Deep Learning for Multimedia Forensics

Other titles in Foundations and Trends® in Computer Graphics and Vision

Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art
Joel Janai, Fatma Güney, Aseem Behl and Andreas Geiger
ISBN: 978-1-68083-688-2

Discrete Graphical Models - An Optimization Perspective
Bogdan Savchynskyy
ISBN: 978-1-68083-638-7

Line Drawings from 3D Models: A Tutorial
Pierre Bénard and Aaron Hertzmann
ISBN: 978-1-68083-590-8

Publishing and Consuming 3D Content on the Web: A Survey
Marco Potenziani, Marco Callieri, Matteo Dellepiane and Roberto Scopigno
ISBN: 978-1-68083-536-6

Crowdsourcing in Computer Vision
Adriana Kovashka, Olga Russakovsky, Li Fei-Fei and Kristen Grauman
ISBN: 978-1-68083-212-9

The Path to Path-Traced Movies
Per H. Christensen and Wojciech Jarosz
ISBN: 978-1-68083-210-5

(Hyper)-Graphs Inference through Convex Relaxations and Move Making Algorithms
Nikos Komodakis, M. Pawan Kumar and Nikos Paragios
ISBN: 978-1-68083-138-2

Deep Learning for Multimedia Forensics

Irene Amerini
Sapienza University of Rome
amerini@diag.uniroma1.it

Aris Anagnostopoulos
Sapienza University of Rome
aris@diag.uniroma1.it

Luca Maiano
Sapienza University of Rome
ELIS Innovation Hub
maiano@diag.uniroma1.it

Lorenzo Ricciardi Celsi
ELIS Innovation Hub
l.ricciardicelsi@elis.org

the essence of knowledge
Boston — Delft

Foundations and Trends® in Computer Graphics and Vision

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

I. Amerini, A. Anagnostopoulos, L. Maiano and L.R. Celsi. *Deep Learning for Multimedia Forensics*. Foundations and Trends® in Computer Graphics and Vision, vol. 12, no. 4, pp. 309–457, 2021.

ISBN: 978-1-68083-855-8

Foundations and Trends® in Computer Graphics and Vision
Volume 12, Issue 4, 2021

Editorial Scope

Topics

Foundations and Trends® in Computer Graphics and Vision publishes survey and tutorial articles in the following topics:

- Rendering
- Shape
- Mesh simplification
- Animation
- Sensors and sensing
- Image restoration and enhancement
- Segmentation and grouping
- Feature detection and selection
- Color processing
- Texture analysis and synthesis
- Illumination and reflectance modeling
- Shape representation
- Tracking
- Calibration
- Structure from motion
- Motion estimation and registration
- Stereo matching and reconstruction
- 3D reconstruction and image-based modeling
- Learning and statistical methods
- Appearance-based matching
- Object and scene recognition
- Face detection and recognition
- Activity and gesture recognition
- Image and video retrieval
- Video analysis and event recognition
- Medical image analysis
- Robot localization and navigation

Information for Librarians

Foundations and Trends® in Computer Graphics and Vision, 2021, Volume 12, 4 issues. ISSN paper version 1572-2740. ISSN online version 1572-2759. Also available as a combined paper and online subscription.

Contents

Deep Learning for Multimedia Forensics

Irene Amerini[1], Aris Anagnostopoulos[1], Luca Maiano[1,2] and Lorenzo Ricciardi Celsi[2]

[1]*Sapienza University of Rome, Italy;*
[2]*ELIS Innovation Hub, Italy*

ABSTRACT

In the last two decades, we have witnessed an immense increase in the use of multimedia content on the internet, for multiple applications ranging from the most innocuous to very critical ones. Naturally, this emergence has given rise to many types of threats posed when this content can be manipulated/used for malicious purposes. For example, fake media can be used to drive personal opinions, ruining the image of a public figure, or for criminal activities such as terrorist propaganda and cyberbullying. The research community has of course moved to counter attack these threats by designing manipulation-detection systems based on a variety of techniques, such as signal processing, statistics, and machine learning. This research and practice activity has given rise to the field of *multimedia forensics*.

The success of deep learning in the last decade has led to its use in multimedia forensics as well. In this survey, we look at the latest trends and deep-learning-based techniques introduced to solve three main questions investigated in the field of multimedia forensics. We begin by examining the manipulations of images and videos produced with editing tools, reporting the deep-learning approaches adopted to

Irene Amerini, Aris Anagnostopoulos, Luca Maiano and Lorenzo Ricciardi Celsi (2021), "Deep Learning for Multimedia Forensics", Foundations and Trends® in Computer Graphics and Vision: Vol. 12, No. 4, pp 309–457. DOI: 10.1561/0600000096.

counter these attacks. Next, we move on to the issue of the source camera model and device identification, as well as the more recent problem of monitoring image and video sharing on social media. Finally, we look at the most recent challenge that has emerged in recent years: recognizing deepfakes, which we use to describe any content generated using artificial-intelligence techniques; we present the methods that have been introduced to show the existence of traces left in deepfake content and to detect them. For each problem, we also report the most popular metrics and datasets used today.

1

Introduction

Over the past years, online and multimedia content has passed traditional media as a preferred source of information, especially for young people (Richard Fletcher, 2020), and in the next few years visual content offered by social networks like Instagram could possibly overtake other platforms as a news source. The web and social media have favored the democratization of information and have allowed much more widespread dissemination of news (BBC-News, 2020). Although access to this content should have promoted the dissemination of reliable and validated content from multiple sources of information, the web and in particular social networks have also become a dangerous source of disinformation and dissemination of criminal content. Recently, fake videos of political leaders like Donald Trump, Vladimir Putin and North Korean leader Kim Jong-un have become increasingly realistic, opening up to the possibility of manipulating elections or public opinion (Staff, 2021 and Hao, 2020). Likewise, fake images and videos can be used for cyberbullying, military propaganda, or other criminal acts. All these problems have something in common. The widespread use of photo and video editing applications and the ease of use and retrieval of these tools have made multimedia manipulation a powerful instrument in the hands of criminals and attackers. Fake news, fake political campaigns, and porn videos, as well as fraud attempts are becoming much easier to spread and produce with a high level of realism. Distinguishing between fake and real is becoming an extremely important but difficult task. When multimedia contents are published on the web, they can easily go viral on social media. Also, *deepfakes*, which consist of fake content artificially generated typically using modern deep-learning approaches, have received a lot of attention in the last few years thanks to the high level of realism reached by this technology. Sophisticated deep-learning architectures such as autoencoders (AE) and generative adversarial networks (GANs) can be used to create highly realistic fake images and videos. Building trust and enabling the assessment of the authenticity of multimedia content is no longer an option but a real necessity.

The area of *multimedia forensics* combines principles and approaches from diverse research areas such as computer vision and signal processing, when it comes to addressing the authenticity and source of an image or a video. The three topics that multimedia forensics investigates mostly are the following: (1) *forgery detection*, which involves the detection of the authenticity of an image or video as well as of the presence of any manipulations; (2) *source identification*, which is the reconstruction of the history of some digital content, addressing which camera model, brand, or even specific device has captured that content, or whether it has been downloaded from social media; (3) *deepfake detection*, defining a *deepfake* as any synthetic medium accounting for the replacement of a person in an existing image or video with someone else's likeness (see Fig. 1 for instance). Figure 1.1 shows these three main problems.

Researchers have been studying the problem of forgery detection for more than twenty years now. Every day, thousands of professionals around the world use editing tools such as GIMP, Photoshop, Lightroom, After Effects Pro, and Final Cut Pro X as basic applications for their work. Multimedia forensic researchers have tried to provide an immediate response to all such applications, developing new tools to spot fake content. These methods can be used to detect subtle modifications, such as double compression or blurring, as well as more sophisticated attacks that could be used to change the semantic of a content. The most widespread examples of these manipulations are *splicing* (an object is

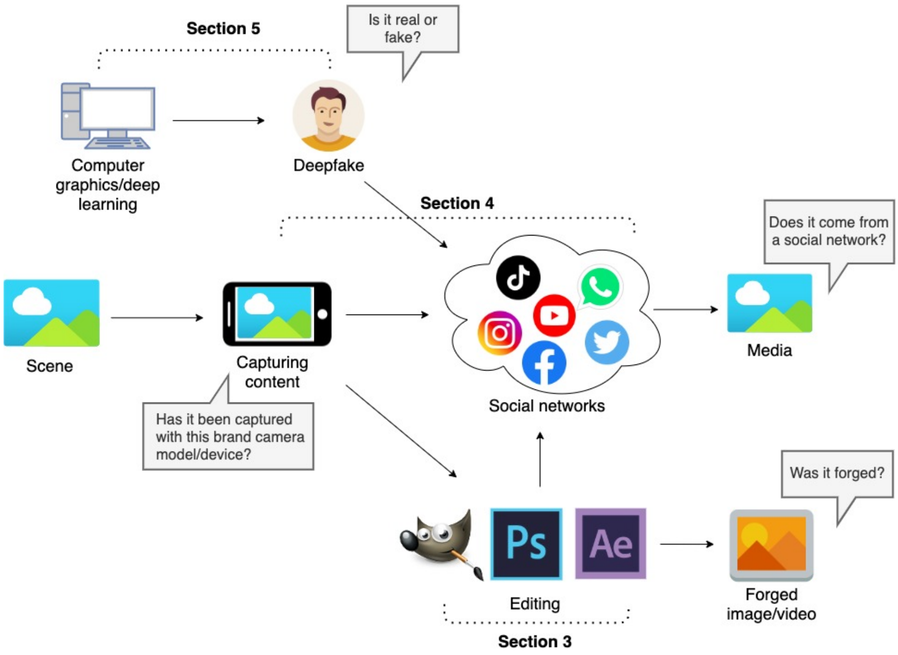


Figure 1.1: An overview of multimedia forensic investigations that we present in this work.

copied from an image and pasted into another picture), *copy–move* (the reproduction of an object into the same image), and *video-frame deletion* and *addition* in the case of video sequences. Recently, the advancements of artificially generated manipulations have attracted the attention of many researchers. *Deepfakes* are raising new alarms for the production of fake news, and their entry into the field of large technology giants has accelerated the design of new methods. Figure 1.2 shows some examples of the most recent fakes that spread out over the world.

Parallel to this problem, the identification of the source has been carefully studied as a forensic analysis tool. This becomes extremely important today in a hyper-connected world where information spreads all over the web. In some scenarios, multimedia content may constitute proof in the court proceedings and it becomes necessary to prove not only the authenticity of an image or video but also the source of the image or video itself. First of all, when it comes to assessing the authenticity of

Full text available at: http://dx.doi.org/10.1561/0600000096

(a) Fake Mark Zuckerberg *Bill Posters UK, Instagram page* 2019.

(b) Fake Barack Obama (left) and the actor who is impersonating him (right) *You Won't Believe What Obama Says in This Video!, Youtube video* 2018.

(c) An actor (left) and a fake Donald Trump (right) *Trump: Deepfakes Replacement, Youtube video* 2018.

Figure 1.2: Some of the most recent fakes that spread out over the world.

an image or video, the most advanced techniques for forgery detection allow to identify dishomogeneities in the considered image or video as well as any tampered features responsible for introducing differences from the original image/patch, especially any differences that are not so evident to the naked eye. Source identification can then be used to determine if the content was captured with a specific camera model or brand and even with a specific device. This can be done by exploiting the sequence of processes that a camera uses to convert the input light hitting the lens into an output image or video. This operation leaves important traces on the acquired files that can be used for forensic purposes. With the widespread adoption of social media and messaging applications, the task of deciding whether an image or video has been downloaded from these platforms has become important as well.

Forensic problems have been studied for a long time and they have been surveyed in multiple works such as Stamm *et al.* (2013), Verdoliva (2020), and Yang *et al.* (2020b). For years, researchers with different backgrounds have adopted signal-processing, computer-vision, and machine-learning techniques to solve the main challenges in this research field. Deep learning has recently come up with new designs that are capable of automatically learning both low- and high-level features to be analyzed to solve forensic problems.

In this survey, we present deep-learning methods for multimedia forensics, discussing the most important trends in both architectural and data-processing choices. We begin discussing different techniques used to manipulate content in Section 2. Next, we discuss image and video forgery techniques in Section 3. In Section 4, we review deep learning methods for source identification. Finally, in Section 5 we present the recent solutions for deepfake detection. Section 6 recaps the evaluation metrics considered throughout the cited works and Section 7 lists the datasets that have been mostly adopted for the above-mentioned tasks. Finally, in Section 8 we draw the conclusions.

8

Discussion and Conclusions

With the significant diffusion of fake multimedia content, research in computer vision and its applications in multimedia forensics (especially the deep learning based ones) have become a hot topic and received a great deal of attention. Meanwhile, the enormous amount of data we daily have access to has allowed us to generate highly realistic forged multimedia contents as well as to devise successful methods for automatically spotting such fakes.

This survey provides a comprehensive outlook on the literature on forgery detection to anomaly-based architectures, from source identification to deepfake detection, especially with respect to GAN-generated content. It is clear that deep-learning methods are progressively bridging the long-standing semantic gap between computable low-level visual features and high-level image features. Despite recent progress on punctual tasks, investigating and modeling complex real-world problems still remains challenging.

Given the necessity to tackle these issues for forensic purposes as well as the enormous profit potential relative to such applications, the studies on multimedia forensic tasks will continue to grow and expand: in this respect, the survey highlights the most promising directions for future research. First, as new and more complex generative manipulations and techniques emerge, simpler tools will become less effective. To address this problem, more complex multistream architectures have shown their potential. Therefore, more complex structures, tools, and data must be integrated to take advantage of all subtle information available to address multimedia-forensics problems. Along with the increasing complexity of media manipulation and generation techniques, the number of new tools and techniques being introduced makes it even more difficult to design deep-learning forgery-detection models that are robust to new attacks never seen before. In fact, despite the promising results, the main limitation of deep-neural networks originates from their high dependency on training data. The high number of operations (malevolent and innocent) that can be performed on an input, makes it practically impossible to reproduce all possible examples at training time. Consequently, higher robustness should be pursued by other means. Furthermore, to cope with rapid advances in manipulation technology, deep networks should be able to adapt to new manipulations, without complete retraining, which may simply be impossible because of lack of training data or lead to catastrophic forgetfulness. Still in this direction, the works reviewed in this survey, have been mostly applied in controlled settings. Thus, new techniques are needed to apply multimedia forensics in the wild. One attempt to cope with the complexity of the real world is to take into consideration multiple media at a time. For example, to decide on the authenticity of the news, we can rely not just on an image or video content, but also on the text or audio attached to it. In this direction, DARPA recently launched a new initiative on *semantic forensics*.[1] The challenge is not just to decide on the authenticity of an image or video, but to capture all semantic inconsistencies that can be discovered in a multimodal media asset. A multimodal approach can be particularly useful to detect deepfakes, where a video and an audio track are typically available. Also, semantic inconsistencies can be used in the future to detect anomalies on deepfakes of the entire human body, without examining only the human face.

[1] https://www.darpa.mil/program/semantic-forensics

One of the major current limitations of deep learning is their lack of interpretability. The complexity of deep learning-models makes it difficult to understand why they produce an output value. This problem is particularly relevant in multimedia forensics given the fact that they are often used for law-related applications. This means that it is often not sufficient that a classifier reports an image as fake or that a video is from a certain social network but to also report the features and the procedures that led to such an output. Furthermore, being able to interpret the logic of a deep neural network would allow to improve its design and training phase, and provide higher robustness with respect to malicious attacks. On a related issue, deep neural networks open up new vulnerabilities that can be exploited by an attacker. Despite the neural networks' ability to learn forensic features directly from data, intelligent attackers can use this to their advantage. Because the space of possible inputs to a neural network is substantially larger than the set of images used to train it, an attacker can create modified images that fall into an unseen space and force the neural network to misclassify. One method of accomplishing this involves introducing adversarial perturbations into an image (see Goodfellow *et al.*, 2015). With respect to this, GANs can become a new threat not just by generating very realistic images or videos, but also as counter forensics tools (see Barni *et al.*, 2018 for more details). They have already been used to remove forensic traces left by median filtering Kim *et al.*, 2018, and it is very likely that more GAN-based counter-forensic attacks will be developed in the near future.

Appendices

A

Computer Vision and Signal Processing for Media Forensics

Multimedia forensic is a research area that requires a basic understanding of computer-vision and signal-processing techniques. To facilitate the understanding of readers new to these two fields, in this section we want to introduce some basic background. Obviously, this section is not intended as an exhaustive treatment of these two disciplines, see the relevant books for more details (e.g., Goodfellow *et al.*, 2016). Specifically, in the next pages, we cover basic *deep-learning* topics for *computer-vision* applications and some basic *signal-processing* concepts that we refer to in the main text.

A.1 Deep-Learning Architectures for Computer Vision

Deep learning solves the fundamental problem in representation learning by learning representations that are expressed in terms of other, simpler forms. From a mathematical point of view, an artificial neural network is a mathematical function mapping some set of input values to output values. The function is constructed by composing many simpler functions. We can think of each application of a different mathematical function as providing a new representation of the input.

Feedforward neural networks are typically constructed by composing together many different functions, also informally called *neurons*. The model is associated with a directed acyclic graph describing how the functions are composed together. The network can be structured in several layers of neurons. The overall length of the chain gives the *depth* of the model. The first layer of a feedforward network is called the *input layer* and the last one the *output layer*. The layers in between the input and the output layers are called *hidden layers*. Each neuron in a layer typically performs two basic operations, a linear transformation and a nonlinear transformation. For example, given an input x, the output of a layer will be $\hat{y} = \sigma(W^T x + b)$, where $z = W^T x + b$ is a linear function and $\sigma(z)$ is a nonlinear function also called *activation function*.

In this section, we discuss different architecture choices and explain how each of these configurations can be most useful in solving a specific problem.

A.1.1 Fully Connected Networks

Fully connected networks (FCNs) are an essential method of deep learning. The main advantage of FCNs is that they are independent of the structure, that is, there is no need to make special assumptions about the input (for example, that the input consists of images or videos). They owe their name to the fact that each neuron in a certain layer is connected with all the neurons of the layer that precedes it and each neuron of the layer that follows it. As a result, these networks are fully connected. Figure A.1 shows an example of an FCN.

Although being independent of structure makes FCNs widely applicable, they tend to have lower performance than special networks tuned to the structure of a specific problem space. In fact, because of their structure, these networks are not robust to input data for which there is a two-dimensional or three-dimensional relationship such as images and videos. Furthermore, these networks do not take into account the dependence of input sequences such as text or video sequences. For these reasons, in computer-vision applications these networks are not commonly used to classify input features. Usually, these networks are used after a convolutional neural network or a recurrent neural network that work as feature extractors, that is, they learn how to extract relevant features that are useful to classify the input. Then, the FCN takes the feature vector as input and predicts the corresponding class.

Even if the FNCs are very often used as classifiers, it is still possible to apply them for regression problems or to train a network to project inputs into a latent space as happens, for example, in some applications that use Siamese networks.

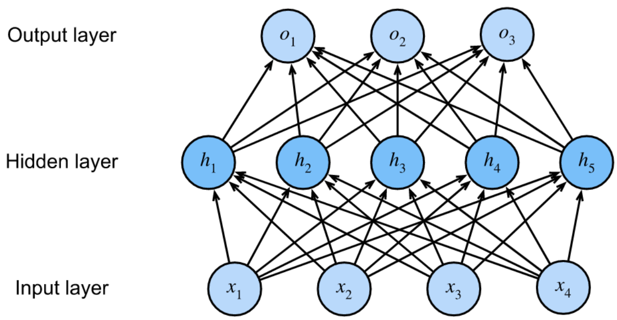


Figure A.1: An example of an FCN with a hidden layer of five hidden units (Zhang *et al.*, 2020).

A.1.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a specific kind of neural network for processing data that has a known grid-like structure. The most representative class of this family is image data, which can be thought of as a two-dimensional grid of pixels. These networks use a mathematical operation called *convolution* in place of a general matrix multiplication in at least one of their layers. Given a two-dimensional image I and a kernel K the convolution between I and K is defined as follows:

$$
\begin{aligned}
(I * K)(i,j) &= \sum_m \sum_n I(m,n)K(i-m, j-n) \\
&= \sum_m \sum_n I(i-m, j-n)K(m,n).
\end{aligned}
$$

Convolution leverages three important ideas that can help improve a computer-vision system: (1) sparse interactions, (2) parameter sharing, and (3) equivariant representations. Traditional neural-network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit, meaning that every output unit interacts with every input unit. CNNs, however, typically have sparse interactions (also referred to as sparse connectivity or sparse weights), which is accomplished by making the kernel size smaller than the input size. Thanks to this strategy, we can use the same parameters for more than one input unit in a model (also referred as parameter sharing). In a traditional neural network, each element of the weight matrix is used exactly once when computing the output of a layer. It is multiplied by one element of the input and then never reused. For CNNs, the particular form of parameter sharing causes the layer to have a property called equivariance to translation. To say a function is equivariant means that if the input changes, the output changes in the same way. Figure A.2 shows an example of a CNN.

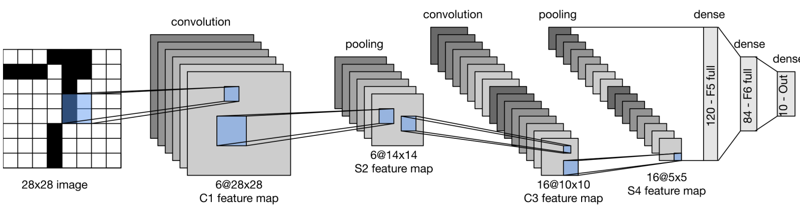


Figure A.2: Example of a CNN consisting of two convolutional layers; and a dense block consisting of three fully-connected layers (Zhang *et al.*, 2020; Lecun *et al.*, 1998).

A.1.3 Recurrent Neural Networks

Similarly to CNNs, *recurrent neural networks* (RNNs) are specialized neural networks for processing sequential data of the form $x^{(1)}, \ldots, x^{(t)}$. At each time step t, the state of a hidden unit h depends on its state at time $t-1$, that is:

$$\begin{aligned} h^{(t)} &= \sigma_h(W_{hh} \cdot h^{(t-1)} + W_{hx} \cdot x^{(t)} + b_h) \\ &= \sigma_h([W_{hh} W_{hx}] \cdot [h^{(t-1)} x^{(t)}] + b_h) \end{aligned}$$

where σ_h is a nonlinear (activation) function, $x^{(t)}$ represents the input at time t, W_{hh}, W_{hx} are the weight matrices associated to the actual hidden state $h^{(t-1)}$ and input $x^{(t)}$ respectively, and b_h a parameter vector. Forward propagation typically begins with a specification of the initial state $h^{(0)}$.

Depending on the problems on which they are applied, RNNs can be structured in different ways: (1) RNNs that generate an output at each time step and have recurrent connections between hidden units, (2) RNNs that produce an output at each time step and have recurrent connections only from the output at one time step to the hidden units at the next time step, and (3) RNNs with recurrent connections between hidden units, that read an entire sequence and then produce a single output. Figure A.3 shows an example of an RNN applied to character-level language processing.

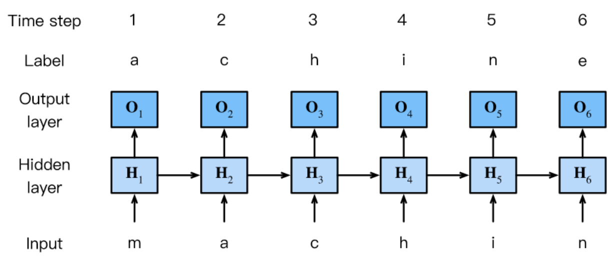


Figure A.3: Example character-level language RNN. The input and label sequences are *machin* and *achine*, respectively (Zhang *et al.*, 2020).

A.2 Common Deep Learning Backbones

Neural networks are often combined into complex design schemes that help them learn better the task they are solving. Every year, new architectures are published for solving new problems or achieving higher performance than previous models. In this section, we present some of the most common architectures used in the architectures of the survey. Obviously, our goal is not to provide an exhaustive discussion of all the backbones that can be used in computer vision or multimedia forensics, but to offer a quick guide to learn about the most used architectures in the works that we survey.

A.2.1 VGG

The VGG network (see Figure A.4) was designed by Simonyan and Zisserman, 2014. The input image passes through a stack of convolutional layers that use 3×3 filters, which is the smallest size to capture the notion of left/right, up/down, center. The convolution stride is fixed to 1 pixel and the padding is 1 pixel.[1] Each of the convolutional blocks is followed by a max-pooling layer which is performed over a 2×2 pixel window, with stride 2. The stack of convolutional layers (which can be constructed with different depths) is followed by three fully connected layers: the first two have 4096 channels each and the third has 1000 neurons corresponding to the output number of classes of the ImageNet dataset. The final layer is the softmax layer. In one of the configurations (VGG16), the network also uses 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by nonlinearity). All hidden layers are followed by ReLU activations. This network can be configured with different depths varying from 11 weight layers to 19 weight layers. The width of the convolutional layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512. Depending on the number of layers N, this network is typically referred to as VGGN. The most common configurations are VGG16 and VGG19.

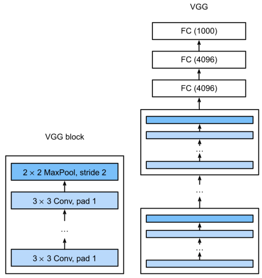


Figure A.4: Example of the VGG architecture from building blocks to the entire model (Zhang *et al.*, 2020).

[1] *Stride* and padding are parameters of CNNs; see Goodfellow *et al.*, 2016.

A.2.2 ResNet

He *et al.*, 2015 introduced *ResNets* (see Figure A.5) to solve the vanishing-gradient problem: When a neural network is too deep, the gradients are easily reduced to zero for the early layers of the network, with the result that the weights no longer update their values and, therefore, the model stops learning. The key idea is to use shortcut connections from early layers up to deeper (later) layers. Formally, denoting the desired underlying mapping as $H(x)$, we let the stacked nonlinear layers fit another mapping of $F(x) = H(x) - x$. The original mapping is recast into $F(x)+x$. The dimensions of x and $F(x)$ must be equal, thus the ResNet performs a linear projection W_s by the shortcut connections to match the dimensions:

$$y = F(x, \{W_i\}) + W_s \cdot x.$$

where $F(x, \{W_i\})$ represents the residual mapping to be learned. For example, it may represent two layers of the form $F = W_2 \cdot \sigma(W_1 \cdot x)$, in which σ denotes the ReLu function.

Skip connections between layers add the outputs from previous layers to the outputs of stacked layers. This allows information to be propagated to later levels without running into the problem of vanishing gradients thus allowing us to train deeper networks than was previously possible. He *et al.*, 2015 designed a plain network with 3×3 filters by following two simple design rules: (1) for the same output feature map size, the layers have the same number of filters and (2) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. The network performs downsampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-dimensional fully connected layer with softmax. Shortcut connections between layers increase the depth of the network. The ResNet network can be configured with different depths varying from 18 to 152 layers. Depending on the number of layers N, the network is typically referred to as ResNet-N. Very commonly, the network is used as ResNet-18, ResNet-50, or ResNet-100.

Figure A.5 shows two examples of residual blocks. ResNet follows VGG's convolutional layer design. The residual block has two 3×3

convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function. Then, a residual connection propagates the input of these two convolution operations directly before the final ReLU activation function. This kind of design requires that the output of the two convolutional layers has to be of the same shape as the input, so that they can be added together. To change the number of output channels, an additional 1×1 convolutional layer can be used to transform the input into the desired shape for the addition operation.

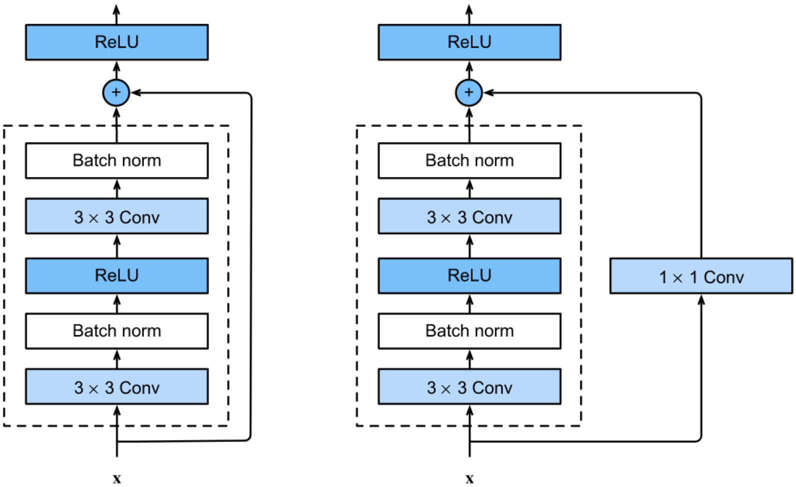


Figure A.5: Example ResNet blocks. A regular block (left) and a residual block (right) (Zhang *et al.*, 2020).

A.2.3 Inception

Parts of interest in an image can have extremely large variations in their size. This variety in the area of interest can make difficult the determination of the right kernel size for the convolution operation. A larger kernel is preferred for information that is distributed more globally, whereas a smaller kernel is preferred for information that is distributed more locally. The idea of the *inception network* (also known as *GoogLeNet*; see Szegedy *et al.*, 2014) is to have filters with multiple sizes operating on the same layer, called the *inception layer*. An inception layer performs a convolution on the input with three different kernel sizes: 1×1, 3×3, and 5×5. Additionally, max pooling is also performed in parallel to the filters. However, CNNs are computationally expensive. In GoogLeNet, 1×1 convolution is used as a dimensionality-reduction module to reduce the computation. By reducing the computation bottleneck, depth and width can be increased. Thus, Szegedy *et al.*, 2014 limit the number of input channels by adding an extra 1×1 convolution before the 3×3 and 5×5 convolutions. The 1×1 convolutions require much less computation than 5×5 convolutions, and applying them before the other filters reduces the size of input channels. The 1×1 convolution is also applied after the max-pooling layer. After that, all feature maps at different paths are concatenated together as the input of the next module. Figure A.6 shows an example of the inception block.

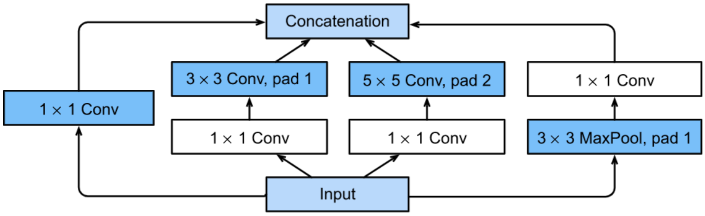


Figure A.6: Example of the structure of the inception block (Zhang *et al.*, 2020).

In GoogLeNet (Figure A.7), global average pooling is used at the end of network by averaging each feature map from 7×7 to 1×1.

The Inception network described so far is also known as Inception-v1. Subsequently, several enhancements of this version were introduced also known as Inception-v2 and Inception-v3 (Szegedy *et al.*, 2015), Inception-v4 and Inception-ResNet (Szegedy *et al.*, 2016).

A frequently used variation of Inception is called Xception (Chollet, 2016), which stands for *extreme inception*. In a traditional CNNs, convolutional layers seek out correlations across both space and depth. In Inception, 1×1 convolutions project the original input onto several separate, smaller input spaces, and from each of these input spaces some other type of filter transforms those smaller 3D blocks of data. Xception takes this one step further. Instead of partitioning input data into multiple compressed chunks, it maps the spatial correlations for

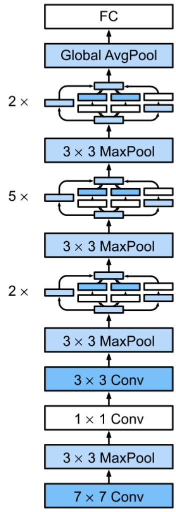


Figure A.7: The GoogLeNet architecture (Zhang *et al.*, 2020).

each output channel separately, and then performs a 1×1 depthwise convolution to capture cross-channel correlation. This is equivalent to an existing operation known as a *depthwise separable convolution*, which consists of a depthwise convolution (a spatial convolution performed independently for each channel) followed by a pointwise convolution (a 1×1 convolution across channels). See Chollet, 2016 for further information.

A.2.4 Long Short-Term Memory Networks

Long short-term memory networks (LSTMs) Hochreiter and Schmidhuber, 1997 are a special kind of RNN, capable of learning long-term dependencies. Typical RNNs suffer from short-term memory. If a sequence is long enough, they will have a hard time carrying information from earlier time steps to later ones. LSTMs are designed to avoid the long-term dependency problem. They have internal mechanisms called *gates* that can regulate the flow of information. These gates can learn what data in a sequence are important to keep or throw away. By doing that, they can pass relevant information down the long chain of sequences to make predictions.

The LSTM has four types of gates (see Figure A.8):

- *Forget gate* (F_t). This gate decides what information should be thrown away or kept. Information from the previous hidden state and information from the current input is passed through a sigmoid function.

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$

- *Input gate* (I_t). It decides what values will be updated. The previous hidden state and current input are passed into a sigmoid function. This decides what values will be updated by transforming them to be between 0 and 1.

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

- *Cell state* or *long-term memory* (C_t). The cell state is pointwise multiplied by the forget vector. This has the possibility of dropping values in the cell state if it is multiplied by values close to 0. Then it takes the output from the input gate and computes a pointwise addition with the candidate memory cell $\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$, which updates the cell state to new values that the neural network finds relevant.

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t.$$

- *Output gate* (O_t). It decides what the next hidden state should be. It passes the previous hidden state and the current input into a sigmoid function. Then it passes the newly modified cell state to the tanh function. The output of the tanh is multiplied with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state. The new cell state and the new hidden state are then carried over to the next time step.

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

Finally, the output hidden state can be simply calculated as $H_t = O_t \odot \tanh(C_t)$. If the output gate approximates 1 then it passes all memory information through to the predictor, whereas if the output gate is close to 0, it retains all the information only within the memory cell and performs no further processing.

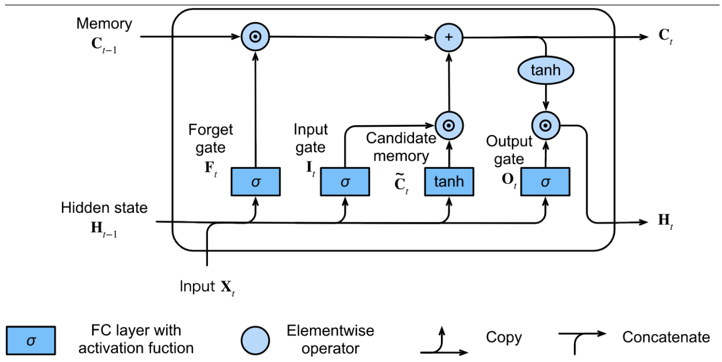


Figure A.8: Example of a hidden state in an LSTM model (Zhang *et al.*, 2020).

A.3 Signal Processing for Multimedia Forensics

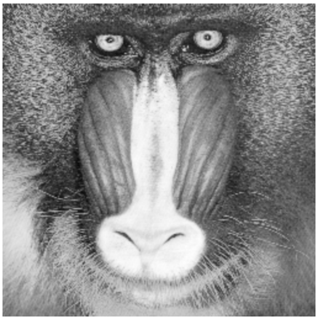
Signal processing is an important part of multimedia forensics. Indeed, image data can be represented as a signal that can be modeled by waves. For grayscale images, we can model them as a matrix of values, where the element at position (i, j) in the matrix corresponds to the pixel at position (i, j) in the image, and the value of that matrix element is the pixel's intensity. For example, 0 may correspond to black pixels, and 255 to white pixels. Pixel intensities between 0 and 255 are interpreted as colors between black and white. Figure A.9 shows an example applied on a grayscale image.

A similar approach can be used for color images modelling colors as separate signals or as a three-dimensional signal (one dimension for each color channel).

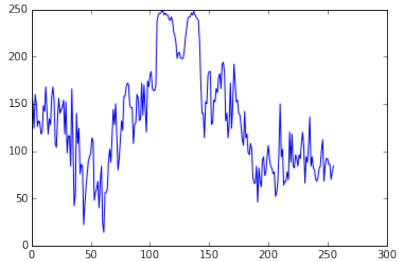
For most concepts (discrete Fourier transform, filters, etc.) consult textbooks on signal and image processing Vetterli *et al.*, 2018; Szeliski, 2011. Here we present some more specific concepts that may help in reading this survey.

A.3.1 Discrete Cosine Transform

Discrete cosine transform (DCT) is a signal-processing operation that expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is a type of



(a) Greyscale image (Shanker, 2021).



(b) Signal representation.

(c) Row of pixels from image A.9a.

Figure A.9: An example (Shanker, 2021) grayscale image (A.9a). Given a pixel's row of pixels extracted from the image ((A.9c)), it can be represented as a signal (A.9b).

Fourier-related transformation and is commonly used as a lossy compression technique. A Fourier transform is the process of decomposing a digital signal into the sum of some trigonometric functions. A Fourier transform is called a transform because it transforms the data from one form (the amplitude or pixel intensity over time) into a list of frequency coefficients controlling their contribution. The DCT has the property that most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. For this reason, in image processing applications, DCT is very often used as a form of lossy compression technique. As an example, the DCT is at the heart of the international standard JPEG and MPEG algorithms. In the frequency representation of an image, some of the higher frequency components, such as the smaller changes in amplitude leading up to peaks, are less important, and could be removed without losing visual components that are needed to understand the image content. Once that the image has been decomposed into a collection of trigonometric functions, it becomes easy to remove less important frequency functions that don't contribute as much to the core structures of the image.

The DCT is a linear transformation that transforms a vector of length n of pixel intensities (a row of pixels of an image), and returns a different vector of length n containing the coefficients for n different cosine functions. Thus, the vector is encoded by an $n \times n$ matrix, in which each row corresponds to a cosine function of a different frequency. Using n cosine functions is the key to being able to get our data back in terms of amplitudes after converting it to cosine coefficients. To represent each cosine wave as a row in the $n \times n$ matrix X, we compute it as:

$$X_{i,j} = \cos\left(\frac{\pi}{n} i \left(j + \frac{1}{2}\right)\right)$$

where i and j indicate rows and columns of the matrix respectively. In the equation above, each row corresponds to a different cosine function and the higher values of i correspond to cosine waves of higher frequency.

The last step, after calculating the DCT matrix, is to calculate the decomposition and the correct coefficients for each of the component waves. The decomposition can be easily computed by taking the dot product of the input vector of pixel intensities and X_i. The dot product of these two components can be interpreted as a measure of similarity between the two vectors, that is, if the pixel data is coincident with the values in one particular wave, it will be 0. Therefore, by computing this dot product, we can figure out what coefficient to use for that particular wave. This technique, can be similarly applied on two-dimensional matrices (i.e., two-dimensional image signals) by performing the DCT twice, once along the rows, and once along the columns.

To compress the image, we take the K most significant cosine waves in X_i, and save the coefficients. To get the compressed image back, we pad the matrix with 0s to get an $n \times n$ matrix (the original image's size), and then apply on it the inverse DCT transform to obtain the compressed image.

A.3.2 PRNU

When an photograph is taken by a camera, it is processed through a sequence of operations illustrated in Section 4. These operations may introduce noise and various imperfections to the image. Even if the imaging sensor takes a picture of an absolutely evenly lit scene, the resulting digital image will typically still exhibit small changes in intensity between individual pixels. This is partly because of the *shot* *noise* created by the electronic circuits, which is a random component, and partly because of the pattern noise created by the image sensors, a fixed component that remains approximately the same if multiple pictures of the exact same scene are taken. This implies that the pattern noise is impressed in every image the sensor takes and, thus, can be used for camera identification. Averaging over multiple images reduces the random components and enhances the pattern noise.

The two main ingredients of pattern noise are *fixed pattern noise* (FPN) and *photo-response nonuniformity* (PRNU). The FPN is caused by dark currents, that is, by pixel-to-pixel differences when the sensor array is not exposed to light. As it is an additive noise, very commonly, consumer cameras suppress it automatically by subtracting a dark frame from every image they take. Therefore, the dominant part of the pattern noise of an image is the PRNU. It is caused primarily by pixel nonuniformity (PNU), which is the different sensitivity of pixels to light caused by the inhomogeneity of silicon wafers and imperfections during the sensor manufacturing process. Because of its origin, it is unlikely that even sensors coming from the same wafer would exhibit correlated PNU patterns. So, the PNU noise is not affected by ambient temperature or humidity, but light refraction on dust particles and optical surfaces and zoom settings contribute to the PRNU noise. Since these low-frequency components are not a characteristic of the sensor, if we capture this noise pattern, we can create a distinctive link between a camera and its photos.

Formally, given a digital image I taken from camera a C, it can be modeled as:

$$I = I^{den} + I^{den}K + \theta$$

where I it the acquired image, I^{den} is the denoised image, K is the PRNU and θ represents other noise terms (e.g., shot noise). PRNU is usually estimated from N images captured with the same camera. The estimate can be computed with two simple steps: (1) the application of high-pass filtering $W_i = I_i - I_i^{den}$ on each image i, followed by (2) an estimate operation:

$$\hat{K} = \frac{\sum_{i=1}^{N} W_i \cdot I_i}{\sum_{i=1}^{N} (I_i)^2}.$$

The PRNU fingerprint $\hat{K}$ is obtained through a minimum variance estimator as indicated in the equation above, where N is the number of images used for the estimation.

A.3.3 JPEG Compression

JPEG is an acronym for *joint photographic experts group* and it refers to the *JPEG file interchange format* (JFIF). Usually, the files with the `.jpg` extension are JFIF files. It was created as a standard for digital image compression. JPEG is *lossy* compression technique, meaning that the image changes and loses some detail as a result of the compression. JPEG compression is actually composed of three different compression techniques, which are applied in successive layers: (1) chrominance subsampling, (2) DCT and quantization, and (3) delta, run-length, and Huffman encoding. Chrominance subsampling is the process of representing an image's color components at a lower resolution than its actual luminance components. This step is used to reduce the file size of colored images. For grayscale images, this step can be skipped. This step begins by converting the image from RGB to YUV color space. Because the human eye is more sensitive to luminance than to chrominance, typically JPEGs discard most of the chrominance information before any other compression takes place, so the image contains only half as much color information as it originally did. This first step already reduces the amount of information of the image to be stored. Next, the image is partitioned into 8×8 nonoverlapping pixel blocks and the DCT of each block is computed, resulting into a set of 64 subbands of DCT coefficients. The DCT coefficients are then quantized by dividing them by the entry in a quantization matrix that corresponds to the coefficient's subband and then rounding the resulting value to the nearest integer. Because the human visual system has different sensitivities to luminance and color distortions, different quantization tables are generally used to quantize the luminance and chrominance layers. Finally, each quantized DCT coefficient is converted to binary and then reordered into a single bit stream using the zigzag scan order [2]. The third and last compression

[2]See https://www.ece.ucdavis.edu/cerl/reliablejpeg/compression/ for further details.

layer is lossless. Initially, each DCT coefficient is converted from an absolute value to a relative value: Adjacent blocks in an image tend to have a high degree of correlation, so the protocol encodes the DCT term of a given block as a difference from the previous DCT term; the difference is typically a very a very small number and can be stored in a small number of bits—we call this encoding *delta encoding*. This process will typically create a lot of differences of value equal to zero. The next step encodes zeros into a *run-length encoding*, that is, it only stores the count of consecutive (differences of) zero values. Finally, the image is compressed with Huffman encoding, which is stored in the JPEG header.

MPEG (moving picture experts group) is a standard for video coding. It is used to compress video sequences and it is very similar to JPEG. The main difference with videos is that it also performs block-based motion compensation (see Sullivan *et al.*, 2012): it encodes the difference between each block and a predicted set of pixel values obtained from a shifted block in the previous frame. In fact, the encoder splits the video frame sequence into smaller segments called *group of pictures* (GOP). Each GOP starts with an *I-frame* which is an image independently encoded using a process similar to JPEG compression and continues with the predicted frames (*P-frames*) and bidirectional frames (*B-frames*). P-frames are predicted from preceding frames and B-frames can be predicted from I-frames or P-frames preceding or following them in the GOP. Check the MPEG official web page[3] of the MPEG group for further details.

In Section 2.1 you will find more details on how compression can be used in multimedia forensics applications.

[3] https://www.mpegstandards.org

B

Tables

B.1 Forgery Detection Methods

Table B.1: Deep learning architectures for image forgery detection.

Archit.	Convolutional layers					F.C. layers		DB	Ref.
	Preproc.	Input	Streams	Activ.	Pool.	Layers	Activ.		
RRU-Net	Gaussian noise, JPEG compr.	$384 \times 256 \times 3$	1 (27 layers)	ReLU, sigmoid	Max	—	—	Hsu and Chang 2006 Dong *et al.*, 2013	Bi *et al.*, 2019
DRN-C-26	Resizing, JPEG compr., brightness, contrast, saturation	400 or 700 in the shorter size	1 (26 layers)	ReLU	Max	—	—	Rössler *et al.*, 2019	Wang *et al.*, 2019a
BusterNet	—	$256 \times 256 \times 3$	2	—	Bilinear, per-centile	—	—	Tralic *et al.*, 2013 Dong *et al.*, 2013	Wu *et al.*, 2018
Multi-Task Fully Convolutional Network (MFCN)	—	—	2	ReLU	Max	—	—	Dong *et al.*, 2013 Hsu and Chang, 2006 Guan *et al.*, 2019 Carvalho *et al.*, 2013	Salloum *et al.*, 2018
Multi-domain CNN	DCT	$64 \times 64 \times 3$, 909×1	2	ReLU	Max	3	Softmax	Tralic *et al.*, 2013 Dong *et al.*, 2013	Amerini *et al.*, 2017b
Two-Stream Faster R-CNN	SRM filter	600 in the shorter size	2	ReLU	Bilinear, max	1	Softmax	Hsu and Chang, 2006 Dong *et al.*, 2013 Wen *et al.*, 2016 Guan *et al.*, 2019	Zhou *et al.*, 2018a
Two-Stream Neural Networks for Tampered Face Detection	patches, steganalysis	$299 \times 299 \times 3$, $3 \times 128 \times 128 \times 3$	2	ReLU	Global avg, max	—	—	Zhou *et al.*, 2018b	Zhou *et al.*, 2018b

Table B.1: (Continued) Deep learning architectures for image forgery detection

Archit.	**Convolutional layers**					**F.C. layers**		**DB**	**Ref.**
	Preproc.	**Input**	**Streams**	**Activ.**	**Pool.**	**Layers**	**Activ.**		
Hybrid LSTM and Encoder-Decoder	Laplacian filter, radon transform, FFT	$256 \times 256 \times 3$, $256 \times 256 \times 3$	2	ReLU	Max	—	—	Wen *et al.*, 2016 Guan *et al.*, 2019 Gloe and Böhme, 2010 Society, 2014	Bappy *et al.*, 2019
Forensic Similarity Network	Patches	$256 \times 256 \times 3$ / $128 \times 128 \times 3$	2	ReLU	Max	2	Tanh, sigmoid	Gloe and Böhme, 2010	Mayer and Stamm, 2020
ForensicTransfer	Third-order derivative	$256 \times 256 \times 3$	1	ReLU, tanh	—	—	—	Rössler *et al.*, 2018b Cozzolino *et al.*, 2018	Cozzolino *et al.*, 2018
ManTra-Net	—	$256 \times 256 \times 3$ or $512 \times 512 \times 3$	1	ReLU, L2norm, sigmoid	Max, avg, ZPool2D	—	—	Gloe and Böhme, 2010 Society, 2018 Bestagini, 2018 Wu *et al.*, 2019	Wu *et al.*, 2019

Table B.2: Deep learning architectures for video forgery detection.

Archit.	Convolutional layers					F.C. layers		DB	Ref.
	Preproc.	**Input**	**Streams**	**Activ.**	**Pool.**	**Layers**	**Activ.**		
TS-N	I-frames, P-frames	256×256	2	ReLU	Max, avg, global avg	3	ReLU, softmax	Montgomery *et al.*, 1994 Lin *et al.*, 2015 Almohamedh *et al.*, 2015	Nam *et al.*, 2019
C3D-based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot	Convert frames to motion residual images by means of an absolute difference algorithm	16-frames	1	ReLU	3D pool	2	Softmax	Long *et al.*, 2017	Long *et al.*, 2017
C2F-DCNN	—	64-frames	2	ReLU	Max, avg	—	—	Guan *et al.*, 2019 Long *et al.*, 2018	Long *et al.*, 2018
Video Codec Forensics Based on Convolutional Neural Networks	Patches	$64 \times 64 \times 3$	2	ReLU, SeLU	Max	1	SeLU, softmax	Verde *et al.*, 2018	Verde *et al.*, 2018

B.2 Source Camera Model Identification Methods

Table B.3: Deep learning architectures for source camera model identification.

Archit.	Convolutional layers					F.C. layers		DB	Ref.
	Preproc.	Input	Streams	Activ.	Pool.	Layers	Activ.		
Noiseprint	Patches	$48 \times 48 \times 3$	1	ReLU	—	—	—	Guan *et al.*, 2019 Zhou *et al.*, 2018b Carvalho *et al.*, 2013 Bianchi and Piva, 2012 Shullani *et al.*, 2017 Gloe and Böhme, 2010	Cozzolino and Verdoliva, 2020
Forensic Similarity Network	Patches	$256 \times 256 \times 3$ or $128 \times 128 \times 3$	2	ReLU	Max	2	Tanh, sigmoid	Gloe and Böhme, 2010	Mayer and Stamm, 2020
ACFM-based CNN	Green channel, MFR	$256 \times 256 \times 2$	1	—	Max, avg	3	Tanh, softmax	Gloe and Böhme, 2010	Bayar and Stamm, 2017a
Inception-Based Data-Driven Ensemble Approach to Camera Model identification	Patches as Bondi *et al.*, 2017b	$64 \times 64 \times 3$, $71 \times 71 \times 3$, $224 \times 224 \times 3$, $256 \times 256 \times 3$ or $299 \times 299 \times 3$	2	ReLU	Max, global avg	2	Softmax	Society, 2018 Bestagini, 2018	Ferreira *et al.*, 2018
Augmented convolutional feature maps for robust CNN-based camera model identification	Green channel, MFR	$256 \times 256 \times 2$	1	Tanh	Max, avg	3	Tanh, softmax	Gloe and Böhme, 2010 Bayar and Stamm, 2017a	Bayar and Stamm, 2017a
Content-adaptive fusion network	Patches	$64 \times 64 \times 3$	3	ReLU	Avg, global avg	—	Softmax	Gloe and Böhme, 2010 Yang *et al.*, 2017	Yang *et al.*, 2017
CNN-based fast source device identification	PRNU, noise residual as Chen *et al.*, 2008	from 80×80 to 720×720	1	Leaky ReLU	Max, pair-wise correlation pooling	1	—	Gloe and Böhme, 2010 Shullani *et al.*, 2017 Mandelli *et al.*, 2020b	Mandelli *et al.*, 2020b

B.3 Datasets

Table B.4: Image forgery detection datasets.

Dataset	Forgery	Auth./Forg.	Size	Format	Avail.	Year	Ref.
Columbia gray	Splicing	933-912	128×128	BMP	online	2004	Ng and Chang, 2004
Columbia color	Splicing	182 / 180	$757\times568 - 1152\times768$	BMP, TIF	online	2006	Hsu and Chang, 2006
MICC F2000	Copy-move	110 / 110	2048×1536	JPEG	online	2011	Amerini *et al.*, 2011
Erlangen	Copy-move	—	3000×2300	JPEG	online	2013	Christlein *et al.*, 2012
CASIA 1	Splicing, copy-move	800 / 921	374×256	JPEG	—	2013	Dong *et al.*, 2013
CASIA 2	Splicing, copy-move	7,200 / 5,123	$320\times240 - 800\times600$	JPEG, BMP, TIF	-	2013	Dong *et al.*, 2013
CoMoFoD	Copy-move	— / 260	$512 \times 512 - 3000 \times 2000$	JPEG	-	2013	Tralic *et al.*, 2013
COVERAGE	Copy-move	100 / 100	400×486	TIF	online	2016	Wen *et al.*, 2016
NIST NC2016	Splicing, copy-move, removal	560 / 564	$500 \times 500 - 5,616 \times 3,744$	JPEG	upon request	2016	Guan *et al.*, 2019
NIST NC2017	Various	2667 / 1410	$160 \times 120 - 8000 \times 5320$	RAW, PNG, BMP, JPEG	upon request	2017	Guan *et al.*, 2019
FaceSwap	Face swapping	1,758 / 1,927	450×338-7360×4912	JPEG	-	2017	Zhou *et al.*, 2018b
NIST NC2018	Various	14,156 / 3,265	$128 \times 104 - 7952 \times 5304$	RAW, PNG, BMP, JPEG	upon request	2017	Guan *et al.*, 2019
PS Battles	Various	11,142 / 102,028	$130 \times 60 - 10,000 \times 8558$	PNG, JPEG	online	2018	Heller *et al.*, 2018
Synthetic Dataset	Splicing, copy-move	— / 170,000	—	PNG, BMP, JPEG, TIF	online	2019	Bappy *et al.*, 2019
NIST NC2019	Various	10,279 / 5,750	$128 \times 104 - 7952 \times 5304$	PNG	upon request	2019	Guan *et al.*, 2019

Table B.5: Source identification datasets.

Dataset	**Cam.**	**Dev.**	**Size**	**Format**	**Social**	**Avail.**	**Year**	**Ref.**
Dresden	25	73	14,000	JPEG	None	online	2010	Gloe and Böhme, 2010
RAISE	4	—	8,156	RAW	None	online	2015	Dang-Nguyen *et al.*, 2015
Image Ballistic and Social Networks	—	—	2,720	JPEG	Facebook, Google+, Twitter, Flickr, Instagram, Tumblr, Imgur, Tinypic, Whatsapp, Telegram	online	2016	Giudice *et al.*, 2016
UCID	—	—	30,000	JPEG	Flickr, Facebook, Twitter	online	2017	Caldelli *et al.*, 2017
PUBLIC	—	—	3,000	JPEG	Flickr, Facebook, Twitter	online	2017	Caldelli *et al.*, 2017
VISION	35	35	34,427 images / 1,914 videos	JPEG, mp4	Facebook, Youtube, Whatsapp	online	2017	Shullani *et al.*, 2017
IEEE SPS Camera Model identification	10	20	3,025	JPEG	None	online	2018	Society, 2018, Bestagini, 2018

Table B.6: Deepfakes datasets.

Dataset	**Forgery**	**Size**	**Format**	**Availab.**	**Year**	**Ref.**
DeepfakeTIMIT	Deepfake	— / 620	JPEG	upon request	2018	Korshunov and Marcel, 2018
Fake video corpus (FVC)	Various	2,458 / 3,957	—	online	2018	Papadopoulou *et al.*, 2018
Fake Faces in the Wild (FFW)	Deepfake, splicing, CGI	— / 150	H264, Youtube	online	2018	Khodabakhsh *et al.*, 2018
FaceForensics++	Deepfake, CGI	1,000 / 4,000	H264 crf 0/23/40	online	2019	Rössler *et al.*, 2019
DeepFake Detection Dataset	Deepfake	363 / 3,068	H264 crf 0/23/40	online	2019	Nick Dufour, 2019
Celeb-DF	Deepfake	590 / 5,639	M4PEG	online	2019	Li *et al.*, 2020
Deepfake Detection Challenge (DFDC)	Deepfake	19,154 / 100,000	H264	online	2019	AWS, Facebook, Microsoft, Partnership on AI's Media Integrity Steering Committee, 2020
DeeperForensics-1.0	Deepfake	50,000 / 10,000	—	online	2020	Jiang *et al.*, 2020

References

Afchar, D., V. Nozick, J. Yamagishi, and I. Echizen. (2018). "MesoNet: a Compact Facial Video Forgery Detection Network". *CoRR*. abs/1809.00888. arXiv: 1809.00888. URL: http://arxiv.org/abs/1809.00888.

Agarwal, S. and L. R. Varshney. (2019). "Limits of Deepfake Detection: A Robust Estimation Viewpoint". *CoRR*. abs/1905.03493. arXiv: 1905.03493. URL: http://arxiv.org/abs/1905.03493.

Agarwal, S., H. Farid, Y. Gu, M. He, K. Nagano, and H. Li. (2019). "Protecting World Leaders Against Deep Fakes". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Almohamedh, H., F. Qurashi, and I. Kostanic. (2015). "Mobile Video Quality Prediction (MVQP) for Long Term Evolution (LTE)". *IAENG International Journal of Computer Science*. 42(Feb.): 46–53.

Altman, N. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". English (US). *American Statistician*. 46(3): 175–185. DOI: 10.1080/00031305.1992.10475879.

Amerini, I., L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. (2011). "A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery". *IEEE Transactions on Information Forensics and Security*. 6(3): 1099–1110. DOI: 10.1109/TIFS.2011.2129512.

Amerini, I., C. Li, and R. Caldelli. (2019a). "Social Network Identification Through Image Classification With CNN". *IEEE Access*. 7: 35264–35273. DOI: 10.1109/ACCESS.2019.2903876.

Amerini, I., A. Anagnostopoulos, L. Maiano, and L. Ricciardi Celsi. (2021). "Learning Double-Compression Video Fingerprints Left from Social-Media Platforms".

Amerini, I., R. Caldelli, A. D. Mastio, A. D. Fuccia, C. Molinari, and A. P. Rizzo. (2017a). "Dealing with video source identification in social networks". *Signal Processing: Image Communication*. 57: 1–7. DOI: https://doi.org/10.1016/j.image.2017.04.009.

Amerini, I., L. Galteri, R. Caldelli, and A. Del Bimbo. (2019b). "Deepfake Video Detection through Optical Flow Based CNN". In: *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

Amerini, I., T. Uricchio, L. Ballan, and R. Caldelli. (2017b). "Localization of JPEG double compression through multi-domain convolutional neural networks". *CoRR*. abs/1706.01788. arXiv: 1706.01788. URL: http://arxiv.org/abs/1706.01788.

Amerini, I., T. Uricchio, and R. Caldelli. (2017c). "Tracing images back to their social network of origin: A CNN-based approach". *2017 IEEE Workshop on Information Forensics and Security (WIFS)*: 1–6.

AWS, Facebook, Microsoft, Partnership on AI's Media Integrity Steering Committee. (2020). "Deepfake Detection Challenge: Identify videos with facial or voice manipulations". URL: https://www.kaggle.com/c/deepfake-detection-challenge/.

Bakas, J. and R. Naskar. (2018). "A Digital Forensic Technique for Inter-Frame Video Forgery Detection Based on 3D CNN". DOI: 10.1007/978-3-030-05171-6-16.

Baldini, G. and I. Amerini. (2019). "Smartphones Identification Through the Built-In Microphones With Convolutional Neural Network". *IEEE Access*. 7: 158685–158696. DOI: 10.1109/ACCESS.2019.2950859.

Baldini, G., G. Steri, I. Amerini, and R. Caldelli. (2017). "The identification of mobile phones through the fingerprints of their built-in magnetometer: An analysis of the portability of the fingerprints". In: *2017 International Carnahan Conference on Security Technology (ICCST)*. 1–6. DOI: 10.1109/CCST.2017.8167855.

Bappy, J. H., A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath. (2017). "Exploiting Spatial Structure for Localizing Manipulated Image Regions". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 4980–4989. DOI: 10.1109/ICCV.2017.532.

Bappy, J. H., C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury. (2019). "Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries". *IEEE Transactions on Image Processing*. 28(7): 3286–3300. DOI: 10.1109/TIP.2019.2895466.

Barni, M., Q.-T. Phan, and B. Tondi. (2019). "Copy Move Source-Target Disambiguation through Multi-Branch CNNs".

Barni, M., M. C. Stamm, and B. Tondi. (2018). "Adversarial Multimedia Forensics: Overview and Challenges Ahead". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 962–966. DOI: 10.23919/EUSIPCO.2018.8553305.

Bas, P., T. Filler, and T. Pevný. (2011). ""Break Our Steganographic System": The Ins and Outs of Organizing BOSS". In: *Information Hiding*. Ed. by T. Filler, T. Pevný, S. Craver, and A. Ker. Berlin, Heidelberg: Springer Berlin Heidelberg. 59–70.

Bayar, B. and M. C. Stamm. (2017a). "Augmented convolutional feature maps for robust CNN-based camera model identification". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 4098–4102.

Bayar, B. and M. C. Stamm. (2017b). "On the robustness of constrained convolutional neural networks to JPEG post-compression for image resampling detection". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2152–2156.

Bayar, B. and M. C. Stamm. (2018). "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection". *IEEE Transactions on Information Forensics and Security*. 13(11): 2691–2706.

Bayar, Belhassen, Stamm, and M. C. (2017). “Design Principles of Convolutional Neural Networks for Multimedia Forensics”. DOI: https://doi.org/10.2352/ISSN.2470-1173.2017.7.MWSF-328.

Bayar, B. and M. C. Stamm. (2016). “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. IH&MMSec ’16*. Vigo, Galicia, Spain: Association for Computing Machinery. 5–10. DOI: 10.1145/2909827.2930786.

Bazarevsky, V., Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann. (2019). “BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs”. arXiv: 1907.05047 [cs.CV].

BBC-News. (2020). “Instagram will overtake Twitter as a news source”. URL: https://www.bbc.com/news/technology-53050959.

Bestagini, M. C. S. P. (2018). “IEEE Signal Processing Cup 2018 Database - Forensic Camera Model Identification”. DOI: 10.21227/H2XM2P.

Bi, X., Y. Wei, B. Xiao, and W. Li. (2019). “RRU-Net: The Ringed Residual U-Net for Image Splicing Forgery Detection”. In: *CVPR Workshops*.

Bianchi, T. and A. Piva. (2012). “Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts”. *IEEE Transactions on Information Forensics and Security*. 7(3): 1003–1017.

Bill Posters UK, Instagram page. (2019). URL: https://www.instagram.com/p/BypkGIvFfGZ/?utm_source=ig_embed&utm_campaign=embed_video_watch_again.

Bondi, L., L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro. (2017a). “First Steps Toward Camera Model Identification With Convolutional Neural Networks”. *IEEE Signal Processing Letters*. 24(3): 259–263. DOI: 10.1109/LSP.2016.2641006.

Bondi, L., E. D. Cannas, P. Bestagini, and S. Tubaro. (2020). “Training Strategies and Data Augmentations in CNN-based DeepFake Video Detection”. arXiv: 2011.07792 [cs.CV].

Bondi, L., D. Güera, L. Baroffio, P. Bestagini, E. Delp, and S. Tubaro. (2017b). "A Preliminary Study on Convolutional Neural Networks for Camera Model Identification". DOI: 10.2352/ISSN.2470-1173.2017.7.MWSF-327.

Bonettini, N., E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro. (2020). "Video Face Manipulation Detection Through Ensemble of CNNs". arXiv: 2004.07676 [cs.CV].

Bromley, J., J. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Sackinger, and R. Shah. (1993). "Signature Verification using a "Siamese" Time Delay Neural Network". *International Journal of Pattern Recognition and Artificial Intelligence*. 7(Aug.): 25. DOI: 10.1142/S0218001493000339.

Burt, P. and E. Adelson. (1983). "The Laplacian Pyramid as a Compact Image Code". *IEEE Transactions on Communications*. 31(4): 532–540.

Caldelli, R., I. Amerini, and C. T. Li. (2018a). "PRNU-based Image Classification of Origin Social Network with CNN". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 1357–1361.

Caldelli, R., R. Becarelli, and I. Amerini. (2017). "Image Origin Classification Based on Social Network Provenance". *IEEE Transactions on Information Forensics and Security*. 12(6): 1299–1308.

Caldelli, R., I. Amerini, and C.-T. Li. (2018b). "PRNU-based Image Classification of Origin Social Network with CNN". DOI: 10.23919/EUSIPCO.2018.8553160.

Caldelli, R., L. Galteri, I. Amerini, and A. Del Bimbo. (2021). "Optical Flow based CNN for detection of unlearnt deepfake manipulations". *Pattern Recognition Letters*. 146: 31–37. DOI: https://doi.org/10.1016/j.patrec.2021.03.005.

Carreira, J. and A. Zisserman. (2017). "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". *CoRR*. abs/1705.07750. arXiv: 1705.07750. URL: http://arxiv.org/abs/1705.07750.

Carvalho, T. J. d., C. Riess, E. Angelopoulou, H. Pedrini, and A. d. R. Rocha. (2013). "Exposing Digital Image Forgeries by Illumination Color Classification". *IEEE Transactions on Information Forensics and Security*. 8(7): 1182–1194.

Chen, M., J. Fridrich, M. Goljan, and J. Lukas. (2008). "Determining Image Origin and Integrity Using Sensor Noise". *IEEE Transactions on Information Forensics and Security*. 3(1): 74–90.

Chen, M., J. Fridrich, M. Goljan, and J. Lukás. (2007). "Source digital camcorder identification using sensor photo response non-uniformity - art. no. 65051G". *SPIE*. 6505(Feb.). DOI: 10.1117/12.696519.

Chen, S., S. Tan, B. Li, and J. Huang. (2016). "Automatic Detection of Object-Based Forgery in Advanced Video". *IEEE Transactions on Circuits and Systems for Video Technology*. 26: 2138–2151.

Cho, K., B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". *CoRR*. abs/1406.1078. arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078.

Choi, Y., M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. (2017). "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation". arXiv: 1711.09020 [cs.CV].

Chollet, F. (2016). "Xception: Deep Learning with Depthwise Separable Convolutions". *CoRR*. abs/1610.02357. arXiv: 1610.02357. URL: http://arxiv.org/abs/1610.02357.

Christlein, V., C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. (2012). "An Evaluation of Popular Copy-Move Forgery Detection Approaches". *CoRR*. abs/1208.3665. arXiv: 1208.3665. URL: http://arxiv.org/abs/1208.3665.

Ciftci, U. A. and I. Demir. (2019). "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals". *CoRR*. abs/1901.02212. arXiv: 1901.02212. URL: http://arxiv.org/abs/1901.02212.

Cortes, C. and V. Vapnik. (1995). "Support-Vector Networks." *Mach. Learn.* 20(3): 273–297. URL: http://dblp.uni-trier.de/db/journals/ml/ml20.html#CortesV95.

Cozzolino, D., D. Gragnaniello, and L. Verdoliva. (2014a). "Image forgery detection through residual-based local descriptors and block-matching". In: *2014 IEEE International Conference on Image Processing (ICIP)*. 5297–5301.

Cozzolino, D., D. Gragnaniello, and L. Verdoliva. (2014b). "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques". In: *2014 IEEE International Conference on Image Processing (ICIP)*. 5302–5306.

Cozzolino, D., G. Poggi, and L. Verdoliva. (2015a). "Splicebuster: A new blind image splicing detector". In: *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6. DOI: 10.1109/WIFS.2015.7368565.

Cozzolino, D. and L. Verdoliva. (2016). "Single-image splicing localization through autoencoder-based anomaly detection". In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

Cozzolino, D. and L. Verdoliva. (2018a). "Camera-based Image Forgery Localization using Convolutional Neural Networks". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 1372–1376.

Cozzolino, D. and L. Verdoliva. (2020). "Noiseprint: A CNN-Based Camera Model Fingerprint". *IEEE Transactions on Information Forensics and Security*. 15: 144–159. DOI: 10.1109/TIFS.2019.2916364.

Cozzolino, D., F. Marra, D. Gragnaniello, G. Poggi, and L. Verdoliva. (2020). "Combining PRNU and noiseprint for robust and efficient device source identification". *EURASIP Journal on Information Security*. Jan. DOI: 10.1186/s13635-020-0101-7.

Cozzolino, D., G. Poggi, and L. Verdoliva. (2015b). "Copy-move forgery detection based on PatchMatch". *2014 IEEE International Conference on Image Processing, ICIP 2014*. Jan.: 5312–5316. DOI: 10.1109/ICIP.2014.7026075.

Cozzolino, D., G. Poggi, and L. Verdoliva. (2017). "Recasting Residual-based Local Descriptors as Convolutional Neural Networks: an Application to Image Forgery Detection". arXiv: 1703.04615 [cs.CV].

Cozzolino, D., J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. (2018). "ForensicTransfer: Weakly-supervised Domain Adaptation for Forgery Detection". *CoRR*. abs/1812.02510. arXiv: 1812.02510. URL: http://arxiv.org/abs/1812.02510.

Cozzolino, D. and L. Verdoliva. (2018b). "Camera-based Image Forgery Localization using Convolutional Neural Networks". *CoRR*. abs/1808.09714. arXiv: 1808.09714. URL: http://arxiv.org/abs/1808.09714.

Cozzolino Giovanni Poggi Luisa Verdoliva, D. (2019). "Extracting camera-based fingerprints for video forensics". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

D'Avino, D., D. Cozzolino, G. Poggi, and L. Verdoliva. (2017). "Autoencoder with recurrent neural networks for video forgery detection". *Electronic Imaging*. 2017(Jan.): 92–99. DOI: 10.2352/ISSN.2470-1173.2017.7.MWSF-330.

Dang, H., F. Liu, J. Stehouwer, X. Liu, and A. Jain. (2019). "On the Detection of Digital Face Manipulation". arXiv: 1910.01717 [cs.CV].

Dang-Nguyen, D.-T., C. Pasquini, V. Conotter, and G. Boato. (2015). "RAISE: A Raw Images Dataset for Digital Image Forensics". In: *Proceedings of the 6th ACM Multimedia Systems Conference. MMSys '15*. Portland, Oregon: Association for Computing Machinery. 219–224. DOI: 10.1145/2713168.2713194.

Davis, J. and M. Goadrich. (2006). "The Relationship between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning. ICML '06*. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery. 233–240. DOI: 10.1145/1143844.1143874.

de Rezende, E. R. S., G. C. S. Ruppert, and T. Carvalho. (2017). "Detecting Computer Generated Images with Deep Convolutional Neural Networks". In: *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 71–78.

"Deepfakes Faceswap". (2018). URL: https://github.com/deepfakes/faceswap/.

Ding, X., Y. Chen, Z. Tang, and Y. Huang. (2019). "Camera Identification Based on Domain Knowledge-Driven Deep Multi-Task Learning". *IEEE Access*. 7: 25878–25890.

Do Nhu, T., I. Na, and S. Kim. (2018). "Forensics Face Detection From GANs Using Convolutional Neural Network".

Dolhansky, B., J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. (2020). "The DeepFake Detection Challenge (DFDC) Dataset". arXiv: 2006.07397 [cs.CV].

Donahue, J., L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. (2016). "Long-term Recurrent Convolutional Networks for Visual Recognition and Description". arXiv: 1411.4389 [cs.CV].

Dong, J., W. Wang, and T. Tan. (2013). "CASIA Image Tampering Detection Evaluation Database". In: *2013 IEEE China Summit and International Conference on Signal and Information Processing*. 422–426. DOI: 10.1109/ChinaSIP.2013.6625374.

Du, M., S. Pentyala, Y. Li, and X. Hu. (2019). "Towards Generalizable Forgery Detection with Locality-aware AutoEncoder". arXiv: 1909.05999 [cs.CV].

Ekman, P. and W. V. Friesen. (1976). "Measuring facial movement". *Environmental psychology and nonverbal behavior*. 1(1): 56–75. DOI: 10.1007/BF01115465.

Everingham, M., L. Gool, C. K. Williams, J. Winn, and A. Zisserman. (2010). "The Pascal Visual Object Classes (VOC) Challenge". *Int. J. Comput. Vision*. 88(2): 303–338. DOI: 10.1007/s11263-009-0275-4.

"Fake App". (2018). URL: https://www.fakeapp.com/.

Fernandes, S., S. Raj, E. Ortiz, I. Vintila, M. Salter, G. Urosevic, and S. Jha. (2019). "Predicting Heart Rate Variations of Deepfake Videos using Neural ODE". In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 1721–1729.

Fernando, T., C. Fookes, S. Denman, and S. Sridharan. (2019). "Exploiting Human Social Cognition for the Detection of Fake and Fraudulent Faces via Memory Networks". arXiv: 1911.07844 [cs.CV].

Ferreira, A., H. Chen, B. Li, and J. Huang. (2018). "An Inception-Based Data-Driven Ensemble Approach to Camera Model Identification". In: *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–7.

Freire-Obregón, D., F. Narducci, S. Barra, and M. Castrillón-Santana. (2019). "Deep learning for source camera identification on mobile devices". *Pattern Recognition Letters*. 126(Sept.): 86–91. DOI: 10.1016/j.patrec.2018.01.005.

Fridrich, J. and J. Kodovsky. (2012). "Rich Models for Steganalysis of Digital Images". *IEEE Transactions on Information Forensics and Security*. 7(3): 868–882.

Fridrich, J. J. (2009). "Digital Image Forensics Using Sensor Noise".

Fuji Tsang, C. and J. Fridrich. (2018). "Steganalyzing Images of Arbitrary Size with CNNs". In: vol. 2018. DOI: 10.2352/ISSN.2470-1173.2018.07.MWSF-121.

Galdi, C., M. Nappi, J.-L. Dugelay, and Y. Yu. (2018). "Exploring New Authentication Protocols for Sensitive Data Protection on Smartphones". *IEEE Communications Magazine*. 56(Jan.): 136–142. DOI: 10.1109/MCOM.2017.1700342.

Gatys, L. A., A. S. Ecker, and M. Bethge. (2015). "A Neural Algorithm of Artistic Style". *CoRR*. abs/1508.06576. arXiv: 1508.06576. URL: http://arxiv.org/abs/1508.06576.

Giudice, O., A. Paratore, M. Moltisanti, and S. Battiato. (2016). "A Classification Engine for Image Ballistics of Social Data". *CoRR*. abs/1610.06347. arXiv: 1610.06347. URL: http://arxiv.org/abs/1610.06347.

Gloe, T. and R. Böhme. (2010). "The Dresden Image Database for Benchmarking Digital Image Forensics". *Journal of Digital Forensic Practice*. 3(2-4): 150–159. DOI: 10.1080/15567281.2010.531500. eprint: https://doi.org/10.1080/15567281.2010.531500.

Goljan, M., J. Fridrich, and T. Filler. (2009). "Large scale test of sensor fingerprint camera identification". In: *Media Forensics and Security*. Ed. by E. J. D. III, J. Dittmann, N. D. Memon, and P. W. Wong. Vol. 7254. International Society for Optics and Photonics. SPIE. 170–181. DOI: 10.1117/12.805701.

Goodfellow, I., Y. Bengio, and A. Courville. (2016). *Deep Learning*. MIT Press.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. (2014). "Generative Adversarial Networks". arXiv: 1406.2661 [stat.ML].

Goodfellow, I. J., J. Shlens, and C. Szegedy. (2015). "Explaining and Harnessing Adversarial Examples". arXiv: 1412.6572 [stat.ML].

Guan, H., M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. (2019). “MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation”. In: *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. 63–72. DOI: 10.1109/WACVW.2019.00018.

Guera, D. and E. J. Delp. (2018). “Deepfake Video Detection Using Recurrent Neural Networks”. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*: 1–6.

Hadwiger, B. and C. Riess. (2020). “The Forchheim Image Database for Camera Identification in the Wild”. arXiv: 2011.02241 [cs.CV].

Hao, K. (2020). “Deepfake Putin is here to warn Americans about their self-inflicted doom”. URL: https://www.technologyreview.com/2020/09/29/1009098/ai-deepfake-putin-kim-jong-un-us-election/.

He, K., X. Zhang, S. Ren, and J. Sun. (2014). “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”. *CoRR*. abs/1406.4729. arXiv: 1406.4729. URL: http://arxiv.org/abs/1406.4729.

He, K., X. Zhang, S. Ren, and J. Sun. (2015). “Deep Residual Learning for Image Recognition”. *CoRR*. abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

He, P., X. Jiang, T. Sun, and H. Li. (2018). “Computer Graphics Identification Combining Convolutional and Recurrent Neural Networks”. *IEEE Signal Processing Letters*. 25(9): 1369–1373.

He, P., H. Li, and H. Wang. (2019). “Detection of Fake Images Via The Ensemble of Deep Representations from Multi Color Spaces”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2299–2303.

He, P., X. Jiang, T. Sun, S. Wang, B. Li, and Y. Dong. (2017). “Frame-wise detection of relocated I-frames in double compressed H.264 videos based on convolutional neural network”. *Journal of Visual Communication and Image Representation*. 48: 149–158. DOI: https://doi.org/10.1016/j.jvcir.2017.06.010.

Heller, S., L. Rossetto, and H. Schuldt. (2018). “The PS-Battles Dataset - an Image Collection for Image Manipulation Detection”. arXiv: 1804.04866 [cs.MM].

Hochreiter, S., Y. Bengio, P. Frasconi, and J. Schmidhuber. (2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". *A Field Guide to Dynamical Recurrent Neural Networks*.

Hochreiter, S. and J. Schmidhuber. (1997). "Long Short-term Memory". *Neural computation*. 9(Dec.): 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

Hsu, Y.-F. and S.-F. Chang. (2006). "Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency". In: *International Conference on Multimedia and Expo*. Toronto, Canada.

Hu, J., L. Shen, and G. Sun. (2017). "Squeeze-and-Excitation Networks". *CoRR*. abs/1709.01507. arXiv: 1709.01507. URL: http://arxiv.org/abs/1709.01507.

Huang, G., Z. Liu, and K. Q. Weinberger. (2016). "Densely Connected Convolutional Networks". *CoRR*. abs/1608.06993. arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993.

Huh, M., A. Liu, A. Owens, and A. A. Efros. (2018). "Fighting Fake News: Image Splice Detection via Learned Self-Consistency". *CoRR*. abs/1805.04096. arXiv: 1805.04096. URL: http://arxiv.org/abs/1805.04096.

Jaderberg, M., K. Simonyan, A. Zisserman, and K. Kavukcuoglu. (2015). "Spatial Transformer Networks". *CoRR*. abs/1506.02025. arXiv: 1506.02025. URL: http://arxiv.org/abs/1506.02025.

Jiang, L., R. Li, W. Wu, C. Qian, and C. C. Loy. (2020). "DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection". In: *CVPR*.

Kalkowski, S., C. Schulze, A. Dengel, and D. Borth. (2015). "Real-Time Analysis and Visualization of the YFCC100m Dataset". In: *Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions. MMCommons '15*. Brisbane, Australia: Association for Computing Machinery. 25–30. DOI: 10.1145/2814815.2814820.

Kang, X., M. C. Stamm, A. Peng, and K. J. R. Liu. (2013). "Robust Median Filtering Forensics Using an Autoregressive Model". *IEEE Transactions on Information Forensics and Security*. 8(9): 1456–1468.

Karras, T., T. Aila, S. Laine, and J. Lehtinen. (2017). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". *CoRR*. abs/1710.10196. arXiv: 1710.10196. URL: http://arxiv.org/abs/1710.10196.

Karras, T., S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. (2019). "Analyzing and Improving the Image Quality of StyleGAN". arXiv: 1912.04958 [cs.CV].

Khodabakhsh, A., R. Ramachandra, K. Raja, P. Wasnik, and C. Busch. (2018). "Fake Face Detection Methods: Can They Be Generalized?" In: *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 1–6.

Kim, D., H.-U. Jang, S.-M. Mun, S. Choi, and H.-K. Lee. (2018). "Median Filtered Image Restoration and Anti-Forensics Using Adversarial Networks". *IEEE Signal Processing Letters*. 25(2): 278–282. DOI: 10.1109/LSP.2017.2782363.

Korshunov, P. and S. Marcel. (2019). "Vulnerability assessment and detection of Deepfake videos". In: *2019 International Conference on Biometrics (ICB)*. 1–6.

Korshunov, P. and S. Marcel. (2018). "DeepFakes: a New Threat to Face Recognition? Assessment and Detection". *CoRR*. abs/1812.08685. arXiv: 1812.08685. URL: http://arxiv.org/abs/1812.08685.

Korus, P. and J. Huang. (2016). "Evaluation of random field models in multi-modal unsupervised tampering localization". In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

Korus, P. and J. Huang. (2017). "Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization". *IEEE Transactions on Information Forensics and Security*. 12(4): 809–824.

Krizhevsky, A., I. Sutskever, and G. Hinton. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". *Neural Information Processing Systems*. 25(Jan.). DOI: 10.1145/3065386.

Kuzin, A., A. Fattakhov, I. Kibardin, V. I. Iglovikov, and R. Dautov. (2018). "Camera Model Identification Using Convolutional Neural Networks". In: *2018 IEEE International Conference on Big Data (Big Data)*. 3107–3110.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*. 86(11): 2278–2324. DOI: 10.1109/5.726791.

Li, C. (2010). "Source Camera Identification Using Enhanced Sensor Pattern Noise". *IEEE Transactions on Information Forensics and Security*. 5(2): 280–287.

Li, C. and Y. Li. (2012). "Color-Decoupled Photo Response Non-Uniformity for Digital Image Forensics". *IEEE Transactions on Circuits and Systems for Video Technology*. 22(2): 260–271.

Li, J., T. Shen, W. Zhang, H. Ren, D. Zeng, and T. Mei. (2019). "Zooming into Face Forensics: A Pixel-level Analysis".

Li, Y., M. Chang, and S. Lyu. (2018). "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking". In: *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–7. DOI: 10.1109/WIFS.2018.8630787.

Li, Y. and S. Lyu. (2018). "Exposing DeepFake Videos By Detecting Face Warping Artifacts". *CoRR*. abs/1811.00656. arXiv: 1811.00656. URL: http://arxiv.org/abs/1811.00656.

Li, Y., P. Sun, H. Qi, and S. Lyu. (2020). "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics". In: *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*. Seattle, WA, United States.

Lin, J. Y., R. Song, C.-H. Wu, T. Liu, H. Wang, and C.-C. J. Kuo. (2015). "MCL-V: A streaming video quality assessment database". *Journal of Visual Communication and Image Representation*. 30: 1–9. DOI: https://doi.org/10.1016/j.jvcir.2015.02.012.

Liu, Z., P. Luo, X. Wang, and X. Tang. (2014). "Deep Learning Face Attributes in the Wild". arXiv: 1411.7766 `[cs.CV]`.

Long, C., A. Basharat, and A. Hoogs. (2018). "A Coarse-to-fine Deep Convolutional Neural Network Framework for Frame Duplication Detection and Localization in Video Forgery". *CoRR*. abs/1811.10762. arXiv: 1811.10762. URL: http://arxiv.org/abs/1811.10762.

Long, C., E. Smith, A. Basharat, and A. Hoogs. (2017). "A C3D-Based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot". DOI: 10.1109/CVPRW.2017.237.

Long, J., E. Shelhamer, and T. Darrell. (2014). "Fully Convolutional Networks for Semantic Segmentation". *CoRR*. abs/1411.4038. arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

Lukas, J., J. Fridrich, and M. Goljan. (2006). "Digital camera identification from sensor pattern noise". *IEEE Transactions on Information Forensics and Security*. 1(2): 205–214.

Mandelli, S., N. Bonettini, P. Bestagini, and S. Tubaro. (2020a). "Training CNNs in Presence of JPEG Compression: Multimedia Forensics vs Computer Vision". arXiv: 2009.12088 [cs.CV].

Mandelli, S., D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro. (2020b). "CNN-based fast source device identification". arXiv: 2001.11847 [cs.NE].

Maras, M.-H. and A. Alexandrou. (2018). "Determining Authenticity of Video Evidence in the Age of Artificial Intelligence and in the Wake of Deepfake Videos". *International Journal of Evidence and Proof*. 23(Oct.). DOI: 10.1177/1365712718807226.

Marra, F., D. Gragnaniello, D. Cozzolino, and L. Verdoliva. (2018a). "Detection of GAN-Generated Fake Images over Social Networks". In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. 384–389. DOI: 10.1109/MIPR.2018.00084.

Marra, F., C. Saltori, G. Boato, and L. Verdoliva. (2019a). "Incremental learning for the detection and classification of GAN-generated images". In: *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

Marra, F., D. Gragnaniello, L. Verdoliva, and G. Poggi. (2018b). "Do GANs leave artificial fingerprints?" arXiv: 1812.11842 [cs.CV].

Marra, F., D. Gragnaniello, L. Verdoliva, and G. Poggi. (2019b). "A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection". arXiv: 1909.06751 [cs.CV].

Matern, F., C. Riess, and M. Stamminger. (2019). "Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations". In: *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. 83–92. DOI: 10.1109/WACVW.2019.00020.

Mayer, O., B. Hosler, and M. C. Stamm. (2020). "Open Set Video Camera Model Verification". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2962–2966. DOI: 10.1109/ICASSP40776.2020.9054261.

Mayer, O. and M. C. Stamm. (2018). "Learned Forensic Source Similarity for Unknown Camera Models". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012–2016.

Mayer, O. and M. C. Stamm. (2019). "Exposing Fake Images with Forensic Similarity Graphs". arXiv: 1912.02861 [eess.IV].

Mayer, O. and M. C. Stamm. (2020). "Forensic Similarity for Digital Images". *IEEE Transactions on Information Forensics and Security*. 15: 1331–1346. DOI: 10.1109/tifs.2019.2924552.

Mazaheri, G., N. Chowdhury Mithun, J. H. Bappy, and A. K. Roy-Chowdhury. (2019). "A Skip Connection Architecture for Localization of Image Manipulations". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Montgomery, C. *et al.* (1994). "Xiph. org video test media (derf's collection), the xiph open source community". URL: https://media.xiph.org/video/derf/.

Nam, S., J. Park, D. Kim, I. Yu, T. Kim, and H. Lee. (2019). "Two-Stream Network for Detecting Double Compression of H.264 Videos". In: *2019 IEEE International Conference on Image Processing (ICIP)*. 111–115.

Nataraj, L., T. M. Mohammed, B. S. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury. (2019). "Detecting GAN generated Fake Images using Co-occurrence Matrices". *CoRR*. abs/1903.06836. arXiv: 1903.06836. URL: http://arxiv.org/abs/1903.06836.

Ng, T.-T. and S. Chang. (2004). "A data set of authentic and spliced image blocks".

Nguyen, H. H., F. Fang, J. Yamagishi, and I. Echizen. (2019a). "Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos". *CoRR*. abs/1906.06876. arXiv: 1906.06876. URL: http://arxiv.org/abs/1906.06876.

Nguyen, T. T., C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi. (2019b). "Deep Learning for Deepfakes Creation and Detection". arXiv: 1909.11573 [cs.CV].

Nick Dufour, A. G. (2019). "Contributing Data to Deepfake Detection Research". URL: https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

Niu, Y., B. Tondi, Y. Zhao, R. Ni, and M. Barni. (2021). "Image Splicing Detection, Localization and Attribution via JPEG Primary Quantization Matrix Estimation and Clustering". arXiv: 2102.01439 [eess.IV].

Noh, H., S. Hong, and B. Han. (2015). "Learning Deconvolution Network for Semantic Segmentation". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). ICCV '15*. USA: IEEE Computer Society. 1520–1528. DOI: 10.1109/ICCV.2015.178.

Papadopoulou, O., M. Zampoglou, S. Papadopoulos, and I. Kompatsiaris. (2018). "A corpus of debunked and verified user-generated videos". *Online Information Review*. DOI: 10.1108/OIR-03-2018-0101.

Park, J., D. Cho, W. Ahn, and H.-K. Lee. (2018). "Double JPEG Detection in Mixed JPEG Quality Factors using Deep Convolutional Neural Network". In: *The European Conference on Computer Vision (ECCV)*.

Parkhi, O. M., A. Vedaldi, and A. Zisserman. (2015). "Deep Face Recognition". In: *British Machine Vision Conference*.

Pengpeng, Y., R. Ni, and Y. Zhao. (2017). "Recapture Image Forensics Based on Laplacian Convolutional Neural Networks". DOI: 10.1007/978-3-319-53465-7-9.

Pevný, T., P. Bas, and J. Fridrich. (2010). "Steganalysis by Subtractive Pixel Adjacency Matrix". *IEEE Transactions on Information Forensics and Security*. 5(July). DOI: 10.1145/1597817.1597831.

Phan, Q., G. Boato, R. Caldelli, and I. Amerini. (2019). "Tracking Multiple Image Sharing on Social Networks". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 8266–8270. DOI: 10.1109/ICASSP.2019.8683144.

Phan, Q.-T., C. Pasquini, G. Boato, and F. Natale. (2018). "Identifying Image Provenance: An Analysis of Mobile Instant Messaging Apps". DOI: 10.1109/MMSP.2018.8547050.

Powers, D. (2008). "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation".

Qi Cui Suzanne McIntosh, H. S. (2018). "Identifying Materials of Photographic Images and Photorealistic Computer Generated Graphics Based on Deep CNNs". *Computers, Materials & Continua*. 55(2): 229–241. DOI: 10.3970/cmc.2018.01693.

Qian, Y., J. Dong, W. Wang, and T. Tan. (2015). "Deep learning for steganalysis via convolutional neural networks". In: *Electronic Imaging*.

Quan, Y., X. Lin, and C.-T. Li. (2019). "Provenance Analysis for Instagram Photos". In: *Data Mining*. Singapore: Springer Singapore. 372–383.

Radford, A., L. Metz, and S. Chintala. (2015). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". arXiv: 1511.06434 [cs.LG].

Rafi, A. M., U. Kamal, R. Hoque, A. Abrar, S. Das, R. Laganière, and M. K. Hasan. (2018). "Application of DenseNet in Camera Model Identification and Post-processing Detection". arXiv: 1809.00576 [eess.IV].

Rahmouni, N., V. Nozick, J. Yamagishi, and I. Echizen. (2017). "Distinguishing computer graphics from natural images using convolution neural networks". In: *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. 1–6.

Rebuffi, S., A. Kolesnikov, and C. H. Lampert. (2016). "iCaRL: Incremental Classifier and Representation Learning". *CoRR*. abs/1611.07725. arXiv: 1611.07725. URL: http://arxiv.org/abs/1611.07725.

Ren, S., K. He, R. B. Girshick, and J. Sun. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". *CoRR*. abs/1506.01497. arXiv: 1506.01497. URL: http://arxiv.org/abs/1506.01497.

Richard Fletcher Nic Newman, A. S. (2020). "A Mile Wide, an Inch Deep: Online News and Media Use in the 2019 UK General Election". URL: https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2020-02/Fletcher_News_Use_During_the_Election_FINAL.pdf.

Ronneberger, O., P. Fischer, and T. Brox. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". *CoRR*. abs/1505.04597. arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

Rössler, A., D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. (2018a). "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces". *CoRR*. abs/1803.09179. arXiv: 1803.09179. URL: http://arxiv.org/abs/1803.09179.

Rössler, A., D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. (2018b). "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces". *CoRR*. abs/1803.09179. arXiv: 1803.09179. URL: http://arxiv.org/abs/1803.09179.

Rössler, A., D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. (2019). "FaceForensics++: Learning to Detect Manipulated Facial Images". arXiv: 1901.08971 [cs.CV].

Sabir, E., J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan. (2019). "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos". *CoRR*. abs/1905.00582. arXiv: 1905.00582. URL: http://arxiv.org/abs/1905.00582.

Salloum, R., Y. Ren, and C.-C. Jay Kuo. (2018). "Image Splicing Localization using a Multi-task Fully Convolutional Network (MFCN)". *Journal of Visual Communication and Image Representation*. 51(Feb.): 201–209. DOI: 10.1016/j.jvcir.2018.01.010.

Schmid, C. (2001). "Constructing models for content-based image retrieval". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 2. II–II. DOI: 10.1109/CVPR.2001.990922.

Shanker, S. (2021). *The Discrete Cosine Transform in Action*. URL: https://squidarth.com/rc/math/2018/06/24/fourier.html. (accessed: 05.05.2021).

Shullani, D., M. Fontani, M. Iuliani, O. Alshaya, and A. Piva. (2017). "VISION: a video and image dataset for source identification". *EURASIP Journal on Information Security*. 2017(Oct.): 15. DOI: 10.1186/s13635-017-0067-2.

Simonyan, K. and A. Zisserman. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". *arXiv 1409.1556*. Sept.

Society, I. S. P. (2014). "IEEE IFS-TC Image Forensics Challenge Dataset". URL: http://ifc.recod.ic.%20unicamp.br/fc.website/index.py.

Society, I. S. P. (2018). "Camera Model Identification". URL: https://www.kaggle.com/c/sp-society-camera-model-identification/.

Soomro, K., A. R. Zamir, and M. Shah. (2012). "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild". arXiv: 1212.0402 [cs.CV].

Springenberg, J. T., A. Dosovitskiy, T. Brox, and M. Riedmiller. (2015). "Striving for Simplicity: The All Convolutional Net". arXiv: 1412.6806 [cs.LG].

Staff, R. (2021). "Fact check: Donald Trump concession video not a confirmed deepfake". URL: https://www.reuters.com/article/uk-factcheck-trump-consession-video-deep-idUSKBN29G2NL.

Stamm, M. C., M. Wu, and K. J. R. Liu. (2013). "Information Forensics: An Overview of the First Decade". *IEEE Access*. 1: 167–200.

Sullivan, G. J., J.-R. Ohm, W.-J. Han, and T. Wiegand. (2012). "Overview of the High Efficiency Video Coding (HEVC) Standard". *IEEE Transactions on Circuits and Systems for Video Technology*. 22(12): 1649–1668. DOI: 10.1109/TCSVT.2012.2221191.

Sun, D., X. Yang, M. Liu, and J. Kautz. (2017). "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume". *CoRR*. abs/1709.02371. arXiv: 1709.02371. URL: http://arxiv.org/abs/1709.02371.

Szegedy, C., S. Ioffe, and V. Vanhoucke. (2016). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". *CoRR*. abs/1602.07261. arXiv: 1602.07261. URL: http://arxiv.org/abs/1602.07261.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. (2014). "Going Deeper with Convolutions". *CoRR*. abs/1409.4842. arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. (2015). "Rethinking the Inception Architecture for Computer Vision". *CoRR*. abs/1512.00567. arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

Szeliski, R. (2011). *Computer vision algorithms and applications*. URL: http://dx.doi.org/10.1007/978-1-84882-935-0.

Tan, M. and Q. V. Le. (2020). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". arXiv: 1905.11946 `[cs.LG]`.

Tariq, S., S. Lee, H. Kim, Y. Shin, and S. S. Woo. (2018). "Detecting Both Machine and Human Created Fake Face Images In the Wild". In: *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*. *MPS '18*. Toronto, Canada: Association for Computing Machinery. 81–87. DOI: 10.1145/3267357.3267367.

Thies, J., M. Zollhöfer, and M. Nießner. (2019). "Deferred Neural Rendering: Image Synthesis using Neural Textures". *CoRR*. abs/1904.12356. arXiv: 1904.12356. URL: http://arxiv.org/abs/1904.12356.

Thies, J., M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. (2020). "Face2Face: Real-time Face Capture and Reenactment of RGB Videos". arXiv: 2007.14808 `[cs.CV]`.

Tokuda, E., H. Pedrini, and A. Rocha. (2013). "Computer generated images vs. digital photographs: A synergetic feature and classifier combination approach". *Journal of Visual Communication and Image Representation*. 24(8): 1276–1292. DOI: https://doi.org/10.1016/j.jvcir.2013.08.009.

Tralic, D., I. Zupancic, S. Grgic, and M. Grgic. (2013). "CoMoFoD — New database for copy-move forgery detection". In: *Proceedings ELMAR-2013*. 49–54.

Tran, D., L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. (2014). "C3D: Generic Features for Video Analysis". *CoRR*. abs/1412.0767. arXiv: 1412.0767. URL: http://arxiv.org/abs/1412.0767.

Trump: Deepfakes Replacement, Youtube video. (2018). URL: https://www.youtube.com/watch?v=hoc2RISoLWU&feature=emb_title.

Tuama, A., F. Comby, and M. Chaumont. (2016). "Camera model identification with the use of deep convolutional neural networks". In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

Verde, S., L. Bondi, P. Bestagini, S. Milani, G. Calvagno, and S. Tubaro. (2018). "Video Codec Forensics Based on Convolutional Neural Networks". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 530–534.

Verdoliva, L., D. Cozzolino, and G. Poggi. (2014). "A feature-based approach for image tampering detection and localization". In: *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. 149–154.

Verdoliva, L. (2020). "Media Forensics and DeepFakes: an overview". arXiv: 2001.06564 [cs.CV].

Vetterli, M., J. Kovacevic, and V. Goyal. (2018). "Foundations of Signal Processing". May: xxv–xxviii. DOI: 10.1017/cbo9781139839099.001.

Wang, J., Y. song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. (2014). "Learning Fine-grained Image Similarity with Deep Ranking". arXiv: 1404.4661 [cs.CV].

Wang, Q. and R. Zhang. (2016). "Double JPEG compression forensics based on a convolutional neural network". *EURASIP Journal on Information Security*. 2016(Dec.). DOI: 10.1186/s13635-016-0047-y.

Wang, S., O. Wang, A. Owens, R. Zhang, and A. A. Efros. (2019a). "Detecting Photoshopped Faces by Scripting Photoshop". *CoRR*. abs/1906.05856. arXiv: 1906.05856. URL: http://arxiv.org/abs/1906.05856.

Wang, S.-Y., O. Wang, R. Zhang, A. Owens, and A. A. Efros. (2019b). "CNN-generated images are surprisingly easy to spot... for now". arXiv: 1912.11035 [cs.CV].

Wen, B., Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler. (2016). "COVERAGE – A NOVEL DATABASE FOR COPY-MOVE FORGERY DETECTION". In: *IEEE International Conference on Image processing (ICIP)*. 161–165.

Wojna, Z., V. Ferrari, S. Guadarrama, N. Silberman, L. Chen, A. Fathi, and J. R. R. Uijlings. (2017). "The Devil is in the Decoder". *CoRR*. abs/1707.05847. arXiv: 1707.05847. URL: http://arxiv.org/abs/1707.05847.

Wu, Y., W. AbdAlmageed, and P. Natarajan. (2019). "ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9535–9544.

Wu, Y., W. Abd-Almageed, and P. Natarajan. (2018). "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization". In: *Computer Vision – ECCV 2018*. Cham: Springer International Publishing. 170–186.

Xia, Y., X. Cao, F. Wen, G. Hua, and J. Sun. (2015). "Learning Discriminative Reconstructions for Unsupervised Outlier Removal". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 1511–1519.

Xu, G., H. Wu, and Y. Shi. (2016). "Structural Design of Convolutional Neural Networks for Steganalysis". *IEEE Signal Processing Letters*. 23(5): 708–712.

Xuan, X., B. Peng, J. Dong, and W. Wang. (2019). "On the generalization of GAN image forensics". *CoRR*. abs/1902.11153. arXiv: 1902.11153. URL: http://arxiv.org/abs/1902.11153.

Yang, P., D. Baracchi, M. Iuliani, D. Shullani, R. Ni, Y. Zhao, and A. Piva. (2020a). "Efficient Video Integrity Analysis Through Container Characterization". *IEEE Journal of Selected Topics in Signal Processing*. 14(5): 947–954. DOI: 10.1109/JSTSP.2020.3008088.

Yang, P., D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva. (2020b). "A Survey of Deep Learning-Based Source Image Forensics". *Journal of Imaging*. 6(3): 9. DOI: 10.3390/jimaging6030009.

Yang, P., W. Zhao, R. Ni, and Y. Zhao. (2017). "Source Camera Identification Based On Content-Adaptive Fusion Network". *CoRR*. abs/1703.04856. arXiv: 1703.04856. URL: http://arxiv.org/abs/1703.04856.

Yang, X., Y. Li, and S. Lyu. (2019a). "Exposing Deep Fakes Using Inconsistent Head Poses". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 8261–8265. DOI: 10.1109/ICASSP.2019.8683164.

Yang, X., Y. Li, H. Qi, and S. Lyu. (2019b). "Exposing GAN-synthesized Faces Using Landmark Locations". *CoRR*. abs/1904.00167. arXiv: 1904.00167. URL: http://arxiv.org/abs/1904.00167.

Yao, Y., W. Hu, W. Zhang, T. Wu, and Y.-Q. Shi. (2018). "Distinguishing Computer-Generated Graphics from Natural Images Based on Sensor Pattern Noise and Deep Learning". *Sensors*. 18(4): 1296. DOI: 10.3390/s18041296.

Yao, Y., Y. Shi, S. Weng, and B. Guan. (2017). "Deep Learning for Detection of Object-Based Forgery in Advanced Video". *Symmetry*. 10(Dec.): 3. DOI: 10.3390/sym10010003.

Yarlagadda, S. K., D. Güera, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp. (2018). "Satellite Image Forgery Detection and Localization Using GAN and One-Class Classifier". *CoRR*. abs/1802.04881. arXiv: 1802.04881. URL: http://arxiv.org/abs/1802.04881.

You Won't Believe What Obama Says in This Video!, Youtube video. (2018). URL: https://www.youtube.com/watch?v=cQ54GDm1eL0.

Yu, F. and V. Koltun. (2015). "Multi-Scale Context Aggregation by Dilated Convolutions". arXiv: 1511.07122 [cs.CV].

Yu, F., V. Koltun, and T. A. Funkhouser. (2017a). "Dilated Residual Networks". *CoRR*. abs/1705.09914. arXiv: 1705.09914. URL: http://arxiv.org/abs/1705.09914.

Yu, I., D. Kim, J. Park, J. Hou, S. Choi, and H. Lee. (2017b). "Identifying photorealistic computer graphics using convolutional neural networks". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 4093–4097.

Yu, N., L. Davis, and M. Fritz. (2018). "Attributing Fake Images to GANs: Analyzing Fingerprints in Generated Images". *CoRR*. abs/1811.08180. arXiv: 1811.08180. URL: http://arxiv.org/abs/1811.08180.

Zagoruyko, S. and N. Komodakis. (2015). "Learning to Compare Image Patches via Convolutional Neural Networks". *CoRR*. abs/1504.03641. arXiv: 1504.03641. URL: http://arxiv.org/abs/1504.03641.

Zhang, A., Z. C. Lipton, M. Li, and A. J. Smola. (2020). *Dive into Deep Learning*.

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. (2015). "Learning Deep Features for Discriminative Localization". arXiv: 1512.04150 [cs.CV].

Zhou, P., X. Han, V. I. Morariu, and L. S. Davis. (2018a). "Learning Rich Features for Image Manipulation Detection". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1053–1061. DOI: 10.1109/CVPR.2018.00116.

Zhou, P., X. Han, V. I. Morariu, and L. S. Davis. (2018b). "Two-Stream Neural Networks for Tampered Face Detection". *CoRR*. abs/1803.11276. arXiv: 1803.11276. URL: http://arxiv.org/abs/1803.11276.

Zhu, J.-Y., T. Park, P. Isola, and A. A. Efros. (2017). "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". arXiv: 1703.10593 [cs.CV].