery method, employed another inpainting mechanism they've considered relevant. Of course, there is another aspect, especially with the deep learning methods: usually, authors try to compare themselves with the latest inpainting methods thus there is also the need to have an open dataset that can grow bigger and bigger. Also, another aspect is related to automatic vs. human-based generated datasets. For e.g., there are very few research articles that take into consideration manually well-crafted forgery using various tools (like Photoshop). In order to circumvent these limitations, several authors tried to standardize the datasets. Our focus will be to present datasets that are pertaining to the inpainting/object removal detections (but if there are some datasets that have several other categories they will be presented as well). For a deep review of generic datasets used in forgery detection systems, we recommend the work of [5], [102]–[104].

To make the distinction clearer, the first focus was on generic datasets – that include inpainting/object removal (either automatic or manually generated) and in the second part, datasets specifically designed for the task of detecting inpainting methods.

4.1. General forgery datasets

One of the first truly realistic forgery datasets is done in [105] and in [62]. The MICC dataset was one of the widely accepted datasets for image forgery. It actually consists of 4 sub-datasets called F220, F2000, F8multi, and F600. The CMFD dataset contains updates that were done manually using professional tools. It is a generic dataset for copy-move scenarios, not only object removal use cases. Another relevant property worth mentioning is the fact that the dataset is grouped by the types of cameras which acquire the pictures. Analyzing the literature for usage of this dataset, it was observed that the dataset was not referenced, probably because it contains large images, and using a block-based comparison needed a lot of time to compute. Approximately at the same time, authors in [106] and [107] proposed a similar dataset again focused on copy-move. Still, some of the pictures can be considered object removal rather than copy-pasted and thus constitute a good measure to test inpainting methods. What differentiates these datasets from the CMFD one is the fact that the authors proposed here some variants of the forged ones which undergo some post-processing anti-forensic measures like jpeg compression, blurring, and noise adding. An important item to mention is that the images for the Comofod dataset were generated using Adobe Photoshop (especially for the splicing artifacts). The interesting, unique area for the Comofod dataset is actually that the authors tried to have images which naturally contain similar but genuine objects. For the Casia dataset, the authors actually introduce two sub-datasets namely Casia V1 and Casia V2. The second version contains pictures with several sizes (up to 900 x 600) and an important item is that they add also in some instances, some pre-processing and post-processing operations like resizing, distortion, blurring, noising, etc – thus making the detection harder. Some the images are quite realistic generated – the copy region is not a simple copy-move region, but the copied region is also resized, rotate and went through additional processing. Also is important to notice that the original dataset didn't offered the masks, so the following [108] can be used as a reference for the masks and also some filtering of the initial dataset (miss-categorizations or duplicates removal). In the same category but newer versions are the COVERAGE dataset [109] or the work of Pawel Korus in his papers [110], [111]. The COVERAGE dataset although rather small (100 images with an average size of around 400x500 pixels) tries to make things even harder by suggesting pictures that contain similar but genuinely valid objects. The "copied" items inside their images, thus already resemble multiple instances of similar objects, and the difficult task is in detecting real-faked areas vs similar instances. The work of Pawel Korus contains one of the most realistic forgeries which focuses on several types

of different cameras (some images were own made some were taken from [112]). The size of the images is relatively bigger than previous work (1920 x 1080), but in total there are around only 220 forged images, and the forgery methods vary from copy-move to object removal. An important item to mention is that they used GIMP + Affinity photo for generating the forgeries and the dataset consists of 4 different camera model. The authors basically suggest that when copy-move or object removal is applied on the input image they distort the overall noise between patches inside the image. Another relevant dataset that started in 2015 with the Media Forensic Challenge for Darpa[113] [114], now consists of millions of media that cover several forged mechanisms including both images and videos. The majority of media data is collected from the internet from various sources and in different contexts to ensure a more generic dataset. From these millions of media, approximately 200k images are what the authors called High Provenance – basically the authors ensure that the images have not undergone any forgery attacks. For this large dataset, the authors employed experienced people for performing the various forgery attacks (and also not limited to using only a specific tool for the forgery generation). The MFC dataset consists of several sub-tasks – like image/video manipulation localization, splice image/video manipulation, provenance, graph building (detecting all the steps and sources a target image has undergone), GAN, etc. At this point, it is still considered one of the referenced forgery datasets both from quality and quantitatively perspectives. The MFC dataset (2019 version) has approximately 180 000 original images, obtained directly from camera, thus ensure the images are not manipulated in any way and all device specific settings – like e.g. ISO are known beforehand. Also to ensure a large diversity of the images, the authors used more than 500 different cameras in order to have a large poolset of different noise that can be used for training (PRNU based algorithms). From this approximately 16k images have been altered. Still, the main problem with our object-removal (content aware fill) / inpainting tasks is that a relatively small from this dataset consists of the type of forgery we are focused on. Even the authors in MFC recognized that they add several methods under the generic umbrella of copy-move alterations.

Table 3. General forgery datasets - for which a subset can be used for inpainting/object removal detection.

Name	Dataset Size (GB)	Number of pristine / forged pictures	Image size(s)	Type*	Mode	Observation
MICC	6.5 GB	1850 / 978	722 x 480 to 2048 x 1536	CM/OR	M	Some of the images are not very realistic, but it tries to generate several types of copy-move by applying rotation and scaling. The problem is that always the forged area is a rectangular
CMFD	1 GB	48 / 48	3264 x 2448 3888 x 2592 3072 x 2304 3039 x 2014	CM/OR	M	Very realistic dataset. Some of the images due to the fact that they use professional tools – are a mixing of copy-move, object removal, and sampling. Group by camera type. Important thing to notice is that there is no post processing operations done on the image, but because of the high-quality / size researches can do their own post processing. Images were processed using GIMP
CoMoFoD	3 GB	200 + 60 / 200 + 60	512 x 512 3000x2000	CM/OR	M	Canon camera used only. They've used 6 post processing operations, for e.g. JPEG compression with 9 various quality level, or changing brightness, noise etc. The operations were done in Photoshop. 3Gb is only the small variant of the dataset

CASIA	3GB V2	7491 / 5123	160x240 to 900x600	CM+S	N	Contains different type of copied areas like resize, rotate and post-processing of the forged area.
COVERAGE	150MB	100/100	235x340 to 752x472	CP	N	Original images already contain similar objects, thus makes harder to detect. The forged is relative large – 60% of the images have at least 10% forged area.
Realistic Tamperin g Dataset	1.5 GB	220/220	1920x1080	CP/OR	M/A	The dataset contains 4 different types of camera, and focus on the inconsistencies at noise level between patches. The images were pre/post processed with GIMP
MFC	150GB	16k / 2 M	All sizes	CP/OR	N	They've used a series of techniques like simple copy-move, to content aware fill, seam carving etc.

*Type CP = Copy-Move, OR = Object removal, S = Splicing. Mode M=Manual, A=Automatic, N=Not mentioned.

4.2. Image inpainting specific datasets

One of the first datasets with a specific division of object removal was the work in [115]. They have indeed a very large dataset about 25000 images which were generated using [116]. In order to generate the dataset, they actually use the MSCOCO [117] which contains descriptive information. Then they select randomly from each image an descriptive region and dilate that raw mask and apply the inpainting algorithm and removing images for which standard deviation for the inpainting regions was below a given threshold. The problem with the dataset is that they generated the masks randomly, thus in highly texturized areas (or where the MSCOCO description is not accurate), or at border regions, the forged images are blurring and can be easily detected even by human observation. In the same methodology as the previous authors, is the dataset entitled IMD2020 [118]. In this dataset, the authors took 35k pictures from Flickr, manually reviewed them, and then used a combination of classical inpainting from OpenCV and a machine learning-based model for inpainting [119]. One issue pertaining to this dataset was the incomplete differentiation between the original and forged areas in the masks.. If we compare the pixels from the original vs forged area, we will have a lot of differences, not only the altered region (probably the images went some post-processing which were not mentioned in the paper, or the images were altered as part of the inpainting procedure). Another problem with this dataset is that not all forged regions are realistic. One of the recent datasets proposed especially for the inpainting (object removal) detection is the work [85]. They present about 10k images, but as a novelty they do not target one specific inpainting method but rather propose 10 methods (6 classical method and 4 machine learning based). Still we observe the same problem with the mask generated for the inpainting process. Because they've generated the mask with different sizes and shapes, and do not take into account the nature of each image, some of the generated inpaintings are not very satisfactory. Also another problem is that the dataset consists of very small images – all have the same 256x256 size.

Table 4. Image inpainting forensic dataset.

Name	Dataset Size (GB)	Number of pristine / forged pictures	Image size(s)	Observation
DEFACTO INPAINTING	13 GB	10312 / 25000 (they've applied inpainting for same image but for different areas)	180x240 to 640x640	Some of the images are not very realistic inpainted due to the automatic randomized selection of the area from the MSCOCO dataset
IMD2020	38GB	35k / 35K	640x800 to 1024x920	Some of the images are not very realistic, and the forged image underwent some additional changes (probably some noise filtering / color

				unifirmatization). An interesting fact is that they manually selecting area and than using an automated algorithm, that means no post-processing / enhacements
IID-NET	1.2GB	11k / 11k	256x256	Random masks (based on MSCOCO) + 11 different automated algorithms of filling / removing object/ The idea is interesting to try to tackle different inpainting algorithms, still there are some problems in how the mask inpainted area is choosen. Also another problem is that although several inpainting algoritms are tested, they are applied on different images.

5. Results and Discussion

In the following section an analysis is performed on the results obtained from various forgery detection methods applied on various image inpainting methods. The first focus is the dataset. There are some generic forgery datasets, but they lack specificacity (for e.g. Casia, MFC, etc) - they are not focus 100% on image inpainting / object removal. Also from previous analysis (see Dataset chapter), it was observed that in general each forgery detection method comes with its own dataset. With each new dataset, the authors take the some generic input dataset (e.g. MS Coco etc) and apply a different inpainting method, and then they anaylize their method on this context. But, by not being backwards compatible, each new method forgery method is not properly compared with previous others - especially if we talk about machine learning based methods. Also the way the inpainting masks are genereted – usually randomly take a region from the input image– focus the network model to detect different inconsistencies whicht the inpainting method can’t overcome. To address this, we’ve used the Google’s Open Images Dataset V7 released in October 2022 [120]. We’ve manually select 400 images with additional segmented masks - we’ve only used one mask per image. The selected segmented masks were choose not to be in a very texturized area - we’ve made this limitation because mostly all inpainting methods have problems filling highly texturized areas. Additionally because the forgery methods do not work properly well on big images we’ve emposed that selected images should be maximum of 1024x1024 in size. Another relevant aspect is that on the provided mask from Google’s dataset, we’ve noticed that some of the masks were very close at borders defined,so in order to enhance the inpainting methods, we’ve added a dilation with a kernel of 5x5.

Our next focus is generating the dataset based on various inpainting methods. Our work is inspired from [85], in which the authors select several inpainting methods an proposed a forgery mechanism to learn the intricacies of all those inpainting methods. The difference is that we did not generate a random mask, but rather we took a valid object from the image and also we apply same inpainting method with same mask to several inpainting methods - in this way we can throughly evaluate given on pair image / mask - and having several inpainted ouputs, how each forgery method is able to determine the traces. For the inpainting methods, we’ve used a number of 5 different inpainting methods: Criminisi original method, an improved version of the Patch Match algorithm [121], professional editing tools – GIMP [122] and two machine learning based methods – Lama [45] and a newer improved version called MAT: Mask-Aware Transformer for Large Hole Image Inpainting [123].

Next we’ve selected 6 different forgery detection mechanism to evaluate them on the above dataset. The first pick was the CMFD method proposed in [62]. Although the CMFD is focus on generic copy-move, we’ve wanted to check how good the method is able to detect inpainted areas. As parameters, we’ve used Zernike moments with a block size 13 - based on authors suggestions. The next item on our list was an classical object removal detection. Here we’ve picked the original paper from [63]. We’ve selected this one and not a newer variant - like [65] - because the newer variants improve only in both speed and accuracy - eliminating a lot of false positives. The next four methods we’ve picked were machine learning based. The first on in the list was the Mantranet method [91] -

as it was the basis for numerous newer methods. Based on the authors claims the method should work on detecting also object removal because they've trained with images generated by Opencv inpainting methods and additionally they used more than 300 different classes - like blur inconsistencies, gaussian inconsistencies etc - to train the network. The next method based on matranet net was the IID network [85] which solely focus on inpainting detection. Additionally we've included two more newer methods - Focal [124] which adds a clustering method to be able to differentiate between forged and not forged areas in the image and PSCC-Net which ecompasses a Spatio-Channel Correlation Module to be able to focus on various traces. All the machine learning based detection tests were generated on machine with a NVIDIA Quadro RTX 8000 video card.

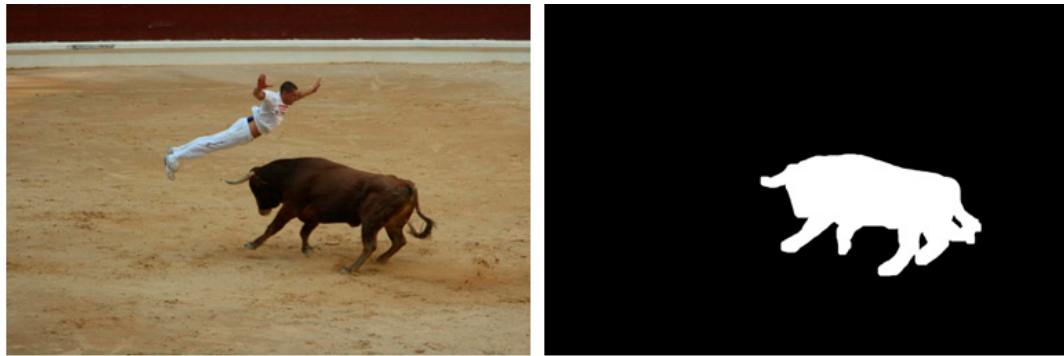


Figure 6. Original and mask image from dataset.

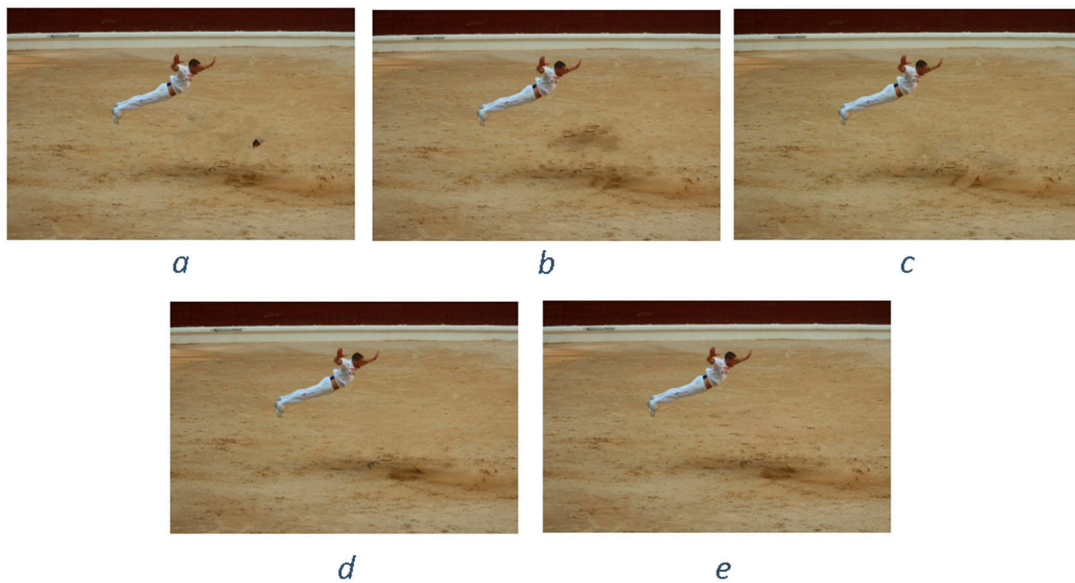


Figure 7. Inpainted results: a – Criminisi[25], b – Gimp[122], c – NonLocalPatch[121], Lama[45], Mat[123].

As it is noticed, on the above figure, the first result of [25] does contain some visible artifacts, while the others are able to complete the image in a very natural way. [122] by duplicating a section with a lower luminosity and afterwards contrasting it with the overall context of the image, it becomes possible to ascertain that said area has been replicated from a neighboring locatio, so a simple block lookup comparison determines the similar regions. [121] duplicated and interpolated more smootly the regions nearby, but it is noticed it introduced a blurring arficat on the region of the removed object as one can noticed in the below figure in the highlighted area.



Figure 8. Sample artifact introduced by [121].

Next we will analyze how each of the six algorithms DEBI [63], CMFD[62], IID[85], Mantranet[91], PSCCNET[89] and Focal[124].

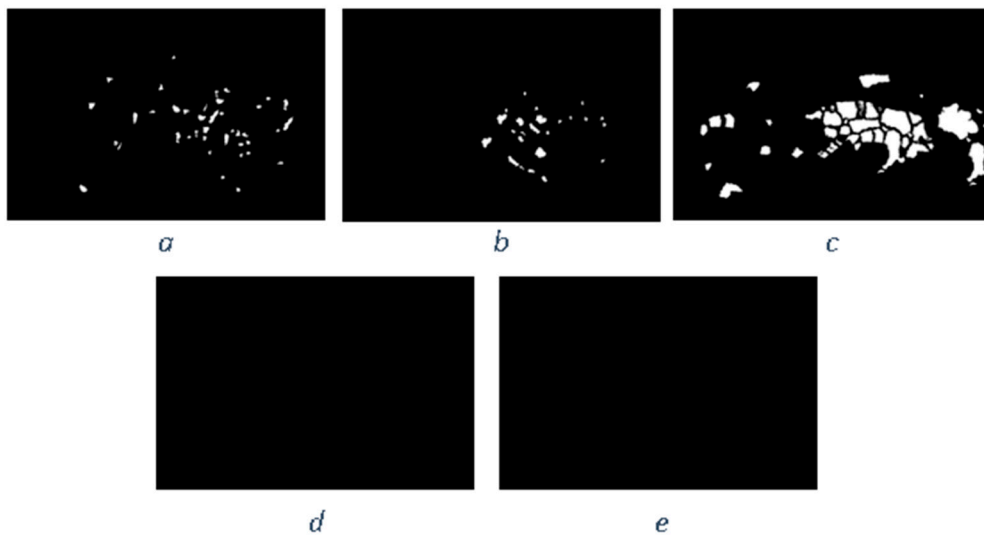


Figure 9. DEBI [63] results on inpainted images with: a – Criminisi[25], b – Gimp[122], c – NonLocalPatch[121], Lama[45], Mat[123].

Because DEBI[63] uses the idea of block comparison, we can easily see that the method is able to detect some clues on images produced by Criminisi [25], Gimp[122] and NonLocalPatch[121]. The results indicate that some regions are tampered, because both 3 inpainting methods, work rather similarly by copying various patches from different regions. An interesting observation is that except the Criminisi method, all methods affect the overall pixel intensity, not just in the targeted masked area. That is why the [63] method is able to detect different regions - even some which are false positive as one can noticed in the on the above figure on the c picture. Somehow expected are the results for the machine learning inpainting methods. Because the machine learning methods do not copy a patch, but rather try to synthesize, the block based approach is not able to detect any similar blocks. A possible solution we've analyzed, was to search by similar blocks withing a given delta as pixel differences, but by doing so, we had received a lot of false positive results. Similar results we had obtained also from the CMFD framework – because the detection method is rather identical – they work by comparing blocks and applying some filtering logic on similar blocks.

The next analyzed method was Mantranet [91]. As it can be noticed from the below figure, the detection method works with good results on classical inpainting methods, but rather poorly on machine learning based inpainting methods. Also a relevant item, especially in the computation of F1 score, precision, recall etc., is the fact that [85], [89] and [91] give results an image with gray level intensities. This means that if the region is perfect white, there is a high confidence that the pixels are tampered with, while lower pixel values gives lower confidence. In the evaluation metric measurements, we shall analyze different values of these pixels intensities and see how they affect the overall results.

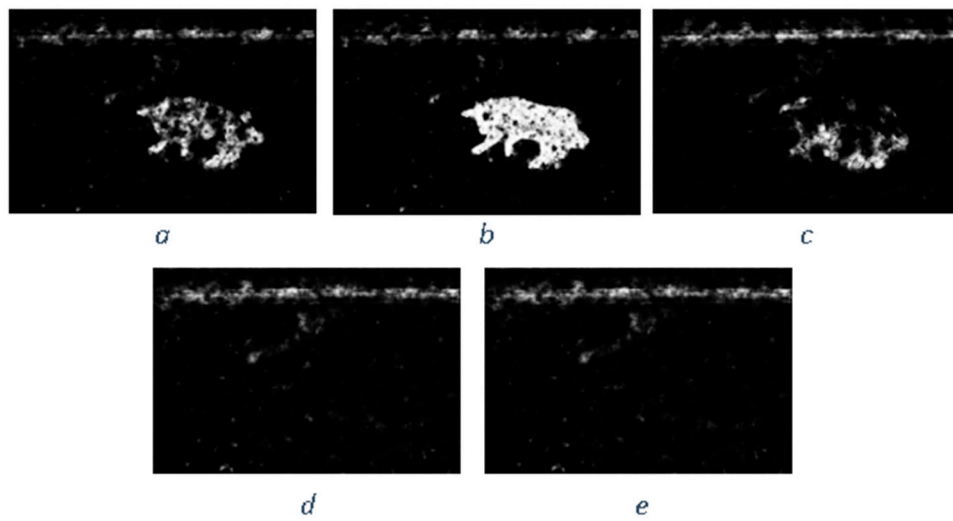


Figure 10. Mantranet [91] results on inpainted images with: a – Criminisi[25], b – Gimp[122], c – Non-LocalPatch[121], d – Lama[45], e – Mat[123].

On IID[85] the results for this image on all 5 different inpainted images were very promising, with a small observation that for [121] it detected only the surrounding area

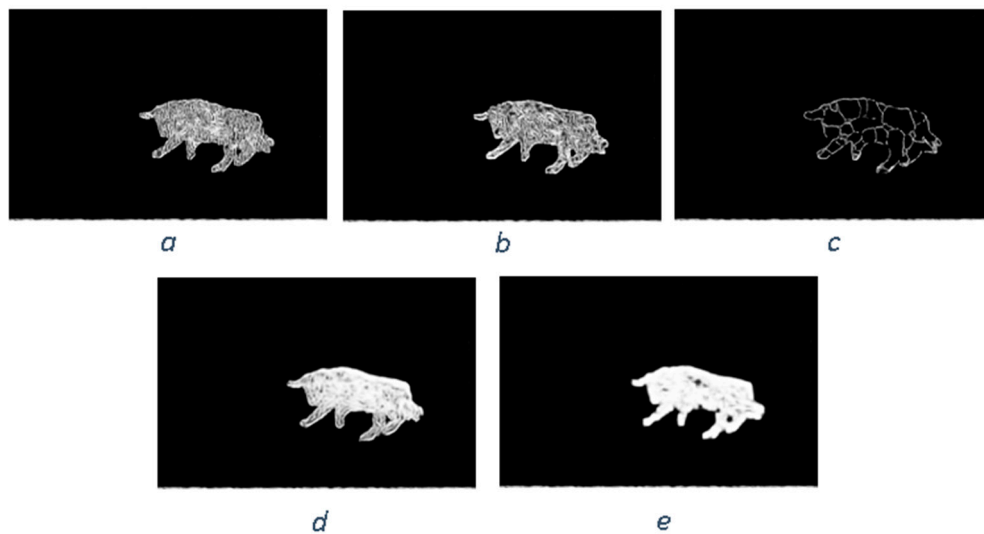


Figure 11. IID [85] results on inpainted images with: a – Criminisi[25], b – Gimp[122], c – Non-LocalPatch[121], d – Lama[45], e – Mat[123].

On PSCCNET [89] for the above image, the results were rather poor, on the other hand, interesting results we've observed on the Focal [124] method. The method is able to successfully detect forged areas for block based inpainting methods. It behaves rather strangely on the machine learning based method, where it detected artifacts incorrectly related to the object which was not removed, but it was altered in the inpainting method.

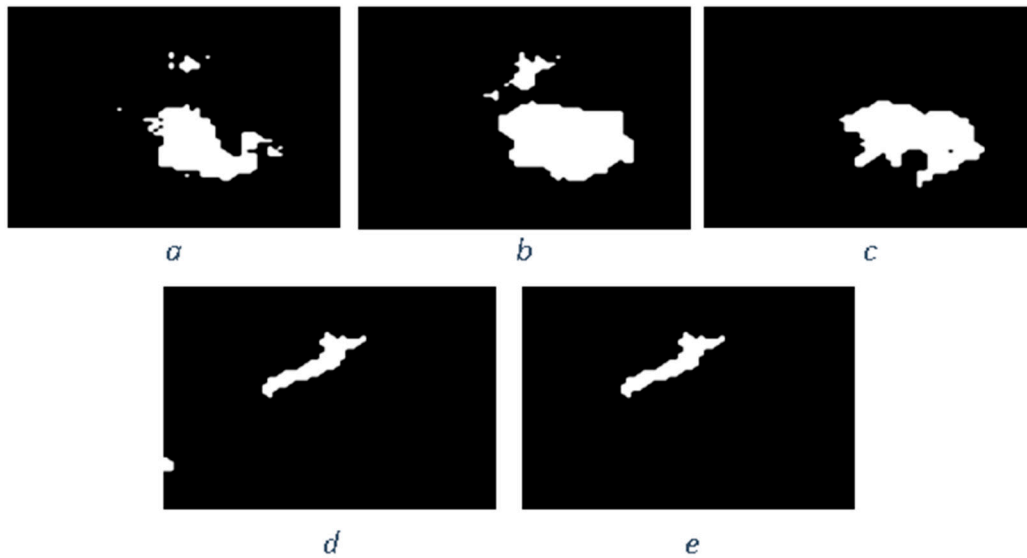


Figure 12. Focal [124] results on inpainted images with: a – Criminisi[25], b – Gimp[122], c – Non-LocalPatch[121], Lama[45], Mat[123].

Following this, a comprehensive study has been undertaken to evaluate the results from the perspective of a measuring meter. Initially, an evaluation was conducted to assess the performance of the two block-based method detection techniques. Upon conducting an analysis of the F1 score, precision, recall, and intersection over union (IoU), it becomes evident that the approaches employed yield unsatisfactory outcomes. The CMFD method yielded superior outcomes in comparison to the DEBI method. It is likely that the implementation of enhancements discussed in reference [65] would enhance the overall performance of DEBI. It is important to acknowledge that the existing methodology (DEBI+CMFD) lacks the capability to detect regions that have experienced indirect replication. This is the reason why we have not shown any results for either DEBI or CMFD in relation to machine learning-based inpainting methods. In summary, the evaluation of the NonLocalPatch inpainting approach using both detection methods suggests a modest level of performance based on the metrics employed. Although the current system demonstrates some accurate detections, there is considerable scope for enhancement, particularly in relation to the bounding box overlap (IoU) and the reduction of erroneous detections (precision). The poor results on detecting images which underwent an inpainting process via Criminisi method, might be explained by the fact that the areas to be inpainted contained some shadow elements, and also the resolution of test images is quite big comparing with the original paper (1024x1024 vs 256x256). Based on the measurements acquired, it is evident that the DEBI and CMFD detection algorithms, which employ the Criminisi and GIMP inpainting approaches, encounter significant challenges when attempting to detect forgeries. The methods have a notable inadequacy in forgery detection, as it demonstrates a failure to accurately identify a considerable proportion of the region (low recall). Moreover, despite its ability to identify potentially problematic areas, the effectiveness of the detection system is often compromised, leading to a diminished Intersection over Union (IoU) score. The precision metric suggests that around one-third of the system's detections are accurate. Nevertheless, with careful examination of the subpar recall and IoU scores, it becomes apparent that improvements are necessary across all facets of the methodologies.

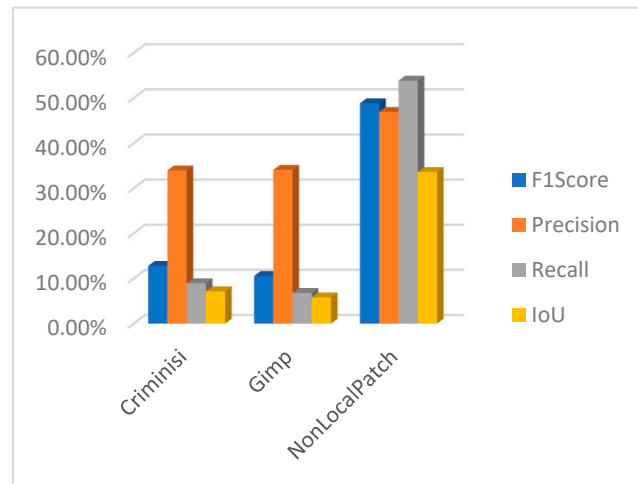


Figure 13. Evaluation metric for the results of DEBI detection method applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch.

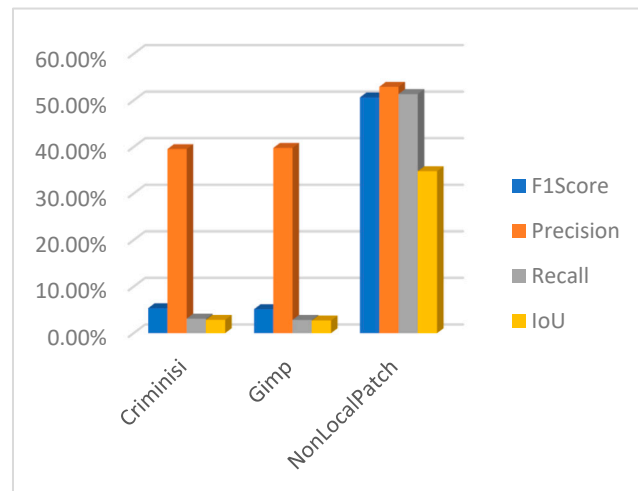


Figure 14. Evaluation metric for the results of CMFD detection method applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch.

The method proposed in [124] demonstrates superior efficacy in detecting the inpainting produced by [121] in comparison to the rest of the methods. In the case of the block-based approaches, namely Criminisi and Gimp, the detection model exhibits a reasonable level of accuracy in identifying inpainted regions, achieving approximately 60%. However, upon closer examination of the Intersection over Union (IoU) metric, which measures the overlap between the predicted and ground truth regions, it becomes evident that the system erroneously identifies additional sections as manipulated, resulting in an IoU of approximately 30%. The Focal method applied on the NonLocalPatch images has a commendable level of precision in its detections and effectively captures a substantial proportion of the objects that are there. The marginally reduced F1 score indicates a small imbalance, albeit without major divergence. The IoU score indicates that the model's localization accuracy is satisfactory, while there is room for improvement. It is possible that the accuracy was affected by the wrong selection of the mask for inpainting. Further information regarding the dataset talks on masks may provide insights into this matter. However, the performance indicators for Focal reveal subpar outcomes in relation to the Lama and Mat inpainting approaches and are subjective to a closer review.

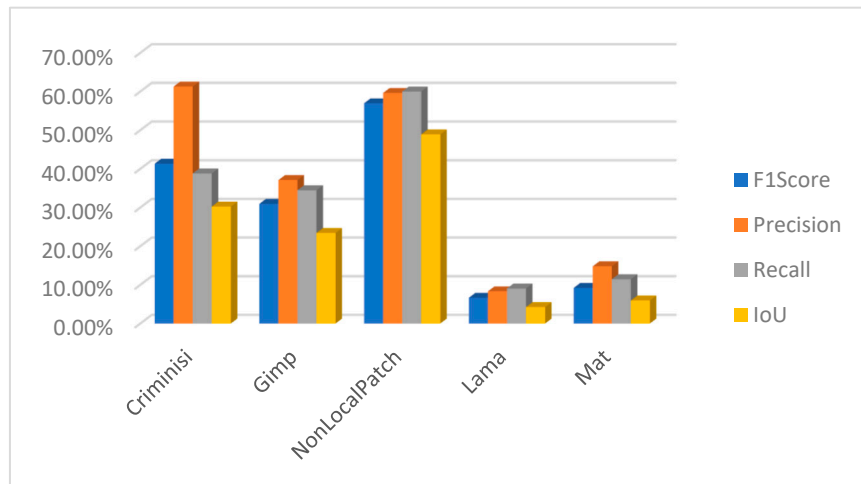


Figure 15. Evaluation metric for the results of Focal detection method applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch, Lama and Mat.

The methodologies described in [85], [89], and [91] do not produce a binary mask that may be used to identify counterfeit regions. In contrast, a heat map is presented by them. Pixels with higher values, which are represented by white pixels, imply that the utilized approach possesses a higher degree of confidence in accurately identifying modified pixels. Moreover, as an illustration, the methodology outlined in reference [89] uses a softmax function on the entire output to augment the response by integrating a binary categorization of the image as either counterfeit or genuine. In order to adequately evaluate the three procedures, we have utilized three separate approaches to analyze the outcomes of the tests. In this study, a set of three sample values (20, 70, 127) has been chosen as threshold for the purpose of identifying forged pixels. Pixels that surpass these values are categorized as indicative of forging, whilst those that fall below are classified as genuine. For example, when the threshold for [85] is set to 127 instead of 20, the Precision increases by 3%, but the IoU reduces by around 4%. A similar trend may be observed with the Matranet approach when the threshold is set to 127 compared to 20. with this case, there is a 12% gain in Precision, but a decrease of approximately 5% in IoU.

The utilization of [85] in machine learning approaches has demonstrated superior outcomes, alongside [123].

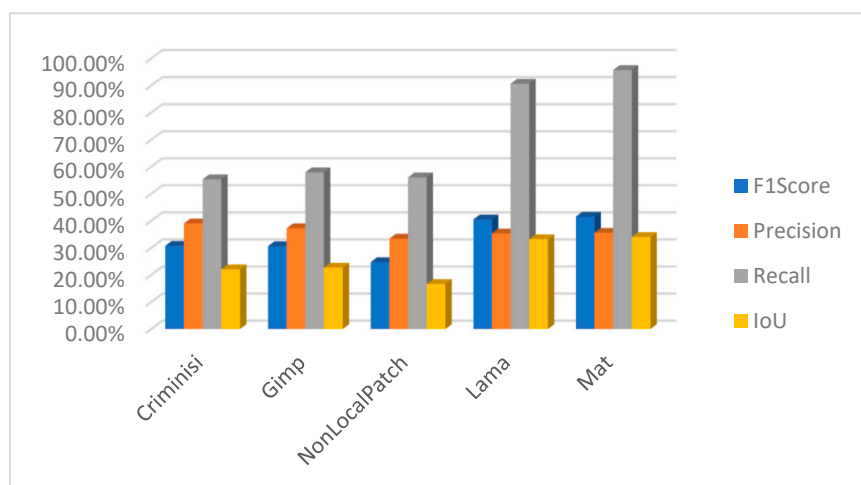


Figure 16. Evaluation metric for the results of IID-NET detection method applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch, Lama and Mat.

One interesting characteristic of the IID / Matranet / PSCC-NET methods is the observation of a high recall rate accompanied by relatively lower values of precision, F1 score, and IoU (see above and below diagrams). This pattern suggests that the model has an excessive tendency to identify detections, successfully capturing a majority of genuine objects. However, it also tends to incorrectly identify numerous non-objects. Furthermore, even when the model’s identifications are accurate, its ability to precisely localize objects may be compromised. This tendency may provide challenges in several situations, particularly when accuracy is of utmost importance. Additionally, it is possible that the model could get advantages from other optimization techniques in order to enhance precision while minimizing any substantial trade-offs in recall.

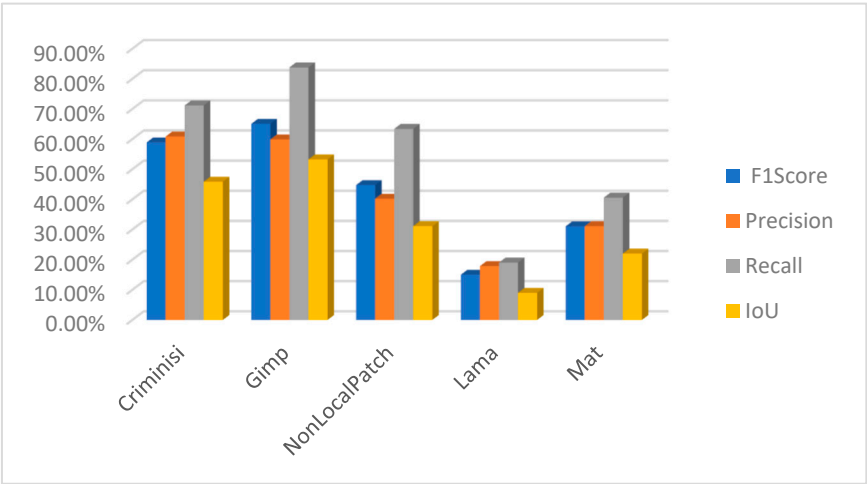


Figure 17. Evaluation metric for the results of Mantranet detection method applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch, Lama and Mat.

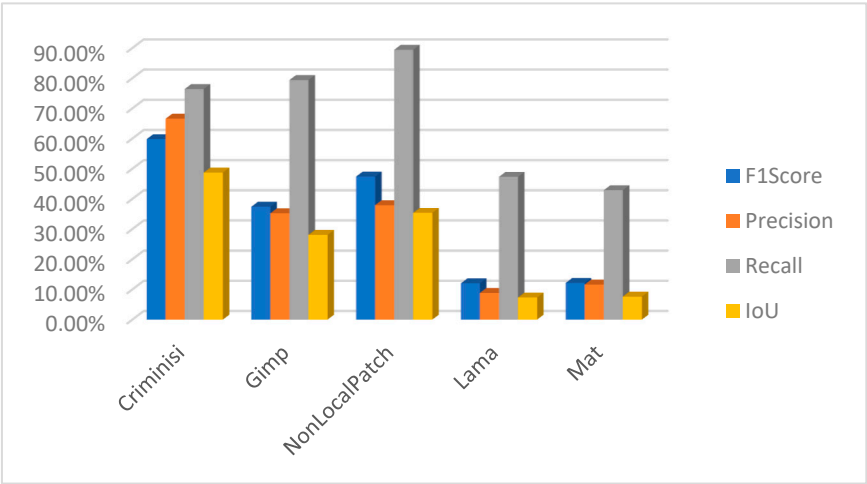


Figure 18. Evaluation metric for the results of PSCC-NET detection method applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch, Lama and Mat.

Based on the summary of the results presented below, it is observed that Focal [124] performs well for NonLocalBased inpainting methods, Mantranet [91] is able to detect older variants of patch based methods, and IID performs the best on the machine learning based inpainting methods.

	Inpainting method	F1	Precision	Recall	IoU
IIDNET	Criminisi	37.49%	39.05%	70.70%	29.30%
	Gimp	35.66%	37.16%	69.51%	28.32%
	NonLocalPatch	27.78%	33.51%	62.75%	19.20%
	Lama	40.40%	33.92%	96.20%	33.20%
	Mat	40.54%	33.75%	98.29%	33.11%
FOCAL	Criminisi	41.32%	61.30%	38.80%	30.24%
	Gimp	30.95%	37.10%	34.45%	23.41%
	NonLocalPatch	56.96%	59.64%	59.99%	48.89%
	Lama	6.67%	8.33%	9.05%	4.25%
	Mat	9.21%	14.79%	11.45%	5.94%
Mantranet	Criminisi	58.92%	60.87%	71.19%	45.87%
	Gimp	65.09%	59.91%	83.81%	53.27%
	NonLocalPatch	44.76%	40.23%	63.40%	31.11%
	Lama	15.03%	17.87%	19.04%	9.00%
	Mat	31.07%	31.11%	40.58%	22.00%
PSCC-NET	Criminisi	48.04%	77.34%	48.72%	37.61%
	Gimp	30.38%	41.13%	52.94%	23.03%
	NonLocalPatch	53.61%	52.37%	75.89%	42.57%
	Lama	7.95%	8.14%	29.04%	4.69%
	Mat	6.24%	8.81%	23.40%	3.74%

Figure 19. Summary evaluation metrics for the results of IID-NET,Focal,Mantranet,PSCC-NET detection methods applied on the inpainted Open Images Dataset V7 dataset with the following inpainting methods: Criminisi, Gimp, NonLocalPatch, Lama and Mat.

The domain of picture and video inpainting has witnessed significant advancements in recent years. The usage of exemplar-based techniques has been a particularly intriguing feature of this evolution. Through the use of these methodologies, it has become feasible to rectify substantial segments of impaired or absent regions within a picture or video. At its inception, the practice of inpainting was primarily limited to making modest modifications, such as repairing minor scratches or concealing imperfections in sensor devices. Nevertheless, at present, it has the capacity to address significantly more intricate difficulties, including the removal of considerable items. The applied strategies can be categorized into two primary groups: those that rely on partial differential equations and patch-based methods, sometimes referred to as exemplar-based methods. Contemporary photo editing software, designed for both professional and novice users, frequently incorporates sophisticated inpainting techniques. Exemplar-based inpainting can be understood as a technologically advanced and automated approach to the detection and removal of copy-move forgeries. In this procedure, segments are extracted from different regions of the image or video and adeptly merged to yield enhanced visual outcomes. The blending operation is a vital element of this process, since it must be executed flawlessly in order to guarantee the cohesiveness of the finished product. In the field of Convolutional Neural Networks, notable progress has been made in inpainting approaches, which have demonstrated superior performance compared to skilled human editors. This is particularly evident when employing content-aware filling methodologies. The capacity of CNN to construct a coherent visual storyline using limited data sets has showcased its substantial potential inside this domain.

Nevertheless, the task of detecting these inpainted alterations might be a significant problem, as conventional copy-move detection methods frequently have limited effectiveness in this context. There are various explanations for this phenomenon. Firstly, the target area under consideration may be too minuscule to be effectively detected. Secondly, the modified regions may closely resemble pre-

existing areas within the original image. Lastly, the inpainted areas could potentially consist of multiple distinct regions. In response to these constraints, a number of authors have proposed automated techniques for the detection of inpainting forgeries. These methodologies, akin to the ones employed in the detection of copy-move forgery, exploit visually identical picture patches to emphasize locations that raise suspicion. In addition, heuristic criteria are utilized by them in order to minimize the occurrence of false alarms. The heuristic rules exhibit a range of characteristics, since different authors employ various approaches, including the utilization of fuzzy logic, to address this issue. Efforts have also been made to exclude regions that lack indications of amalgamation from several geographical areas. However, the conventional approaches possess inherent constraints. The utilization of substantial computational resources and effort is frequently necessary in order to enhance detection accuracy by reducing patch sizes. Additionally, these methods encounter challenges in effectively mitigating false positives. In recent times, there has been an introduction of machine learning approaches to analyze disparities among various patches. An investigation is conducted to analyze noise, specifically focusing on Photo Response Non-Uniformity (PRNU), in order to quantify noise levels inside individual patches and detect any irregularities in the distribution of noise patterns throughout these patches. Prior research has focused on analyzing artifacts originating from the Color Filter Array and has attempted to utilize similar methodologies employed in noise analysis.. In the current epoch of deep learning, endeavors have been undertaken to devise an automated methodology capable of seamlessly amalgamating diverse artifacts and proficiently identifying anomalies. Nevertheless, the efficacy of these techniques is significantly impacted by the implementation of countermeasures such as noise reduction or addition, color correction, gamma correction, and other similar factors.

One of the key challenges in the field pertains to the limited availability of datasets, which hinders progress. Although there are several well-known datasets for detecting copy-move anomalies, the availability of datasets specifically designed for inpainting tasks, such as object removal or content-aware fill, is limited. Currently, there are just three datasets that can be categorized into this particular area. The restricted accessibility of datasets poses a significant obstacle in the examination and enhancement of inpainting detection techniques, impeding the prospective advancement in this captivating field of study. Hence, it is crucial that additional resources are allocated towards the creation and upkeep of extensive and superior inpainting datasets in order to advance the discipline

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, A.B. and R.B.; methodology, A.B. and R.B.; software, A.B.; validation, A.B., C.C.; formal analysis, R.B.; investigation, A.B. and R.B.; resources, A.B.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, R.B. and C.C.; visualization, A.B.; supervision, R.B.; project administration, R.B.; funding acquisition, N/A. All authors have read and agreed to the published version of the manuscript.” Please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Data Availability Statement: Once this paper will become available, we shall provide all the information, dataset, results and steps to reproduce. The details shall be available on this github repository: <https://github.com/jmaba/ImageInpaitingDetectionAReview>.

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

References

1. M. A. Qureshi and M. Deriche, “A bibliography of pixel-based blind image forgery detection techniques,” *Signal Process Image Commun*, vol. 39, pp. 46–74, 2015.
2. A. Gokhale, P. Mulay, D. Pramod, and R. Kulkarni, “A Bibliometric Analysis of Digital Image Forensics,” <https://doi.org/10.1080/0194262X.2020.1714529>, 2020, doi: 10.1080/0194262X.2020.1714529.
3. F. Casino *et al.*, “Research Trends, Challenges, and Emerging Topics in Digital Forensics: A Review of Reviews,” *IEEE Access*, vol. 10, pp. 25464–25493, 2022, doi: 10.1109/ACCESS.2022.3154059.
4. NOVA, “NOVA | scienceNOW | Profile: Hany Farid | PBS.” <https://www.pbs.org/wgbh/nova/sciencenow/0301/03.html> (accessed Sep. 09, 2021).

5. P. Korus, "Digital image integrity – a survey of protection and verification techniques," *Digit Signal Process*, vol. 71, pp. 1–26, Dec. 2017, doi: 10.1016/J.DSP.2017.08.009.
6. K. Liu, J. Li, S. Sabahat, and H. Bukhari, "Overview of Image Inpainting and Forensic Technology," 2022, doi: 10.1155/2022/9291971.
7. O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, "Image Inpainting: A Review," *Neural Processing Letters* 2019 51:2, vol. 51, no. 2, pp. 2007–2028, Dec. 2019, doi: 10.1007/S11063-019-10163-0.
8. D. J. B. Rojas, B. J. T. Fernandes, and S. M. M. I. Fernandes, "A Review on Image Inpainting Techniques and Datasets," *Proceedings - 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2020*, pp. 240–247, Nov. 2020, doi: 10.1109/SIBGRAPI51738.2020.00040.
9. J. Jam, C. Kendrick, K. Walker, V. Drouard, J. G. S. Hsu, and M. H. Yap, "A comprehensive review of past and present image inpainting methods," *Computer Vision and Image Understanding*, vol. 203, p. 103147, Feb. 2021, doi: 10.1016/J.CVIU.2020.103147.
10. Z. Tauber, Z. N. Li, and M. S. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 4, pp. 527–540, Jul. 2007, doi: 10.1109/TSMCC.2006.886967.
11. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, New York, New York, USA: ACM Press, 2000, pp. 417–424. doi: 10.1145/344779.344972.
12. M. Bertalmío, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, doi: 10.1109/CVPR.2001.990497.
13. M. Bertalmío, "Contrast invariant inpainting with a 3RD order, optimal PDE," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2, pp. 775–778, 2005, doi: 10.1109/ICIP.2005.1530171.
14. T. F. Chan and J. Shen, "Nontexture Inpainting by Curvature-Driven Diffusions," *J Vis Commun Image Represent*, vol. 12, no. 4, pp. 436–449, Dec. 2001, doi: 10.1006/JVCI.2001.0487.
15. T. F. Chan, S. H. Kang, and J. Shen, "Euler's Elastica and Curvature-Based Inpainting," <https://doi.org/10.1137/S0036139901390088>, vol. 63, no. 2, pp. 564–592, Jul. 2006, doi: 10.1137/S0036139901390088.
16. D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's," *Int J Comput Vis*, vol. 68, no. 1, pp. 65–82, Jun. 2006, doi: 10.1007/S11263-006-5631-Z/METRICS.
17. A. Telea, "An Image Inpainting Technique Based on the Fast Marching Method," <http://dx.doi.org/10.1080/10867651.2004.10487596>, vol. 9, no. 1, pp. 23–34, Jan. 2012, doi: 10.1080/10867651.2004.10487596.
18. C. B. Schönlieb' and A. Bertozzi, "Unconditionally stable schemes for higher order inpainting," *Commun Math Sci*, vol. 9, no. 2, pp. 413–457, 2011, doi: 10.4310/CMS.2011.V9.N2.A4.
19. P. Jidesh and S. George, "Gauss curvature-driven image inpainting for image reconstruction," <https://doi.org/10.1080/02533839.2012.751332>, vol. 37, no. 1, pp. 122–133, Jan. 2014, doi: 10.1080/02533839.2012.751332.
20. G. Sridevi and S. Srinivas Kumar, "p-Laplace Variational Image Inpainting Model Using Riesz Fractional Differential Filter," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 2, pp. 850–857, Apr. 2017, doi: 10.11591/IJECE.V7I2.PP850-857.
21. G. Sridevi and S. Srinivas Kumar, "Image Inpainting and Enhancement using Fractional Order Variational Model," *Def Sci J*, vol. 67, no. 3, pp. 308–315, Apr. 2017, doi: 10.14429/DSJ.67.10665.
22. G. Sridevi and S. Srinivas Kumar, "Image Inpainting Based on Fractional-Order Nonlinear Diffusion for Image Reconstruction," *Circuits Syst Signal Process*, vol. 38, pp. 3802–3817, 2019, doi: 10.1007/s00034-019-01029-w.
23. S. Gamini, V. V. Gudla, and C. H. Bindu, "Fractional-order Diffusion based Image Denoising Model," *International Journal of Electrical and Electronics Research*, vol. 10, no. 4, pp. 837–842, 2022, doi: 10.37391/IJEER.100413.
24. A. A. Efros and T. K. Leung, "Texture Synthesis by Non-parametric Sampling," 1999.
25. A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004, doi: 10.1109/TIP.2004.833105.
26. N. Kumar, L. Zhang, and S. Nayar, "What is a good nearest neighbors algorithm for finding similar patches in images?," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5303 LNCS, no. PART 2, pp. 364–378, 2008, doi: 10.1007/978-3-540-88688-4_27/COVER.
27. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing," https://gfx.cs.princeton.edu/pubs/Barnes_2009_PAR/ (accessed May 19, 2023).

28. T. Ružić and A. Pižurica, "Context-aware patch-based image inpainting using Markov random field modeling," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 444–456, Jan. 2015, doi: 10.1109/TIP.2014.2372479.
29. K. H. Jin and J. C. Ye, "Annihilating Filter-Based Low-Rank Hankel Matrix Approach for Image Inpainting," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3498–3511, Nov. 2015, doi: 10.1109/TIP.2015.2446943.
30. N. Kawai, T. Sato, and N. Yokoya, "Diminished Reality Based on Image Inpainting Considering Background Geometry," *IEEE Trans Vis Comput Graph*, vol. 22, no. 3, pp. 1236–1247, Mar. 2016, doi: 10.1109/TVCG.2015.2462368.
31. Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang, "Patch-Based Image Inpainting via Two-Stage Low Rank Approximation," *IEEE Trans Vis Comput Graph*, vol. 24, no. 6, pp. 2023–2036, Jun. 2018, doi: 10.1109/TVCG.2017.2702738.
32. H. Lu, Q. Liu, M. Zhang, Y. Wang, and X. Deng, "Gradient-based low rank method and its application in image inpainting," *Multimed Tools Appl*, vol. 77, no. 5, pp. 5969–5993, Mar. 2018, doi: 10.1007/S11042-017-4509-0/TABLES/5.
33. L. Shen, Y. Xu, and X. Zeng, "Wavelet inpainting with the ℓ_0 sparse regularization," *Appl Comput Harmon Anal*, vol. 41, no. 1, pp. 26–53, Jul. 2016, doi: 10.1016/J.ACHA.2015.03.001.
34. B. M. Waller, M. S. Nixon, and J. N. Carter, "Image reconstruction from local binary patterns," *Proceedings - 2013 International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2013*, pp. 118–123, 2013, doi: 10.1109/SITIS.2013.30.
35. Hong-an Li, Liuqing Hu, Jun Liu, Jing Zhang & Tian Ma (2023) A review of advances in image inpainting research, *The Imaging Science Journal*, DOI: [10.1080/13682199.2023.2212572](https://doi.org/10.1080/13682199.2023.2212572).
36. D. Rasaily and M. Dutta, "Comparative theory on image inpainting: A descriptive review," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, pp. 2925–2930, Jun. 2018, doi: 10.1109/ICECDS.2017.8389991.
37. B. Shen, W. Hu, Y. Zhang, and Y. J. Zhang, "Image inpainting via sparse representation," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 697–700, 2009, doi: 10.1109/ICASSP.2009.4959679.
38. Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1153–1165, May 2010, doi: 10.1109/TIP.2010.2042098.
39. P. Tiefenbacher, M. Sirch, M. Babaei, and G. Rigoll, "Wavelet contrast-based image inpainting with sparsity-driven initialization," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-August, pp. 3528–3532, Aug. 2016, doi: 10.1109/ICIP.2016.7533016.
40. A. Bugeau, M. Bertalmio, V. Caselles, and G. Sapiro, "A comprehensive framework for image inpainting," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2634–2645, Oct. 2010, doi: 10.1109/TIP.2010.2049240.
41. J. F. Aujol, S. Ladjal, and S. Masnou, "Exemplar-Based Inpainting from a Variational Point of View," <https://doi.org/10.1137/080743883>, vol. 42, no. 3, pp. 1246–1285, May 2010, doi: 10.1137/080743883.
42. W. Casaca, M. Boaventura, M. P. De Almeida, and L. G. Nonato, "Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textured images," *Pattern Recognit Lett*, vol. 36, no. 1, pp. 36–45, Jan. 2014, doi: 10.1016/J.PATREC.2013.08.023.
43. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 2536–2544, Dec. 2016, doi: 10.1109/CVPR.2016.278.
44. Z. Qin, Q. Zeng, Y. Zong, and F. Xu, "Image inpainting based on deep learning: A review," *Displays*, vol. 69, p. 102028, Sep. 2021, doi: 10.1016/J.DISPLA.2021.102028.
45. R. Suvorov et al., "Resolution-robust Large Mask Inpainting with Fourier Convolutions," *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, pp. 3172–3182, 2022, doi: 10.1109/WACV51458.2022.00323.
46. Z. Lu, J. Jiang, J. Huang, G. Wu, and X. Liu, "GLaMa: Joint Spatial and Frequency Loss for General Image Inpainting," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2022-June, pp. 1300–1309, May 2022, doi: 10.1109/CVPRW56347.2022.00136.
47. P. Shamsolmoali, M. Zareapoor, and E. Granger, "Image Completion Via Dual-Path Cooperative Filtering," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, Jun. 2023, doi: 10.1109/ICASSP49357.2023.10097260.
48. Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild *," 2015, doi: 10.1109/ICCV.2015.425.
49. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018, doi: 10.1109/TPAMI.2017.2723009.

50. A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "RePaint: Inpainting using Denoising Diffusion Probabilistic Models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 11451–11461, Jan. 2022, doi: 10.1109/CVPR52688.2022.01117.
51. R. K. Cho, K. Sood, and C. S. C. Channapragada, "Image Repair and Restoration Using Deep Learning," *AIST 2022 - 4th International Conference on Artificial Intelligence and Speech Technology*, 2022, doi: 10.1109/AIST55798.2022.10065203.
52. Y. Chen, R. Xia, K. Yang, and K. Zou, "DGCA: high resolution image inpainting via DR-GAN and contextual attention," *Multimed Tools Appl*, pp. 1–21, May 2023, doi: 10.1007/S11042-023-15313-0/FIGURES/10.
53. P. Jeevan, D. S. Kumar, and A. Sethi, "WavePaint: Resource-efficient Token-mixer for Self-supervised Inpainting," Jul. 2023, Accessed: Sep. 10, 2023. [Online]. Available: <https://arxiv.org/abs/2307.00407v1>
54. P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 12868–12878, 2021, doi: 10.1109/CVPR46437.2021.01268.
55. H. Farid, "Image forgery detection," *IEEE Signal Process Mag*, vol. 26, no. 2, pp. 16–25, 2009, doi: 10.1109/MSP.2008.931079.
56. M. Zanardelli, F. Guerrini, R. Leonardi, and N. Adami, "Image forgery detection: a survey of recent deep-learning approaches," *Multimed Tools Appl*, vol. 82, no. 12, pp. 17521–17566, May 2022, doi: 10.1007/S11042-022-13797-W/TABLES/6.
57. N. T. Pham and C. S. Park, "Toward Deep-Learning-Based Methods in Image Forgery Detection: A Survey," *IEEE Access*, vol. 11, pp. 11224–11237, 2023, doi: 10.1109/ACCESS.2023.3241837.
58. H. Li, W. Luo, and J. Huang, "Localization of Diffusion-Based Inpainting in Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3050–3064, Dec. 2017, doi: 10.1109/TIFS.2017.2730822.
59. Y. Zhang, F. Ding, S. Kwong, and G. Zhu, "Feature pyramid network for diffusion-based image inpainting detection," *Inf Sci (N Y)*, vol. 572, pp. 29–42, Sep. 2021, doi: 10.1016/J.INS.2021.04.042.
60. Y. Zhang, T. Liu, C. Cattani, Q. Cui, and S. Liu, "Diffusion-based image inpainting forensics via weighted least squares filtering enhancement," *Multimed Tools Appl*, vol. 80, no. 20, pp. 30725–30739, Aug. 2021, doi: 10.1007/S11042-021-10623-7/TABLES/9.
61. A. K. Al-Jaberi, A. Asaad, S. A. Jassim, and N. Al-Jawad, "Topological Data Analysis for Image Forgery Detection," *Indian Journal of Forensic Medicine & Toxicology*, vol. 14, no. 3, pp. 1745–1751, Jul. 2020, doi: 10.37506/IJFMT.V14I3.10668.
62. V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY An Evaluation of Popular Copy-Move Forgery Detection Approaches", Accessed: Feb. 06, 2022. [Online]. Available: <http://www5.cs.fau.de/our-team>.
63. Q. Wu, S. J. Sun, W. Zhu, G. H. Li, and D. Tu, "Detection of digital doctoring in exemplar-based inpainted images," *Proceedings of the 7th International Conference on Machine Learning and Cybernetics, ICMLC*, vol. 3, pp. 1222–1226, 2008, doi: 10.1109/ICMLC.2008.4620591.
64. I. C. Chang, J. C. Yu, and C. C. Chang, "A forgery detection algorithm for exemplar-based inpainting images using multi-region relation," *Image Vis Comput*, vol. 31, no. 1, pp. 57–71, Jan. 2013, doi: 10.1016/J.IMAVIS.2012.09.002.
65. Z. Liang, G. Yang, X. Ding, and L. Li, "An efficient forgery detection algorithm for object removal by exemplar-based image inpainting," *J Vis Commun Image Represent*, vol. 30, pp. 75–85, Jul. 2015, doi: 10.1016/J.JVCIR.2015.03.004.
66. K. S. Bacchuwar, Aakashdeep, and K. R. Ramakrishnan, "A jump patch-block match algorithm for multiple forgery detection," *Proceedings - 2013 IEEE International Multi Conference on Automation, Computing, Control, Communication and Compressed Sensing, iMac4s 2013*, pp. 723–728, 2013, doi: 10.1109/IMAC4S.2013.6526502.
67. J. Wang, K. Lu, D. Pan, N. He, and B. kun Bao, "Robust object removal with an exemplar-based image inpainting approach," *Neurocomputing*, vol. 123, pp. 150–155, Jan. 2014, doi: 10.1016/J.NEUCOM.2013.06.022.
68. D. Zhang, Z. Liang, G. Yang, Q. Li, L. Li, and X. Sun, "A robust forgery detection algorithm for object removal by exemplar-based image inpainting," *Multimed Tools Appl*, vol. 77, pp. 11823–11842, 2018, doi: 10.1007/s11042-017-4829-0.
69. J. C. Lee, "Copy-move image forgery detection based on Gabor magnitude," *J Vis Commun Image Represent*, vol. 31, pp. 320–334, Aug. 2015, doi: 10.1016/J.JVCIR.2015.07.007.
70. X. Jin, Y. Su, L. Zou, Y. Wang, P. Jing, and Z. J. Wang, "Sparsity-based image inpainting detection via canonical correlation analysis with low-rank constraints," *IEEE Access*, vol. 6, pp. 49967–49978, Aug. 2018, doi: 10.1109/ACCESS.2018.2866089.

71. G. Mahfoudi, F. Morain-Nicolier, F. Retraint, and M. Pic, "Object-Removal Forgery Detection through Reflectance Analysis," *2020 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2020*, Dec. 2020, doi: 10.1109/ISSPIT51521.2020.9408870.
72. P. Kakar, N. Sudha, and W. Ser, "Exposing digital image forgeries by detecting discrepancies in motion blur," *IEEE Trans Multimedia*, vol. 13, no. 3, pp. 443–452, Jun. 2011, doi: 10.1109/TMM.2011.2121056.
73. H. R. Chennamma and L. Rangarajan, "Image Splicing Detection Using Inherent Lens Radial Distortion," *IJCSI International Journal of Computer Science Issues*, vol. 7, pp. 1694–0814, May 2011, Accessed: Jun. 05, 2023. [Online]. Available: <https://arxiv.org/abs/1105.4712v1>
74. J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," <https://doi.org/10.1117/12.640109>, vol. 6072, pp. 362–372, Feb. 2006, doi: 10.1117/12.640109.
75. B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image Vis Comput*, vol. 27, no. 10, pp. 1497–1503, Sep. 2009, doi: 10.1016/J.IMAVIS.2009.02.001.
76. P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012, doi: 10.1109/TIFS.2012.2202227.
77. L. Shen, G. Yang, L. Li, and X. Sun, "Robust detection for object removal with post-processing by exemplar-based image inpainting," *ICNC-FSKD 2017 - 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 2730–2736, Jun. 2018, doi: 10.1109/FSKD.2017.8393211.
78. X. Zhu, Y. Qian, X. Zhao, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Process Image Commun*, vol. 67, pp. 90–99, Sep. 2018, doi: 10.1016/J.IMAGE.2018.05.015.
79. H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 8300–8309, Oct. 2019, doi: 10.1109/ICCV.2019.00839.
80. M. Lu and S. Niu, "A Detection Approach Using LSTM-CNN for Object Removal Caused by Exemplar-Based Image Inpainting," *Electronics 2020, Vol. 9, Page 858*, vol. 9, no. 5, p. 858, May 2020, doi: 10.3390/ELECTRONICS9050858.
81. N. Kumar and T. Meenpal, "Semantic segmentation-based image inpainting detection," *Lecture Notes in Electrical Engineering*, vol. 661, pp. 665–677, 2021, doi: 10.1007/978-981-15-4692-1_51/COVER.
82. A. Li et al., "Noise Doesn't Lie: Towards Universal Detection of Deep Inpainting," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 786–792, Jun. 2021, doi: 10.24963/ijcai.2021/109.
83. X. Zhu, J. Lu, H. Ren, H. Wang, and B. Sun, "A transformer-CNN for deep image inpainting forensics," *Visual Computer*, pp. 1–15, Aug. 2022, doi: 10.1007/S00371-022-02620-0/TABLES/5.
84. Y. Zhang, Z. Fu, S. Qi, M. Xue, Z. Hua, and Y. Xiang, "Localization of Inpainting Forgery With Feature Enhancement Network," *IEEE Trans Big Data*, Jun. 2022, doi: 10.1109/TBDATA.2022.3225194.
85. H. Wu and J. Zhou, "IID-Net: Image Inpainting Detection Network via Neural Architecture Search and Attention," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1172–1185, Mar. 2022, doi: 10.1109/TCSVT.2021.3075039.
86. B. Bayar and M. C. Stamm, "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, Nov. 2018, doi: 10.1109/TIFS.2018.2825953.
87. P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning Rich Features for Image Manipulation Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1053–1061, Dec. 2018, doi: 10.1109/CVPR.2018.00116.
88. Y. Zhou, H. Wang, Q. Zeng, R. Zhang, and S. Meng, "A Discriminative Multi-Channel Noise Feature Representation Model for Image Manipulation Localization," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, Jun. 2023, doi: 10.1109/ICASSP49357.2023.10096501.
89. X. Liu, Y. Liu, J. Chen, and X. Liu, "PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7505–7517, Nov. 2022, doi: 10.1109/TCSVT.2022.3189545.
90. J. Wang et al., "Deep High-Resolution Representation Learning for Visual Recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021, doi: 10.1109/TPAMI.2020.2983686.
91. Y. Wu, W. Abdalimageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9535–9544, Jun. 2019, doi: 10.1109/CVPR.2019.00977.
92. X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "SPAN: Spatial Pyramid Attention Network for Image Manipulation Localization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12366 LNCS, pp. 312–328, 2020, doi: 10.1007/978-3-030-58589-1_19/COVER.

93. H. Sharma, N. Kanwal, and R. S. Batth, "An Ontology of Digital Video Forensics: Classification, Research Gaps & Datasets," *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, pp. 485–491, Dec. 2019, doi: 10.1109/ICCIKE47802.2019.9004331.
94. J. Zhang, Y. Su, and M. Zhang, "Exposing digital video forgery by ghost shadow artifact," *1st ACM Workshop on Multimedia in Forensics - MiFor'09, Co-located with the 2009 ACM International Conference on Multimedia, MM'09*, pp. 49–53, 2009, doi: 10.1145/1631081.1631093.
95. S. L. Das Gopu Darsan Shreyas Divya Devan, "Blind Detection Method for Video Inpainting Forgery," *Int J Comput Appl*, vol. 60, no. 11, pp. 975–8887, 2012.
96. C. S. Lin and J. J. Tsay, "A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis," *Digit Investig*, vol. 11, no. 2, pp. 120–140, 2014, doi: 10.1016/J.DIIN.2014.03.016.
97. S. Bai, H. Yao, R. Ni, and Y. Zhao, "Detection and Localization of Video Object Removal by Spatio-Temporal LBP Coherence Analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11903 LNCS, pp. 244–254, 2019, doi: 10.1007/978-3-030-34113-8_21/COVER.
98. M. Aloraini, M. Sharifzadeh, and D. Schonfeld, "Sequential and Patch Analyses for Object Removal Video Forgery Detection and Localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 917–930, Mar. 2021, doi: 10.1109/TCSVT.2020.2993004.
99. M. Aloraini, M. Sharifzadeh, C. Agarwal, and D. Schonfeld, "Statistical sequential analysis for object-based video forgery detection," *IS and T International Symposium on Electronic Imaging Science and Technology*, vol. 2019, no. 5, p. 543, Jan. 2019, doi: 10.2352/ISSN.2470-1173.2019.5.MWSF-543.
100. X. Jin, Z. He, J. Xu, Y. Wang, and Y. Su, "Object-Based Video Forgery Detection via Dual-Stream Networks," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6. doi: 10.1109/ICME51207.2021.9428319.
101. A. V. Subramanyam and S. Emmanuel, "Video forgery detection using HOG features and compression properties," *2012 IEEE 14th International Workshop on Multimedia Signal Processing, MMSP 2012 - Proceedings*, pp. 89–94, 2012, doi: 10.1109/MMSP.2012.6343421.
102. R. Dixit and R. Naskar, "Review, analysis and parameterisation of techniques for copy-move forgery detection in digital images," *IET Image Process*, vol. 11, no. 9, pp. 746–759, Sep. 2017, doi: 10.1049/IET-IPR.2016.0322.
103. G. Kaur, N. Singh, and M. Kumar, *Image forgery techniques: a review*, no. May. Springer Netherlands, 2022. doi: 10.1007/s10462-022-10211-7.
104. S. Teerakanok and T. Uehara, "Copy-Move Forgery Detection: A State-of-the-Art Technical Review and Analysis," *IEEE Access*, vol. 7, pp. 40550–40568, 2019, doi: 10.1109/ACCESS.2019.2907316.
105. 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 PART 2, pp. 1099–1110, Sep. 2011, doi: 10.1109/TIFS.2011.2129512.
106. "CoMoFoD — New database for copy-move forgery detection | IEEE Conference Publication | IEEE Xplore." <https://ieeexplore.ieee.org/document/6658316> (accessed Jun. 06, 2023).
107. J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," *2013 IEEE China Summit and International Conference on Signal and Information Processing, ChinaSIP 2013 - Proceedings*, pp. 422–426, 2013, doi: 10.1109/CHINASIP.2013.6625374.
108. N. T. Pham, J.-W. Lee, G.-R. Kwon, and C.-S. Park, "Hybrid Image-Retrieval Method for Image-Splicing Validation", doi: 10.3390/sym11010083.
109. B. Wen, Y. Zhu, R. Subramanian, T. T. Ng, X. Shen, and S. Winkler, "COVERAGE - A novel database for copy-move forgery detection," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-August, pp. 161–165, Aug. 2016, doi: 10.1109/ICIP.2016.7532339.
110. P. Korus and J. Huang, "Multi-Scale Fusion for Improved Localization of Malicious Tampering in Digital Images," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1312–1326, Mar. 2016, doi: 10.1109/TIP.2016.2518870.
111. P. Korus and J. Huang, "Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 809–824, Apr. 2017, doi: 10.1109/TIFS.2016.2636089.
112. D. T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE - A raw images dataset for digital image forensics," *Proceedings of the 6th ACM Multimedia Systems Conference, MMSys 2015*, pp. 219–224, Mar. 2015, doi: 10.1145/2713168.2713194.
113. H. Guan *et al.*, "MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation," *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2019*, pp. 63–72, Feb. 2019, doi: 10.1109/WACVW.2019.00018.
114. H. Guan *et al.*, "NISTIR 8377 User Guide for NIST Media Forensic Challenge (MFC) Datasets", doi: 10.6028/NIST.IR.8377.

115. G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and M. Pic, "Defacto: Image and face manipulation dataset," *European Signal Processing Conference*, vol. 2019-September, Sep. 2019, doi: 10.23919/EUSIPCO.2019.8903181.
116. M. Daisy, P. Buysens, D. Tschumperle, and O. Lezoray, "A smarter exemplar-based inpainting algorithm using local and global heuristics for more geometric coherence," *2014 IEEE International Conference on Image Processing, ICIP 2014*, pp. 4622–4626, Jan. 2014, doi: 10.1109/ICIP.2014.7025937.
117. T. Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, May 2014, doi: 10.1007/978-3-319-10602-1_48.
118. A. Novozamsky, S. Saic, and B. Mahdian, "IMD2020: A Large-Scale Annotated Dataset Tailored for Detecting Manipulated Images," *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2020*, pp. 71–80, Mar. 2020, doi: 10.1109/WACVW50321.2020.9096940.
119. J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative Image Inpainting with Contextual Attention," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 5505–5514, Jan. 2018, doi: 10.1109/CVPR.2018.00577.
120. R. Benenson, G. Research, and V. Ferrari, "From colouring-in to pointillism: revisiting semantic segmentation supervision," Oct. 2022, Accessed: Sep. 25, 2023. [Online]. Available: <https://arxiv.org/abs/2210.14142v2>
121. A. Newson, A. Almansa, Y. Gousseau, and P. Pérez, "Non-Local Patch-Based Image Inpainting," *Image Processing On Line*, vol. 7, pp. 373–385, Dec. 2017, doi: 10.5201/IPOL.2017.189.
122. The GIMP Development Team. *GIMP*. Retrieved from <https://www.gimp.org>
123. W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "MAT: Mask-Aware Transformer for Large Hole Image Inpainting," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2022-June, pp. 10748–10758, Jun. 2022, doi: 10.1109/CVPR52688.2022.01049.
124. H. Wu, Y. Chen, and J. Zhou, "Rethinking Image Forgery Detection via Contrastive Learning and Unsupervised Clustering," Aug. 2023, Accessed: Sep. 28, 2023. [Online]. Available: <https://arxiv.org/abs/2308.09307v1>

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.