electronics

MDPI

Article

UFCC: A Unified Forensic Approach to Locating Tampered Areas in Still Images and Detecting Deepfake Videos by Evaluating Content Consistency

Po-Chyi Su [1,*], Bo-Hong Huang [1] and Tien-Ying Kuo [2,*]

[1] Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan
[2] Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
* Correspondence: pochyisu@csie.ncu.edu.tw (P.-C.S.); tykuo@ntut.edu.tw (T.-Y.K.); Tel.: +886-3-4227151 (ext. 35314) (P.-C.S.); +886-2-27712171 (ext. 2117) (T.-Y.K.)

Abstract: Image inpainting and Deepfake techniques have the potential to drastically alter the meaning of visual content, posing a serious threat to the integrity of both images and videos. Addressing this challenge requires the development of effective methods to verify the authenticity of investigated visual data. This research introduces UFCC (Unified Forensic Scheme by Content Consistency), a novel forensic approach based on deep learning. UFCC can identify tampered areas in images and detect Deepfake videos by examining content consistency, assuming that manipulations can create dissimilarity between tampered and intact portions of visual data. The term "Unified" signifies that the same methodology is applicable to both still images and videos. Recognizing the challenge of collecting a diverse dataset for supervised learning due to various tampering methods, we overcome this limitation by incorporating information from original or unaltered content in the training process rather than relying solely on tampered data. A neural network for feature extraction is trained to classify imagery patches, and a Siamese network measures the similarity between pairs of patches. For still images, tampered areas are identified as patches that deviate from the majority of the investigated image. In the case of Deepfake video detection, the proposed scheme involves locating facial regions and determining authenticity by comparing facial region similarity across consecutive frames. Extensive testing is conducted on publicly available image forensic datasets and Deepfake datasets with various manipulation operations. The experimental results highlight the superior accuracy and stability of the UFCC scheme compared to existing methods.

Keywords: media forensics; image tampering; deepfake; Siamese network; deep learning

check for updates

Citation: Su, P.-C.; Huang, B.-H.; Kuo, T.-Y. UFCC: A Unified Forensic Approach to Locating Tampered Areas in Still Images and Detecting Deepfake Videos by Evaluating Content Consistency. *Electronics* **2024**, *13*, 804. https://doi.org/10.3390/electronics13040804

Academic Editor: Jitae Shin

Received: 10 January 2024
Revised: 5 February 2024
Accepted: 16 February 2024
Published: 19 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The widespread use of digital cameras and smartphones enables individuals to effortlessly collect vast amounts of high-resolution imagery data. The accessibility of editing tools further empowers users to manipulate these visual data at will, offering unprecedented convenience, especially in the era of social media dominance. However, this ease of digital content modification poses a significant threat to the integrity and authenticity of media. Malicious users may exploit this vulnerability by tampering with images or videos and disseminating them across social platforms, not only deceiving netizens but also manipulating public opinions. The emergence of Deepfake technology further raises these concerns, allowing malicious users to synthesize facial images without sophisticated processing. This simplicity in content modification may lead to serious consequences, including the spread of fake news, cyberbullying, and revenge pornography. The forged content, once unleashed publicly, can spiral out of control, causing irreparable damage to individuals' reputations and even resulting in physical and psychological harm. In light of these challenges, there is an urgent need to develop media forensic techniques that can

verify the authenticity of digital images and videos, safeguarding against the potential misuse and harmful consequences of manipulated content.

The goals of identifying image inpainting and Deepfake videos differ slightly. Image inpainting typically involves altering a smaller section of a static image to conceal errors, or to maliciously alter its meaning. In this research, the focus is on the latter aspect, aiming to spot potential tampering in a picture. Consequently, the detection not only highlights the presence of tampering but also locates the affected regions within the image. For instance, if the detection process reveals the addition of a person or the removal of a subject, we can gain insights into the image editor's intentions by examining these manipulated areas. On the other hand, Deepfake techniques primarily focus on modifying a person's face to adopt their identity in deceptive videos for conveying misinformation. The objective in detecting Deepfake is to ascertain whether the video is artificially generated rather than authentic. If an examined video is identified as a result of applying Deepfake manipulation, it may lead to the disregard of its content.

It is worth mentioning that numerous current forgery detection methods resort to supervised learning and adopt deep learning strategies. To train robust deep learning models, a substantial amount of both unaltered and forged data is usually required. However, obtaining a comprehensive dataset encompassing both forged data and their original counterparts is impractical. The constant evolution of forgery techniques makes it more difficult to anticipate the specific manipulations applied to an image or video. In other words, it is challenging to ensure the generality of trained deep learning models to deal with various content manipulations by assuming them in advance.

This research introduces UFCC (Unified Forensic Scheme by Content Consistency), a unified forensic scheme designed to detect both image inpainting and Deepfake videos. The core concept revolves around scrutinizing images or video frames for the presence of "abnormal" content discontinuity. A convolutional neural network (CNN) is trained to extract features from patches within images. Subsequently, a Siamese network is employed to evaluate the consistency by assessing the similarity between pairs of image patches. While this proposed method utilizes deep learning techniques, different from many existing methods, it uniquely requires only original or unaltered image patches during the model training phase. This innovation allows for the identification of forged regions without a dependency on specific forgery datasets, offering a more adaptable and practical solution for detecting forgery in imagery data.

The term "Unified" in UFCC indicates that the identical methodology can be utilized for detecting both image manipulation and Deepfake videos. In the realm of manipulated images, the initial step involves identifying potential regions, succeeded by a more detailed localization of forged areas using a deep segmentation network. This process enables a more precise delineation of the regions affected by image manipulation operations. In the cases of detecting Deepfake videos, we first extract facial regions within video frames and subsequently assess the similarity of these facial regions across adjacent frames. The proposed UFCC can thus prove equally adept at detecting both image forgery and Deepfake videos. The contributions of this research are summarized below:

1. The camera-type classification models are improved and extended to apply to digital image forensics and Deepfake detection.
2. A suitable dataset is formed for classifying camera types.
3. A unified method is proposed for effectively evaluating image and video content consistency.
4. A unified method is developed to deal with various Deepfake approaches.

The rest of the paper is organized as follows. Section 2 introduces related work, including traditional signal processing methods and modern deep learning methods. Section 3 details the research methodology, including the adopted deep learning network architecture and related data preparation. Section 4 presents the details of model training, showcases the results, and provides comparisons with existing work. Section 5 provides conclusive remarks and outlines expectations for future developments.

2. Related Work

2.1. Digital Image Forensics

The widespread availability of image editing software, equipped with increasingly potent capabilities, amplifies the challenges from image forgery, prompting an urgent demand for image forensic techniques. Two primary approaches, active and passive, address the detection of image content forgery or manipulation. Active methods involve embedding imperceptible digital watermarks into images, where the extracted watermarks or their specific conditions may reveal potential manipulations applied to an investigated image. However, drawbacks include the protection of only watermarked images, and the content being somewhat affected by the watermark embedding process. Moreover, controversies surround the responsibility for detecting or extracting watermarks to establish image authenticity.

In contrast, passive methods operate on the premise that even seemingly imperceptible image manipulations alter the statistical characteristics of imagery data. Rather than introducing additional signals into images, passive methods seek to uncover underlying inconsistencies to detect image manipulations or forgery [1,2]. Among passive methods, camera model recognition [3] is a promising direction in image forensics. This involves determining the type of camera used to capture the images. Various recognition methods utilizing features such as CFA demosaic effects [4], sensor noise patterns [5], local binary patterns [6], noise models [7], chromatic aberrations [8], and illumination direction [9] have been proposed. Classification based on these features helps to determine the camera model types. Recent work has utilized deep learning [10] to pursue generality.

Despite the effectiveness demonstrated by camera model identification methods, a prevalent limitation is evident in many of these techniques. They operate under the assumption that the camera models to be determined must be part of a predefined dataset, requiring a foundational understanding of these models included in the training data to precisely recognize images captured by specific cameras. However, selecting a suitable set of camera models is not a trivial issue. Expanding the training dataset to encompass all camera types is also impractical, given the fact that the number of camera types continues to grow over time. In image forensics, identifying the exact camera model responsible for capturing an image in various investigative contexts is not necessary as the primary objective here is to confirm whether a set of examined images are from the same camera model to expose instances of intellectual property infringement or to reveal the presence of image splicing fraud [11]. Bondi et al. [12] discovered that CNN-based camera model identification can be employed for classifying and localizing image splicing operations. Ref. [13] fed features extracted by the network into a Siamese network for further comparison. Building upon these findings, the proposed UFCC scheme aims to extend the ideas presented by [13] to develop a unified approach for detecting manipulated image regions and determining the authenticity of videos.

2.2. Deepfake

Deepfake operations are frequently utilized for facial manipulation, changing the visual representation of individuals in videos to diverge from the original content or substitute their identities. The spectrum of Deepfake operations has expanded over time and elicited numerous concerns. Next, we analyze the strengths and weaknesses of each Deepfake technique and track the evolution of these approaches.

1. Identity swapping

Identity swapping entails substituting faces in a source image with the face of another individual, effectively replacing the facial features of the target person while retaining the original facial expressions. The use of deep learning methods for identity replacement can be traced back to the emergence of Deepfakes in 2017 [14]. Deepfakes employed an autoencoder architecture comprising an encoder–decoder pair, where the encoder extracts latent facial features and the decoder reconstructs the target face. Korshunova et al. [15]

utilized a fully convolutional network along with style transfer techniques, and adopted multiple loss functions with variation regularization to generate realistic images. However, these approaches necessitate a substantial amount of data for both the source and target individuals for paired training, making the training process time-consuming.

In an effort to enhance the efficiency of Deepfakes, Zakharov et al. [16] proposed GAN-based few-shot or one-shot learning to generate realistic talking-head videos from images. Various studies have focused on extensive meta-learning using large-scale video datasets over extended periods. Additionally, self-supervised learning methods and the generation of forged identities based on independently encoded facial features and annotations have been explored. Zhu et al. [17] extended the latent spaces to preserve more facial details and employed StyleGAN2 [18] to generate high-resolution swapped facial images.

2. Expression reenactment

Expression reenactment involves transferring the facial expressions, gestures, and head movements of the source person onto the target person while preserving the identity of the target individual. These operations aim to modify facial expressions while synchronizing lip movements to create fictional content. Techniques such as 3D face reconstruction and GAN architectures were employed to capture head geometry and motions. Thies et al. [19] introduced 3D facial modeling combined with image rendering, allowing for the real-time transfer of spoken expressions captured by a regular web camera to the face of the target person.

While GAN-based methods can generate realistic images, achieving highly convincing reenactment for unknown identities requires substantial training data. Kim et al. [20] proposed fusing spatial–temporal encoding and conditional GANs (cGANs) in static images to synthesize target video avatars, incorporating head poses, facial expressions, and eye movements, resulting in highly realistic scenes. Other research has explored fully unsupervised methods utilizing dual cGANs to train emotion–action units for generating facial animations from single images.

Recent advancements include few-shot or one-shot facial expression reenactment techniques, alleviating the training burden on large-scale datasets. These approaches adopt strategies such as image attention analysis, target feature alignment, and landmark transformation to prevent quality degradation due to limited or mismatched data. Such methods eliminate the need for additional identity-adaptive fine-tuning, making them suitable for the practical applications of Deepfake. Fried et al. [21] devised content-based editing methods to fabricate forged speech videos, modifying speakers' head movements to match the dialogue content.

3. Face synthesis

Facial synthesis primarily revolves around the creation of entirely new facial images and finds applications in diverse domains such as video games and 3D modeling. Many facial synthesis methods leverage GAN models to enhance resolution, image quality, and realism. StyleGAN [22] elevated the image resolution initially introduced by ProGAN [23], and subsequent improvements were made with StyleGAN2 [18], which effectively eliminated artifacts to further enhance image quality.

The applications of GAN-based facial synthesis methods are varied, encompassing facial attribute translation [22,24], the combination of identity and attributes, and the removal of specific features. Some synthesis techniques extend to virtual makeup trials, enabling consumers to virtually test cosmetics [25] without the need for physical samples or in-person visits. Additionally, these methods can be applied to synthesis operations involving the entire body; for instance, DeepNude utilized the Pix2PixHD GAN model [26] to patch clothing areas and generate fabricated nude images.

4. Facial attribute manipulation

Facial attribute manipulation, also known as facial editing or modification, entails modifying facial attributes like hair color, hairstyle, skin tone, gender, age, smile, glasses, and makeup. These operations can be considered as a form of conditional partial facial

synthesis. GAN methods, commonly used for facial synthesis, are also employed for facial attribute manipulation. Choi et al. [24] introduced a unified model that simultaneously trains multiple datasets with distinct regions of interest, allowing for the transfer of various facial attributes and expressions. This approach eliminates the need for additional cross-domain models for each attribute. Extracted facial features can be analyzed in different latent spaces, providing more precise control over attribute manipulation within facial editing. However, it is worth noting that the performance may degrade when dealing with occluded faces or when the face lies outside the expected range.

5. Hybrid approaches

A potential trend is emerging wherein different Deepfake techniques are amalgamated to form hybrid approaches, rendering them more challenging to identify. Nirkin et al. [27] introduced a GAN model for real-time face swapping, incorporating a fusion of reenactment and synthesis. Some approaches may involve the use of two separate Variational Autoencoders (VAEs) to convert facial features into latent vectors. These vectors are then conditionally adjusted for the target identity. These methods facilitate the application of multiple operations to any combination of two faces without requiring retraining. Users have the flexibility to freely swap faces and modify facial parameters, including age, gender, smile, hairstyle, and more. More advanced and sophisticated methods exist. [28] adopted a multimodal fusion approach to deal with fake news to provide further protection mechanisms for social media platforms.

2.3. Deepfake Detection

In response to the threat brought by Deepfake, several potential solutions have been proposed to aid in identifying whether an examined video has undergone Deepfake operations.

1. Frame-level detection

Traditional Deepfake classifiers are typically trained directly on both real and manipulated images or video frames, employing methods such as dual-stream neural networks [29], MesoNet [30], CapsuleNet [31], and Xception-Net [32]. However, with the increasing diversity of recent Deepfake methods, direct training on images or video frames is deemed insufficient to handle the wide variety of manipulation techniques. It has been noted that certain details in manipulated images or video frames can still exhibit flaws, including decreased image quality, abnormal content deformations, or fragmented edges. Some methods have leveraged the inconsistencies arising from imperfections in Deepfake images/videos to analyze biological cues such as blinking, head poses, skin textures, iris patterns, teeth colors, etc. Additionally, abnormalities in facial deformation or mouth movements may be utilized to distinguish between real and fake instances. Others decompose facial images to extract details and combine them with the original face to identify key clues. Frequency domain approaches have also been explored. Li et al. [33] developed an adaptive feature extraction module to enhance separability in the embedding space. Liu et al. [34] combined spatial images and phase spectra to address under-sampling artifacts and enhance detection effectiveness.

Attention or segmentation are common strategies in Deepfake classifiers. For instance, attention mechanisms may be employed to highlight key regions for forming improved feature maps in classification tasks or focusing on blended regions to identify manipulated faces without relying on specific facial operations. Nguyen et al. [35] designed a multi-task learning network to locate manipulated areas in forged facial images simultaneously.

2. Video-level detection

The objective is to ascertain whether an entire video comprises manipulated content rather than merely detecting individual frames. These methods typically utilize the temporal information and dynamic features of the video to distinguish between authentic and manipulated content. Approaches relying on temporal consistency determination analyze the temporal relationships and motion patterns among video frames. Güera et al. [36] employed a blend of CNNs and recurrent neural networks (RNNs), where CNNs extract frame

features and RNNs identify temporal inconsistencies. Motion-feature-based methods examine motion patterns and dynamic features within videos, as real and manipulated videos may display distinct motion behaviors. Analyzing motion patterns, optical flows, and motion consistency helps to identify abnormal patterns in manipulated videos. Refs. [37,38] utilized blinking and head pose changes to discern between real and manipulated content. Spatial-feature-based methods utilize texture structures, spectral features, edge sharpening, and other spatial characteristics of video frames to detect potential synthesis artifacts in manipulated videos. It is essential to note that video-level Deepfake detection methods may demand increased computational resources and execution time. Moreover, due to the continuous evolution of Deepfake techniques, detection methods need regular updates to counter new manipulation attacks.

3. Proposed Scheme

This section offers in-depth insights into the UFCC scheme. Section 3.1 presents the system and architecture diagram, while Sections 3.2–3.4 delve into the network design and the mechanism for addressing image tampering. Section 3.5 elaborates on how the same methodology is employed in assessing the authenticity of Deepfake videos.

3.1. System Architecture

The proposed UFCC scheme is designed based on assessing the similarity between local areas in images or video frames. Figure 1 depicts a unified system architecture designed for detecting image inpainting and Deepfake videos. The initial step involves compiling a dataset of images and their respective camera types. Potential duplicates are eliminated. Images or video frames are segmented into patches, and a subset is selected for training the feature extractors. In the detection process for image forgery or inpainting, a target patch is chosen and compared with others to determine their "similarity". Specifically, if two patches are identified as being of the same camera type, they are deemed "similar". The evaluation of similarity is conducted using a Siamese network. Patches identified as distinct from others can help to identify areas affected by image manipulation. The initial detection results are further refined to obtain more precise information about tampered regions.

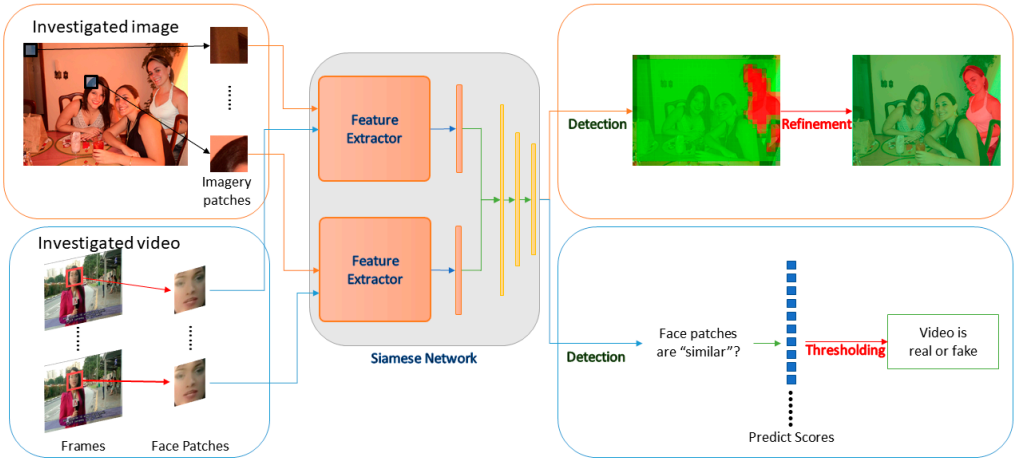


Figure 1. The architecture diagram of the proposed UFCC scheme.

The distinction between static image manipulation detection and Deepfake video detection stems from the nature of the tampered areas identified. Image manipulation detection identifies tampered regions that can be anywhere within an image. On the other hand, Deepfake video detection specifically targets facial areas, given that most Deepfake methods alter only faces within frames. In videos, we utilize frame-by-frame patch comparison to assess whether the faces appearing in target frames are fabricated. This approach takes into account the continuity of visual content, as Deepfake operations

are frequently applied individually to frames, without ensuring consistency in content across adjacent frames.

3.2. Feature Extractor

The goal of the feature extractor depicted in Figure 1 diverges from that of typical CNNs, which often focus on learning to recognize content like people, cars, animals, etc., in images. The scheme is designed to identify subtle signals specific to camera types, signals unrelated to imagery content. To achieve this, the model focuses on high-frequency features through filtering. Bayer [39] introduced the concept of constrained convolutions for extracting high-frequency features. During training, the center of the prediction error filter is consistently set to -1, while the surrounding points are normalized to ensure their sum equals 1. Figure 2 illustrates the design of the feature extraction networks, including detailed parameter settings. The input comprises patches of the investigated image with dimensions set as $128 \times 128 \times 3$ (width $\times$ height $\times$ channels). The constrained convolution layer serves as the initial convolution layer, followed by four convolution blocks with similar structures. Each block includes a convolutional layer, Batch Normalization, ReLU, and max pooling. It is worth noting that all convolution layers have a stride of one and may utilize boundary reflection in their convolution operations. For a clearer understanding, the resulting dimensions of the data and the numbers of parameters in the convolution steps are also listed in Figure 2. Two fully connected layers are added at the end to expedite convergence. The output provides the camera type of the patch, and the Softmax function is applied at the end. Squeeze and excitation layers are inserted between the second and third convolution blocks. These layers aid in accelerating the model's convergence and adaptively learning to determine the importance of individual feature channels, assigning higher weights to channels of primary interest.

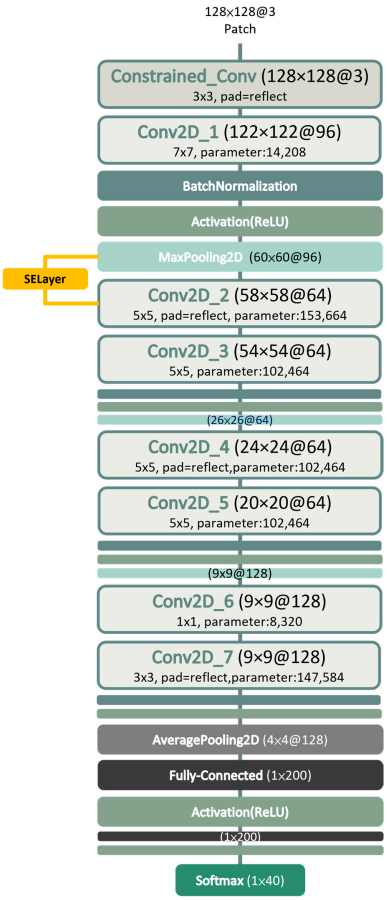


Figure 2. The feature extractor in the proposed UFCC scheme.

3.3. Similarity Network

Our feature extraction classifier demonstrates the ability to recognize known camera types or categories included in the training process. However, its performance may vary when encountering camera types not present in the original training dataset. Establishing a more generalized classification approach would require additional camera-type data for retraining. Unfortunately, obtaining an extensive set of camera types is impractical, given the expectation of new ones emerging. Instead of relying solely on camera type classification, it is more appropriate for image forgery detection to ascertain whether two examined image patches originate from cameras of the same type. Taking inspiration from the work of Huh [40] and Mayer [13], the Siamese network, as depicted in Figure 3, is well suited for conducting similarity comparisons between two inputs. A brief overview of the Siamese network concept is provided below.

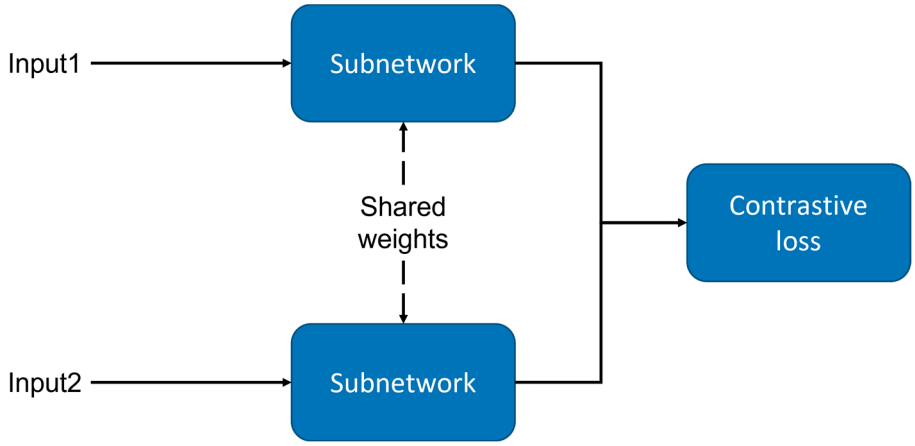


Figure 3. Siamese network schematic diagram.

3.4. Similarity Network

The feature extraction classifier we have designed exhibits the capability to discern the known camera types or categories considered during the training process. However, the outcome may be inconsistent when confronted with the camera types absent from the original training dataset. Deriving a generalized classification approach would necessitate additional camera-type data for retraining. Unfortunately, acquiring an extensive set of camera types is impractical since we can expect many new ones to appear. In contrast to relying solely on camera type classification, it is more suitable for image forgery detection to determine whether two investigated image patches come from cameras of the same type. Drawing inspiration from the work of Huh [40] and Mayer [13], the Siamese network, as illustrated in Figure 3, is well-suited for such similarity comparisons between two inputs. A brief description of the Siamese Network concept is provided below.

The Siamese network consists of two identical subnetworks running in parallel, sharing weights and parameters. These subnetworks take two inputs and map them into a lower-dimensional feature space. The feature vectors of the inputs are compared by calculating their distance or similarity. This concept, introduced by Bromley et al. [41], was initially used for verifying signature matches on checks against a bank's reference signatures. If the similarity score exceeds a threshold, the signatures are considered from the same person; otherwise, a forgery case is suspected. Nair et al. [42] extended this idea to face verification. Unlike typical classifier networks, Siamese networks focus on comparing differences between extracted feature vectors, leading to widespread use in natural language processing and object tracking. For example, [43] incorporated a Bidirectional Long Short-Term Memory Network (Bi-LSTM) as the core of the parallel networks, facilitating the evaluation of semantic similarity between strings of different lengths. Ref. [44] utilized a Siamese network for feature extraction, followed by distinct classification and regression

tasks in both the foreground and background, employing two parallel Region Proposal Networks (RPNs). This arrangement is commonly referred to as Siamese-RPN.

We employ a Siamese network to compare features from different patches, assessing their similarity and determining whether they belong to the same camera type. This approach enhances the generalizability of the extracted features. The design, depicted in Figure 4, involves training a robust feature extractor and freezing its parameters. The classification output layer is removed, and the second-to-last fully connected layer serves as the feature output, serving as the backbone output of the parallel networks. A pair of fully connected layers with shared weights processes the features separately on each side. The resulting features from both sides are multiplied element-wise and concatenated with the multiplied results. Finally, this concatenated feature undergoes two additional fully connected layers and a ReLU activation function to produce the output. The weights are updated using the contrastive loss [45] as the loss function, and the calculation of this loss function is shown in (1).

$$L\left(W, Y, \vec{X_1}, \vec{X_2}\right) = (1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\left\{\max(0, m - D_W)\right\}^2 \quad (1)$$

where D_W denotes the distance between an input sample and a positive sample. The binary label Y is set to 1 if an input sample and a positive sample belong to the same class, and 0 otherwise. The parameter m represents a predefined margin, regulating the minimum distance between negative and positive samples. The dual purpose of this loss function is as follows: when an input sample and a positive sample share the same class, we aim for their distance to be minimal, resulting in a smaller loss value. Conversely, when an input sample and a negative sample belong to different classes, their distance must be at least the predefined margin m, ensuring the loss is 0. This enables the clustering of similar samples in the embedding space, preventing them from being scattered apart.

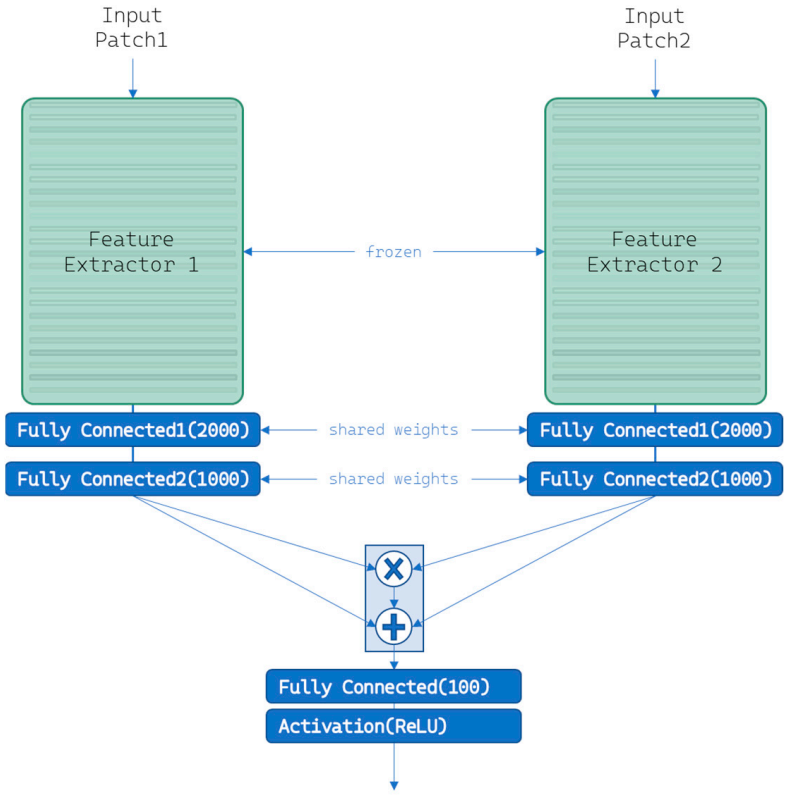


Figure 4. The similarity evaluation network.

As previously stated, incorporating a Siamese network in this context not only improves the model's ability to generalize but also extends its functionality beyond mere classification to patch-level comparisons. The dataset containing camera types utilized for this purpose is distinct from the one employed in training the feature extractor, broadening the detection capability to include unknown cameras.

3.5. Image Forgery Detection

Given the Siamese network's ability to assess similarity between two image patches, we can directly apply it to image tampering detection. A simple approach involves selecting a target patch and then scanning the entire image, comparing it with all other regions. Based on the similarity values or scores, a threshold is set; patches with similarity values above the threshold are classified as similar to the target patch, while those below the threshold are considered dissimilar. Figure 5 illustrates an ideal outcome, where the image is divided into two areas (shown in red and green), with similar patches grouped into one area, distinct from other regions. The right image illustrates the actual tampering location, corresponding to the red area in the left image. However, this evaluation is conducted on individual patches, potentially resulting in stitched regions with overlapping patches. The order of processing patches also influences the completeness of the identified forged areas.

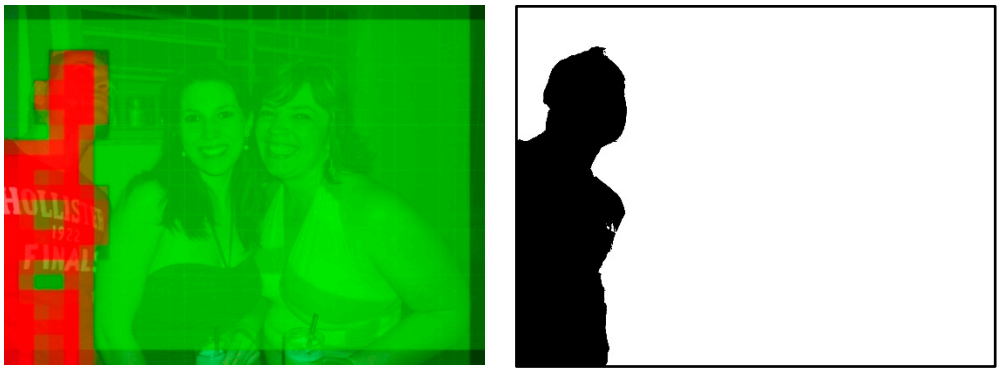


Figure 5. An image forgery detection example.

Additionally, we noticed that patches with similar variations in color or brightness tend to receive higher similarity values, as the adopted network may emphasize high-dimensional features distinct from the image content. For example, when comparing sky and ground patches extracted from the same image, the similarity score between sky patches tends to be higher than that between sky and ground patches. This similarity bias could result in false positives due to content resemblance. To address this issue, we propose the following two strategies.

Selecting Target Patches

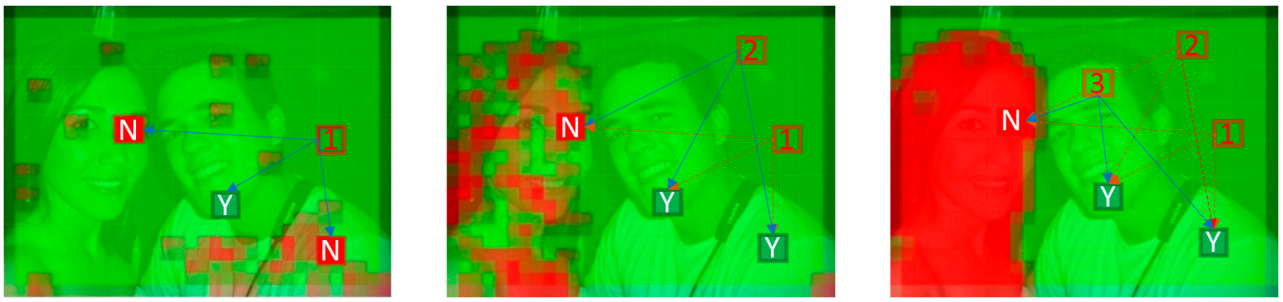
Using a single target patch for comparison is a feasible solution, but it may overlook potentially shared characteristics among other patches, leading to possible detection errors. Randomly selecting target patches for comparison can result in unstable sampling and inconsistent results. To address these limitations, we leverage the Structural Similarity Index (SSIM) [46] as the patch selection criterion to choose candidate patches for comparison. The SSIM measures the product of three similarity components: luminance similarity, contrast similarity, and structure similarity, calculated separately. SSIM values range from 0 to 1, with higher values indicating greater similarity and lower values indicating dissimilarity. The patch selection process is outlined as follows:

1. Randomly select a patch as the initial target patch.
2. Execute a scanning-based detection on the entire image using the target patch.
3. Select candidate patches that exhibit similarity to the target patch with relatively low SSIM values.

4. From the candidate patches, randomly pick one as the new target patch for the next iteration.
5. Repeat Steps 2 to 4 until the preset number of iterations is achieved.

In Step 3, selecting patches similar to the previous one ensures each detection round is performed under a more consistent setting. Low SSIM values indicate that candidate patches have content or structures different from the previous target patch, allowing us to select patches with larger content disparities. This approach facilitates multiple detection rounds by avoiding situations where newly selected target patches closely resemble previously chosen ones, potentially leading to errors determined by the Siamese network.

In addition to using the SSIM to select target patches, the accumulation of similarity measurements from each comparison, combined with the new score, also requires adjustments based on the SSIM. The adjustment refines the weighting of accumulated predictions and the current prediction. Specifically, if the SSIM is higher than a threshold (set as 0.5), the accumulated prediction results from previous rounds contribute M% (M = 40), and the current prediction result contributes $(1 - M)$%, which is larger than M%. Conversely, if the SSIM is lower than or equal to the threshold, the accumulated predictions contribute more than the current prediction. This strategy helps to mitigate the impact of content variations, and it has been observed that these adjustments achieve more consistent outcomes. Figure 6 illustrates an example. The process involves three rounds of comparative detection. The initial target patch in Figure 6a is chosen randomly, while the target patches in (b,c) are chosen based on the previous round's results and the SSIM indicator. We can see that this method effectively corrects biases caused by single-round target patch detection.



(**a**) First round (**b**) Second round (**c**) Third round

Figure 6. Comparison of blocks. "Y" indicates "Similar", and "N" indicates "Different". The numbers show the order of selected target patches.

While the results can roughly identify tampered areas, there is no clear segmentation between tampered and unaltered regions. To address this, patch-level thresholding based on Otsu's algorithm is employed, drawing segmentation lines to ensure no ambiguous zones for inspection and assessment. Further refinement through foreground extraction is then applied to enhance boundary details and avoid a blocky effect. Foreground extraction is achieved through the use of FBA Matting (Foreground–Background-Aware Matting) [47], a technique that accurately extracts the contours of foreground objects from images. FBA Matting employs ResNet-50 and U-Net models to classify pixels, utilizing previously acquired masks to categorize pixels into three classes: background, foreground, and unknown. This is performed by constructing a "Tri-Map", a ternary graph, based on the original mask. The Tri-Map generation involves expanding the original mask to encompass potentially unknown areas, or neutral regions, through dilation. Simultaneously, erosion is applied to draw the center region of the mask with high confidence. These steps are illustrated in Figure 7. Once the completed ternary diagram and the original image are provided to FBA Matting, it applies predictions in the neutral regions of the ternary diagram based on the correlation between foreground and background areas, resulting in refined and accurate segmentation.

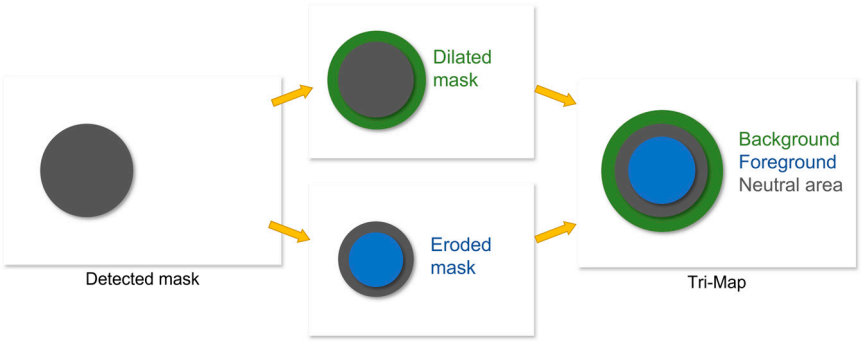


Figure 7. Tri-map generation.

3.6. Deepfake Video Detection

Most existing Deepfake detection techniques treat videos as collections of independent frames, relying solely on the average confidence scores of frames to determine the authenticity of the entire video. This approach adopts a frame-level perspective and overlooks inter-frame correlations. Given that current Deepfake modifications are applied on a per-frame basis, leading to potential inconsistencies, our proposed scheme aims to capture content discrepancies or misaligned defects within frames during the Deepfake manipulation process.

While comparing all video frames would offer the most comprehensive analysis, it can be time consuming. Conversely, selecting a single frame may not provide reliable detection. To strike a balance between the two, we choose to uniformly select a portion (one-tenth) of video frames as detection target frames. The frames before and after these detection frames are then extracted as reference frames. The process is outlined as follows:

1. Select the detection target frames and reference frames for similarity comparison.
2. Locate facial regions and extract associated patches.
3. Employ the similarity network to compare the region similarity of blocks.
4. Average all the detection results to determine the video's authenticity.

It is crucial to select suitable regions within the detected facial areas. We explored three adjustment methods, illustrated in Figure 8, including extracting all facial patches via a sliding window, capturing only the region of interest, and extracting central and peripheral patches. While resizing face blocks to a specified size is a straightforward approach, it may affect the high-dimensional features for content similarity examination. Extracting all facial blocks can preserve more complete features, but the number of extracted facial patches may vary with varying facial sizes, potentially causing significant detection discrepancies. The setting of the sliding window's stride and the issue of block overlapping also need consideration. After thorough testing, we choose to extract patches from the central and peripheral regions of facial areas, as shown in Figure 8d.

Considering the precise positions of the patches, as illustrated in Figure 9, we observed that focusing solely on faces may not be sufficient. Certain portions of the peripheral areas around the faces can retain useful features. To address this, we designated five anchor points for each face: the upper, lower, left, and right, as well as the center. We move the center point inward by two-thirds of the side of the face block to select suitable areas for checking similarity.

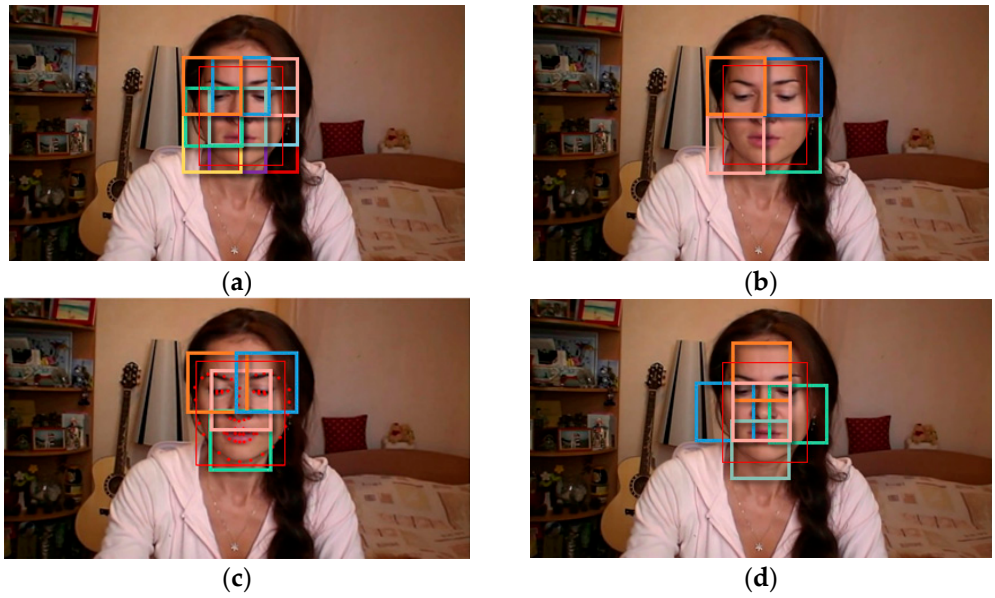


Figure 8. Patch selection: (**a**) overlapped patches, (**b**) non-overlapped patches, (**c**) patches around the region of interest, and (**d**) central and peripheral parts of a face.

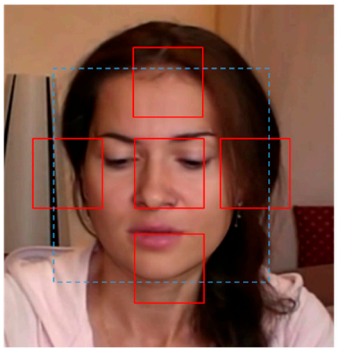


Figure 9. Five positions of selected patches for a face.

4. Experimental Results

4.1. Experiment Settings

This study was conducted on Ubuntu 18.04 LTS using Python as the primary programming language. The environment was built with PyTorch 1.12.1, Torchvision 0.13.1, and OpenCV 4.6.0. The hardware setup includes an Intel® Core(TM) i7-8700K CPU with 3.70 GHz, 64 GB RAM, and a GeForce GTX 2080 Ti GPU. To enhance GPU computational performance, the CUDA-11.2 framework is employed, along with cuDNN 8.0 for accelerating deep learning tasks.

For the feature extractor network, the optimizer used is AdamW, with an initial learning rate of 0.001 and a batch size of 128. The learning rate is reduced by a factor of 1/5 every 20 iterations, and a total of 200 iterations are performed during training. In the Siamese network, the same optimizer (AdamW) is employed, with a batch size of 64 and an initial learning rate of 5×10^{-4}. The new learning rate is calculated as the old learning rate divided by the factor $1.2^{\frac{\text{epoch}}{10}}$ in each iteration. The training process executes for a total of 200 iterations.

4.2. Training Data

When training the feature extractor as described in Section 3.2, we compiled various camera types from datasets including the VISION dataset [48], the Camera Model Identification Challenge (CMIC) dataset [49], and our collected dataset. After removing duplicates

and conducting thorough tests and comparisons, we identified the 40 most suitable classes (24 from VISION, 8 from CMIC, and 8 from our own collection) for our training dataset. For the similarity network, outlined in Section 3.3, we utilized 25 camera-type datasets as our training data, which are from the Dresden dataset [50]. The difference in the selected datasets compared to those used for the feature extractor aims to enable the model to learn from the 40 camera-type classes first. Subsequently, the model can be fine-tuned using the unknown 25 classes during the similarity network learning. The model's capabilities of dealing with unknown camera types can be further enhanced. It should be noted that all the images in these datasets are used for training since the investigated images in the experiments will not be restricted to the dataset containing camera-type information. Besides, it is not easy for us to collect more images related to specific camera types.

In terms of patch sizes, we experimented with three sizes: 256 × 256, 128 × 128, and 64 × 64. Patches of size 256 × 256 were deemed too large for making precise predictions on smaller objects, while blocks of size 64 × 64 were considered a bit small, resulting in significantly increased error rates. Therefore, we settled on a block size of 128 × 128. During the training of the feature extractor, images of each category were partitioned into overlapping or non-overlapping blocks of size 128 × 128 pixels. We randomly selected 20,000 blocks from each category for both overlapping and non-overlapping cases, resulting in a total of 1,560,000 patches for training the feature extractor's classification network. An additional set of 579,973 patches was reserved for validation. For the similarity network, patches of size 128 × 128 were extracted in a non-overlapping manner. We collected 40,000 patches for each category, resulting in a total of 1,000,000 patches for training the similarity network. Another set of 600,000 patches was reserved for validation. Table 1 shows the accuracy rates of the feature extractor and the similarity network, both achieving higher than 0.9, which shows that the model training is quite successful.

Table 1. The accuracy of the network training.

Model	Accuracy
Feature extractor	0.95
Siamese network	0.90

We assessed the accuracy by comparing the camera-type model predictions from the feature extractor against the ground truth, forming a multi-class confusion matrix. Most of the accuracy rates in this matrix exceeded 94%, with the lowest accuracy within a single class at 88.6% and the highest at 99.9%. The original training dataset comprised 85 classes, and due to images from various sources, instances of similar or identical cameras led to misclassifications in the confusion matrix. To mitigate this, we employed k-fold cross-validation to ensure no errors were introduced while removing converged but effective classes. As a result, 65 classes remained, with 40 selected for the feature extractor training and 25 reserved for the training of the Siamese or similarity network. This approach aims to maintain robust performance while addressing potential misclassifications arising from similar cameras.

4.3. Forgery Image Detection Results

4.3.1. Visual Evaluation

The proposed UFCC scheme was initially applied to image forgery detection by computing the patch consistency. Figure 10 presents some detection results using the DSO-1 dataset [51] as the test data. The left column features original images, the middle column displays the ground truth with marked tampered areas, and the right column showcases the accurate detection results. The proposed method effectively identifies the locations of tampered regions.

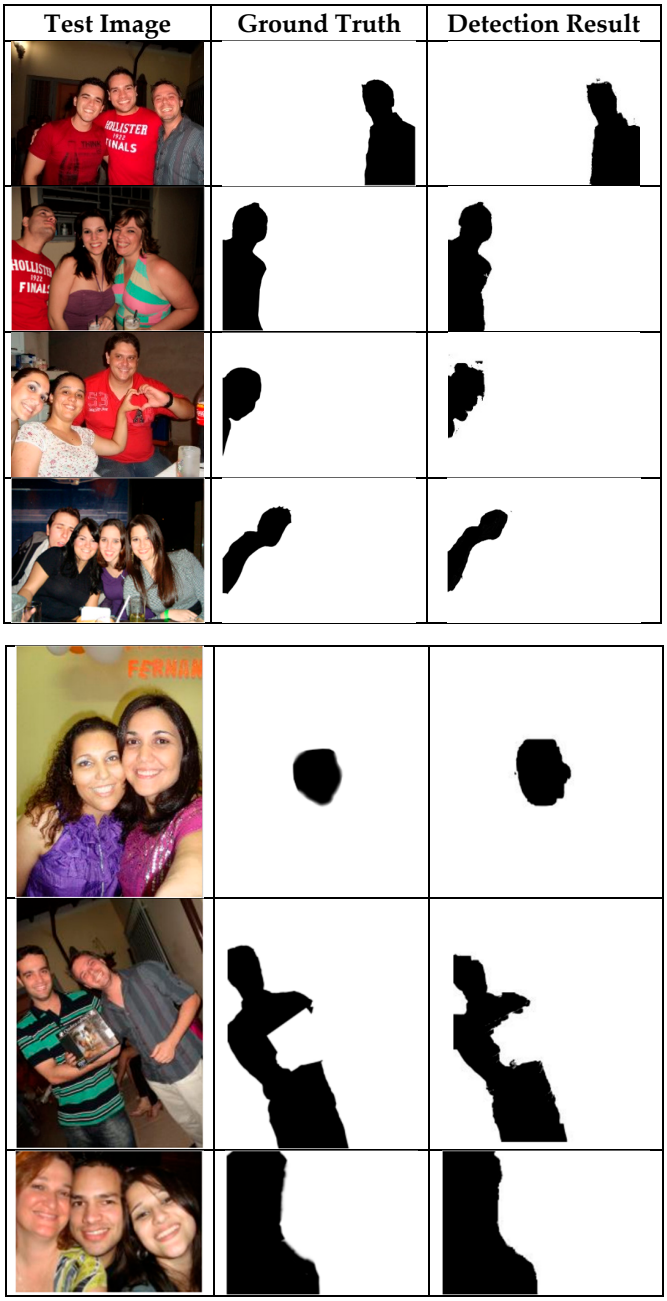


Figure 10. Examples of image forgery detection.

4.3.2. Evaluation Metrics

Next, we assess the model's performance using the following metrics and present the results in a table for comparison with existing studies. The four adopted metrics are mAP [52], F1 measure [53], MCC [54], and cIoU [55]. We categorize all pixels into the four classes according to Table 2.

Table 2. Pixel Detection Classification.

Prediction\Truth	Forgery	Not Forgery
Forgery	True Positive (TP)	False Positive (FP)
Not Forgery	False Negative (FN)	True Negative (TN)

- mAP (mean Average Precision)

The metric mAP (mean Average Precision) is calculated based on the Precision–Recall (PR) curve, which illustrates the relationship between the model's precision and recall at different thresholds. The precision and recall are computed as follows:

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

The PR curve assists in determining the Average Precision (AP) value for each class. AP represents the average precision across various recall levels. The AP values for all classes are averaged to obtain the mAP, which we follow [52]. It is important to note that these calculations are based on each pixel and are subsequently averaged to derive a single detection value for an image.

- F1-measure

The F-score [53], as shown in (4), is a metric that considers both the precision and recall rates. It incorporates a custom parameter denoted as β. When β is set to 1, the resulting F-score is the F1-score, where the precision and recall are given equal weights.

$$\mathrm{F-score} = \frac{\left(\left|1 + \beta^2\right|\right)\left|\mathrm{precision}\right| \times \left|\mathrm{recall}\right|}{\beta^2\left|\mathrm{precision}\right| + \left|\mathrm{recall}\right|} \tag{4}$$

- MCC (Matthews Correlation Coefficient)

The first two evaluation metrics are effective in many cases, but they might not perform well when dealing with situations where only the TP and FP have values, leading to the accuracy paradox. In such cases, the requirements for predicting negative samples might not be adequately met. The MCC considers the quantities of the TP, TN, FP, and FN, providing an effective way to evaluate the model performance on imbalanced datasets. The MCC is a comprehensive metric that overcomes the limitations of simple measures, as shown in (5) [54].

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{5}$$

The MCC has a range of values between -1 and 1, where 1 indicates a perfect prediction, 0 signifies a random prediction, and -1 suggests an opposite prediction. The MCC can handle the imbalance between correct and incorrect predictions on imbalanced datasets. We found that the MCC is the most effective indicator, capable of dealing with extreme scenarios.

- cIoU

cIoU is calculated using a weighted Jaccard score. IoU is computed separately for tampered and non-tampered regions, and then the average value is calculated based on the proportions within their respective frames [55]. All the pixels are involved in this calculation.

4.3.3. Performance Comparison

Table 3 compares the detection results using the four metrics for the various methods listed in [56], focusing on the DSO-I dataset [51]. Among the methods, [40] and the proposed method share similarities in using the consistency of image blocks to identify tampered areas. However, both methods require collecting a substantial amount of training data for each EXIF tag, which might be challenging due to the possible removal of EXIF data in real-world images, making it less feasible to acquire complete data. It is worth noting that the first three methods are traditional approaches, while the latter four are deep learning methods.

Table 3. Performance evaluation of forged image detection.

Method\Metrics	mAP	cIoU	MCC	F1
CFA [57]	0.24	0.46	0.16	0.29
DCT [58]	0.32	0.51	0.19	0.31
NOI [59]	0.38	0.50	0.25	0.34
E-MFCN [60]	-	-	0.41	0.48
Mantranet [61]	-	0.43	0.02	0.48
DCUNet [62]	-	0.53	0.27	0.62
SENet [63]	-	0.54	0.26	0.61
EXIF-Consistency [40]	0.52	0.63	0.42	0.52
Ours	**0.58**	**0.83**	**0.56**	**0.63**

4.3.4. Ablation Tests

In Table 4, we inserted breakpoints at each step in the process of locating tampered areas. It is evident that the inclusion of the SSIM improves performance. Incorporating FBA Matting [47] substantially enhances the mask contours and boosts the scores. The refinement step aims to assign more suitable tampered and unaltered regions by examining their areas, assuming the tampered regions occupy smaller areas. The experimental results confirm that each step contributes to the improvement in generating tampered masks.

Table 4. Ablation tests of forged image detection.

Method\Metrics	mAP	cIoU	MCC	F1
Base	0.43	0.72	0.36	0.46
Base + SSIM	0.45	0.75	0.41	0.48
Base+SSIM + FBAmatting	0.54	0.76	0.48	0.56
Base+SSIM + FBAmatting + Refine	**0.58**	**0.83**	**0.56**	**0.63**

4.4. Detecting Deepfake Videos

4.4.1. Performance Comparison

In Table 5, we employed the FaceForensics++ [32] (FF++) dataset for testing. This dataset comprises 1000 original videos and 4000 manipulated fake videos generated by four different methods: Deepfakes [14] (DF), Face2Face [19] (F2F), FaceSwap [64] (FS), and NeuralTextures [65] (NT). We calculated the accuracy for each of these forgery techniques, and the results show that the proposed approach outperforms most of these studies in terms of the effectiveness. Although the proposed scheme performs slightly worse than [32] in FS (face swapping) probably because the faces in the FS videos are simply replaced, we think that this weakness may not be that serious as FS videos look much less real, compared to other Deepfake operations.

Table 5. Performance evaluation of deepfake video detection.

Method\Tests	DF [14]	F2F [19]	FS [64]	NT [65]
Steg. Features [66]	0.736	0.737	0.689	0.633
Cozzolino et al. [67]	0.855	0.679	0.738	0.780
Bayar and Stamm [39]	0.846	0.737	0.825	0.707
MesoNet [30]	0.896	0.886	0.812	0.766
Rossler et al. [32]	0.976	0.977	**0.968**	0.922
The proposed method	**0.984**	**0.984**	0.932	**0.972**

We compare the results with different models using FF++ [32] and the Celeb-DF [68] (CDF) dataset. It is worth mentioning that Celeb-DF comprises celebrity videos downloaded from YouTube and processed by more advanced Deepfake operations. Table 6 shows that the proposed scheme still yields the best results. Furthermore, some existing work even

used the test dataset's videos for training. The proposed UFCC scheme made use of original/unaltered imagery patches for training, without targeting a specific dataset.

Table 6. FF++ [32] and Celeb-DF [68] Deepfake video detection results.

Method\Dataset	FF++ [32]	CDF [68]
Two-stream [29]	0.701	0.538
Meso4 [30]	0.847	0.548
MesoInception4 [30]	0.830	0.536
FWA [69]	0.801	0.569
DSP-FWA [69]	0.930	0.646
VA-MLP [70]	0.664	0.550
Headpose [38]	0.473	0.546
Capsule [31]	0.966	0.575
SMIL [71]	0.968	0.563
Two-branch [72]	0.932	0.734
SPSL [34]	0.969	0.724
The proposed scheme	**0.97**	**0.754**

4.4.2. Reference Frame Selection

In Section 3.5, we discussed the selection of reference frames in Deepfake video detection and found that the choice of reference frames can affect the performance and stability. We can see from Table 7 that, although selecting a single reference frame before or after the target one already yields satisfactory results, extensive tests revealed that the single-sided detection might lead to errors in certain videos. Including frames from both sides is quite helpful. In addition, the number of reference frames is also an issue to be examined. Table 8 shows that increasing the number of frames indeed enhances the accuracy, but it certainly incurs a heavier computational load in the detection process. The performance also starts to decline when five reference frames are considered in each direction. Considering both the performance improvement and avoiding an excessive detection time, we choose to use four frames preceding and after the target frame as the reference frames.

Table 7. Reference frame selection.

Frames\Tests	DF [14]	F2F [19]	FS [64]	NT [65]
1 preceding frame	0.940	0.955	0.895	0.941
1 following frame	0.952	0.961	0.902	0.940
1 preceding and 1 following frames	**0.954**	**0.963**	**0.905**	**0.944**

Table 8. Number of reference frames.

# of Frames\Tests	DF [14]	F2F [19]	FS [64]	NT [65]
1	0.954	0.963	0.905	0.944
2	0.966	0.970	0.916	0.955
3	0.979	0.982	0.921	0.969
4	**0.984**	**0.984**	**0.932**	0.972
5	0.982	**0.984**	0.928	**0.973**

5. Conclusions and Future Work

5.1. Conclusions

This research introduces a robust image and video tampering detection scheme, UFCC, aimed at identifying manipulated regions in images and verifying video authenticity based on content consistency. The scheme utilizes camera models to train the feature extractor through deep learning, and a Siamese network to assess the similarity between the examined patches. Addressing image tampering involves implementing patch selection strategies for approximating manipulated areas, followed by refinement through FBA

Matting. In Deepfake video detection, facial regions are initially extracted, and similar detection methods are applied to patches in frames to determine whether the investigated video is a Deepfake forgery.

The UFCC scheme, when compared to existing methods, not only demonstrates superior performance but also distinguishes itself by presenting a comprehensive detection methodology capable of handling both image and video manipulation scenarios. In detecting image inpainting, UFCC surpasses existing work on the DSO-I dataset [51] with a higher mAP [52], cIoU [53], MCC [54], and F1 [55] values of 0.58, 0.83, 0.56, and 0.63, respectively. For the detection of Deepfake videos, UFCC excels on the DF [14], F2F [19], and NT [65] tests with accuracy rates of 0.982, 0.984, and 0.973, respectively. Although the accuracy of the FS [64] testing is slightly lower at 0.928, it is worth noting that FS contains less realistic images and is considered a milder threat. Notably, our training data excludes manipulated images or Deepfake datasets, enhancing the proposed scheme's generalization capability. This absence of specific target tampering operations makes the UFCC scheme more flexible and adaptive, enabling it to handle a wide range of content manipulations, including mixed scenarios.

5.2. Future Work

While the proposed scheme demonstrates effectiveness across most tested imagery data, challenges arise in handling severely overexposed regions that lack informative content. Furthermore, our training dataset for the feature extractor consists of randomly sampled patches to form around 20,000 instances from each class of camera types. The use of only raw images without lossy compression during training poses a limitation, impacting detection performance when processing compressed videos or low-quality images. To address this, in addition to collecting and filtering more diverse camera-type images to expand the dataset, it might be worthwhile to consider controlled image degradation or lossy compression in the model training so that a reasonable discrimination capability can still be maintained.

Author Contributions: Conceptualization, P.-C.S. and T.-Y.K.; methodology, P.-C.S. and T.-Y.K.; software, B.-H.H.; validation, B.-H.H. and P.-C.S.; formal analysis, P.-C.S. and T.-Y.K.; investigation, P.-C.S. and T.-Y.K.; resources, P.-C.S.; data curation, B.-H.H.; writing—original draft preparation, P.-C.S. and B.-H.H.; writing—review and editing, P.-C.S. and T.-Y.K.; visualization, B.-H.H. and P.-C.S.; supervision, P.-C.S. and T.-Y.K.; project administration, P.-C.S. and T.-Y.K.; funding acquisition, P.-C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council, Taiwan, grant number NSTC 111-2221-E-008-098 and grant number 112-2221-E-008-077.

Data Availability Statement:

Model training datasets:

1. VISION [48]: https://lesc.dinfo.unifi.it/VISION/
2. CMIC [49]: https://www.kaggle.com/competitions/sp-society-camera-model-identification
3. Dresden [50]: http://forensics.inf.tu-dresden.de/ddimgdb/

Image manipulation test dataset:

DSO-1 [51]: https://recodbr.wordpress.com/code-n-data/#dso1_dsi1

Deepfake video test dataset:

1. Deepfakes [14]: https://github.com/deepfakes/faceswap
2. Face2Face [19]: https://www.kaggle.com/datasets/mdhadiuzzaman/face2face
3. Faceswap [64]: https://github.com/MarekKowalski/FaceSwap
4. NeuralTextures [65]: https://github.com/SSRSGJYD/NeuralTexture
5. FF++ [32]: https://github.com/ondyari/FaceForensics
6. Celeb-DF [68]: https://github.com/yuezunli/celeb-deepfakeforensics

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kuo, T.Y.; Lo, Y.C.; Huang, S.N. Image forgery detection for region duplication tampering. In Proceedings of the 2013 IEEE International Conference on Multimedia and Expo, San Jose, CA, USA, 15–19 July 2013; pp. 1–6. [CrossRef]
2. Muhammad, G.; Hussain, M.; Bebis, G. Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digit. Investig.* **2012**, *9*, 49–57. [CrossRef]
3. Kirchner, M.; Gloe, T. Forensic camera model identification. In *Handbook of Digital Forensics of Multimedia Data and Devices*; Wiley: Hoboken, NJ, USA, 2015; pp. 329–374.
4. Swaminathan, A.; Wu, M.; Liu, K.R. Nonintrusive component forensics of visual sensors using output images. *IEEE Trans. Inf. Forensics Secur.* **2007**, *2*, 91–106. [CrossRef]
5. Filler, T.; Fridrich, J.; Goljan, M. Using sensor pattern noise for camera model identification. In Proceedings of the 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; IEEE: New York, NY, USA, 2008; pp. 1296–1299.
6. Xu, G.; Shi, Y.Q. Camera model identification using local binary patterns. In Proceedings of the IEEE International Conference on Multimedia and Expo, Melbourne, Australia, 9–13 July 2012; IEEE: New York, NY, USA, 2012; pp. 392–397.
7. Thai, T.H.; Cogranne, R.; Retraint, F. Camera model identification based on the heteroscedastic noise model. *IEEE Trans. Image Process.* **2013**, *23*, 250–263. [CrossRef] [PubMed]
8. Van, L.T.; Emmanuel, S.; Kankanhalli, M.S. Identifying source cell phone using chromatic aberration. In Proceedings of the IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007; IEEE: New York, NY, USA, 2007; pp. 883–886.
9. Farid, H. Image forgery detection. *IEEE Signal Process. Mag.* **2009**, *26*, 16–25. [CrossRef]
10. Wang, H.-T.; Su, P.-C. Deep-learning-based block similarity evaluation for image forensics. In Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taoyuan, Taiwan, 28–30 September 2020; IEEE: New York, NY, USA, 2020.
11. Dirik, A.E.; Memon, N. Image tamper detection based on demosaicing artifacts. In Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; IEEE: New York, NY, USA, 2009; pp. 1497–1500.
12. Bondi, L.; Lameri, S.; Guera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. Tampering detection and localization through clustering of camera-based CNN features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1855–1864.
13. Mayer, O.; Stamm, M.C. Learned forensic source similarity for unknown camera models. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; IEEE: New York, NY, USA, 2018; pp. 2012–2016.
14. Deepfakes. Available online: https://github.com/deepfakes/faceswap (accessed on 21 November 2021).
15. Korshunova, I.; Shi, W.; Dambre, J.; Theis, L. Fast face-swap using convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3677–3685.
16. Zakharov, E.; Shysheya, A.; Burkov, E.; Lempitsky, V. Few-shot adversarial learning of realistic neural talking head models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9459–9468.
17. Zhu, Y.; Li, Q.; Wang, J.; Xu, C.-Z.; Sun, Z. One shot face swapping on megapixels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4834–4844.
18. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of styleGAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8110–8119.
19. Thies, J.; Zollhofer, M.; Stamminger, M.; Theobalt, C.; Nießner, M. Face2face: Real-time face capture and reenactment of RGB videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2387–2395.
20. Kim, H.; Garrido, P.; Xu, W.; Thies, J.; Nießner, M.; Pérez, P.; Richardt, C.; Zollhöfer, M.; Theobalt, C. Deep video portraits. *ACM Trans. Graph.* **2018**, *37*, 1–14. [CrossRef]
21. Fried, O.; Tewari, A.; Zollhöfer, M.; Finkelstein, A.; Shechtman, E.; Goldman, D.B.; Genova, K.; Jin, Z.; Theobalt, C.; Agrawala, M. Text-based editing of talking-head video. *ACM Trans. Graph.* **2019**, *38*, 1–14. [CrossRef]
22. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4401–4410.
23. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
24. Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; Choo, J. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8789–8797.
25. Nguyen, T.; Tran, A.T.; Hoai, M. Lipstick ain't enough: Beyond color matching for in-the-wild makeup transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 13305–13314.
26. Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional GANs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8798–8807.

27. Nirkin, Y.; Keller, Y.; Hassner, T. FSGAN: Subject agnostic face swapping and reenactment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7184–7193.
28. Qu, Z.; Meng, Y.; Muhammad, G.; Tiwari, P. QMFND: A quantum multimodal fusion-based fake news detection model for social media. *Inf. Fusion* **2024**, *104*, 102172. [CrossRef]
29. Zhou, P.; Han, X.; Morariu, V.I.; Davis, L.S. Two-stream neural networks for tampered face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 1831–1839.
30. Afchar, D.; Nozick, V.; Yamagishi, J.; Echizen, I. MesoNet: A compact facial video forgery detection network. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 10–13 December 2018; IEEE: New York, NY, USA; pp. 1–7.
31. Nguyen, H.H.; Yamagishi, J.; Echizen, I. Capsule-forensics: Using capsule networks to detect forged images and videos. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: New York, NY, USA, 2019; pp. 2307–2311.
32. Rossler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; Nießner, M. Faceforensics++: Learning to detect manipulated facial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1–11.
33. Li, J.; Xie, H.; Li, J.; Wang, Z.; Zhang, Y. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 6458–6467.
34. Liu, H.; Li, X.; Zhou, W.; Chen, Y.; He, Y.; Xue, H.; Zhang, W.; Yu, N. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 772–781.
35. Nguyen, H.H.; Fang, F.; Yamagishi, J.; Echizen, I. Multi-task learning for detecting and segmenting manipulated facial images and videos. In Proceedings of the IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 23–26 September 2019; IEEE: New York, NY, USA, 2019; pp. 1–8.
36. Güera, D.; Delp, E.J. Deepfake video detection using recurrent neural networks. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; IEEE: New York, NY, USA, 2018; pp. 1–6.
37. Li, Y.; Chang, M.-C.; Lyu, S. In ictu oculi: Exposing AI created fake videos by detecting eye blinking. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 11–13 December 2018; IEEE: New York, NY, USA, 2018; pp. 1–7.
38. Yang, X.; Li, Y.; Lyu, S. Exposing deep fakes using inconsistent head poses. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: New York, NY, USA, 2019; pp. 8261–8265.
39. Bayar, B.; Stamm, M.C. A deep learning approach to universal image manipulation detection using a new convolutional layer. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, Vigo, Spain, 20–22 June 2016; pp. 5–10.
40. Huh, M.; Liu, A.; Owens, A.; Efros, A.A. Fighting fake news: Image splice detection via learned self-consistency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
41. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a Siamese time delay neural network. *Adv. Neural Inf. Process. Syst.* **1993**, *6*, 737–744. [CrossRef]
42. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
43. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with Siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
44. Neculoiu, P.; Versteegh, M.; Rotaru, M. Learning text similarity with Siamese recurrent networks. In Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, 11 August 2016; pp. 148–157.
45. Hadsell, R. Dimensionality reduction by learning an invariant mapping. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: New York, NY, USA, 2006; Volume 2, pp. 1735–1742.
46. Hadsell, R.; Chopra, S.; LeCun, Y. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.
47. Forte, M.; Pitié, F. $ F $, $ B $, Alpha Matting. *arXiv* **2020**, arXiv:2003.07711.
48. Shullani, D.; Fontani, M.; Iuliani, M.; Shaya, O.A.; Piva, A. Vision: A video and image dataset for source identification. *EURASIP J. Inf. Secur.* **2017**, *2017*, 1–16. [CrossRef]
49. Stamm, M.; Bestagini, P.; Marcenaro, L.; Campisi, P. Forensic camera model identification: Highlights from the IEEE Signal Processing Cup 2018 Student Competition. *IEEE Signal Process. Mag.* **2018**, *35*, 168–174. [CrossRef]
50. Gloe, T.; Böhme, R. The Dresden image database for benchmarking digital image forensics. In Proceedings of the ACM Symposium on Applied Computing, Sierre, Switzerland, 21–26 March 2010; pp. 1584–1590.

51. De Carvalho, T.J.; Riess, C.; Angelopoulou, E.; Pedrini, H.; de Rezende Rocha, A. Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1182–1194. [CrossRef]
52. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
53. Van Rijsbergen, C.J.; Retrieval, I. *Information Retrieval*; Butterworth-Heinemann: Oxford, UK, 1979.
54. Matthews, B.W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Et Biophys. Acta (BBA)-Protein Struct.* **1975**, *405*, 442–451. [CrossRef]
55. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceeding of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
56. Gu, A.-R.; Nam, J.-H.; Lee, S.-C. FBI-Net: Frequency-based image forgery localization via multitask learning With self-attention. *IEEE Access* **2022**, *10*, 62751–62762. [CrossRef]
57. Ferrara, P.; Bianchi, T.; De Rosa, A.; Piva, A. Image forgery localization via fine-grained analysis of CFA artifacts. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1566–1577. [CrossRef]
58. Ye, S.; Sun, Q.; Chang, E.-C. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In Proceedings of the IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007; IEEE: New York, NY, USA, 2007; pp. 12–15.
59. Mahdian, B.; Saic, S. Using noise inconsistencies for blind image forensics. *Image Vis. Comput.* **2009**, *27*, 1497–1503. [CrossRef]
60. Salloum, R.; Ren, Y.; Kuo, C.-C.J. Image splicing localization using a multi-task fully convolutional network (MFCN). *J. Vis. Commun. Image Represent.* **2018**, *51*, 201–209. [CrossRef]
61. Wu, Y.; AbdAlmageed, W.; Natarajan, P. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9543–9552.
62. Ding, H.; Chen, L.; Tao, Q.; Fu, Z.; Dong, L.; Cui, X. DCU-Net: A dual-channel U-shaped network for image splicing forgery detection. *Neural Comput. Appl.* **2023**, *35*, 5015–5031. [CrossRef]
63. Zhang, Y.; Zhu, G.; Wu, L.; Kwong, S.; Zhang, H.; Zhou, Y. Multi-task SE-network for image splicing localization. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4828–4840. [CrossRef]
64. Faceswap. Available online: https://github.com/MarekKowalski/FaceSwap (accessed on 13 November 2021).
65. Thies, J.; Zollhöfer, M.; Nießner, M. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
66. Fridrich, J.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882. [CrossRef]
67. Cozzolino, D.; Poggi, G.; Verdoliva, L. Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, Philadelphia, PA, USA, 20–21 June 2017; pp. 159–164.
68. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 3207–3216.
69. Li, Y.; Lyu, S. Exposing deepfake videos by detecting face warping artifacts. *arXiv* **2018**, arXiv:1811.00656.
70. Matern, F.; Riess, C.; Stamminger, M. Exploiting visual artifacts to expose deepfakes and face manipulations. In Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW), Waikoloa Village, HI, USA, 7–11 January 2019; IEEE: New York, NY, USA, 2019; pp. 83–92.
71. Li, X.; Lang, Y.; Chen, Y.; Mao, X.; He, Y.; Wang, S.; Xue, H.; Lu, Q. Sharp multiple instance learning for deepfake video detection. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1864–1872.
72. Masi, I.; Killekar, A.; Mascarenhas, R.M.; Gurudatt, S.P.; AbdAlmageed, W. Two-branch recurrent network for isolating deepfakes in videos. In Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Proceedings, Part VII 16. Springer: Berlin/Heidelberg, Germany, 2020; pp. 667–684.

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.