

# A Dataset of Photos and Videos for Digital Forensics Analysis Using Machine Learning Processing

Sara Ferreira <sup>1,\*</sup> , Mário Antunes <sup>2,3,\*</sup>  and Manuel E. Correia <sup>1,3</sup> 

<sup>1</sup> Department of Computer Science, Faculty of Sciences, University of Porto, 4169-007 Porto, Portugal; mdcorreia@fc.up.pt

<sup>2</sup> Computer Science and Communication Research Centre (CIIC), School of Technology and Management, Polytechnic of Leiria, 2411-901 Leiria, Portugal

<sup>3</sup> INESC TEC, CRACS, 4200-465 Porto, Portugal

\* Correspondence: sara.ferreira@fc.up.pt (S.F.); mario.antunes@ipleiria.pt (M.A.)

**Abstract:** Deepfake and manipulated digital photos and videos are being increasingly used in a myriad of cybercrimes. Ransomware, the dissemination of fake news, and digital kidnapping-related crimes are the most recurrent, in which tampered multimedia content has been the primordial disseminating vehicle. Digital forensic analysis tools are being widely used by criminal investigations to automate the identification of digital evidence in seized electronic equipment. The number of files to be processed and the complexity of the crimes under analysis have highlighted the need to employ efficient digital forensics techniques grounded on state-of-the-art technologies. Machine Learning (ML) researchers have been challenged to apply techniques and methods to improve the automatic detection of manipulated multimedia content. However, the implementation of such methods have not yet been massively incorporated into digital forensic tools, mostly due to the lack of realistic and well-structured datasets of photos and videos. The diversity and richness of the datasets are crucial to benchmark the ML models and to evaluate their appropriateness to be applied in real-world digital forensics applications. An example is the development of third-party modules for the widely used Autopsy digital forensic application. This paper presents a dataset obtained by extracting a set of simple features from genuine and manipulated photos and videos, which are part of state-of-the-art existing datasets. The resulting dataset is balanced, and each entry comprises a label and a vector of numeric values corresponding to the features extracted through a Discrete Fourier Transform (DFT). The dataset is available in a GitHub repository, and the total amount of photos and video frames is 40,588 and 12,400, respectively. The dataset was validated and benchmarked with deep learning Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) methods; however, a plethora of other existing ones can be applied. Generically, the results show a better F1-score for CNN when comparing with SVM, both for photos and videos processing. CNN achieved an F1-score of 0.9968 and 0.8415 for photos and videos, respectively. Regarding SVM, the results obtained with 5-fold cross-validation are 0.9953 and 0.7955, respectively, for photos and videos processing. A set of methods written in Python is available for the researchers, namely to preprocess and extract the features from the original photos and videos files and to build the training and testing sets. Additional methods are also available to convert the original PKL files into CSV and TXT, which gives more flexibility for the ML researchers to use the dataset on existing ML frameworks and tools.



**Citation:** Ferreira, S.; Antunes, M.; Correia, M.E. A Dataset of Photos and Videos for Digital Forensics Analysis Using Machine Learning Processing. *Data* **2021**, *6*, 87. <https://doi.org/10.3390/data6080087>

Academic Editor: Joaquín Torres-Sospedra

Received: 7 July 2021

Accepted: 3 August 2021

Published: 5 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Dataset:** <https://github.com/saraferreirascf/Photos-Videos-Manipulations-Dataset>

**Dataset License:** MIT License

**Keywords:** digital forensics; machine learning; photos and videos manipulation; Discrete Fourier Transform; tampered multimedia; deepfake

## 1. Background and Summary

The manipulation of multimedia content is increasingly appealing, mostly due to its direct influence in the spreading of fake news, defacing, deepfake, and digital kidnap cybercrime activities. The techniques used to manipulate digital photos and videos have been improved considerably and are mostly automated and supported by artificial intelligence methods. The resulting manipulated multimedia content is becoming harder to recognize.

The widespread techniques used to manipulate multimedia files can be broadly classified into the following main types: copy-move, splicing, deepfake, and resampling. Copy-move consists of rearranging the components of a photo by copying or moving them to different places on the same photo. The overall idea is to deceive the observer by giving the illusion of having more elements on the photo than those originally present. Splicing consists of overlapping different regions of two or more different photos into a new one. Resampling consists of changing the scale or even the position of an element in a photo. This type of manipulation can be used to recover old photos or even improve the visibility of photos in general. Figure 1 depicts an example of copy-move, while Figure 2 illustrates the use of splicing.

Deepfake photos and videos have been improved in recent years and have leveraged powerful ML techniques to improve the manipulation of the contents. Deep learning, more specifically, the training of generative neural networks such as auto-encoders or Generative Adversarial Networks (GANs) [1], is the most common ML method used to improve deepfake.

The detection of manipulated multimedia content has gained enthusiasts, especially in the digital forensics context, as the most recurrent today's crimes resort to tampered photos and videos. The Difference of Gaussians (DoG) and Oriented Rotated Brief (ORB) are techniques used to detect copy-move in manipulated photos [2]. DoG applies corners detection with the Sobel algorithm, features extraction with DoG and ORB, and features correspondence. These methods combine detection techniques based on blocks and key points in a single model. A match is found between two points of interest if the distance is less than a predetermined threshold.



(a) Original image.

(b) Manipulated image.

**Figure 1.** Copy-move manipulation.



Figure 2. Splicing manipulation.

Deepfake is the most known type of splicing, in which a person's face in a photo or video is swiped by another person's face [3]. A wide set of cybercrime activities is usually associated with this manipulation technique, being digital kidnapping in its various shapes the most common and those which may cause more damages to the victims. Figure 3 depicts an example of deepfake, where it is possible to observe that a new face was attached to the original torso.



Figure 3. Example of deepfake manipulation extracted from a video of Celeb-DF dataset [4].

Extracting features from photos with the Discrete Fourier Transform (DFT) method is described in [3]. It is based on a classical frequency domain analysis with DFT, in which the frequency characteristics of a photo is analyzed in a space defined by a Fourier transform, namely by applying a spectral decomposition of the input data, which corresponds to the way a signal's energy is distributed over a range of frequencies. DFT is a mathematical technique to decompose a discrete signal into a set of sinusoidal components of various frequencies ranging from 0 (constant frequency, corresponding to the image mean value) up to the maximum of the admissible frequency, given by the spatial resolution [5,6]. The frequency-domain representation of a signal, namely its amplitude and phase at each frequency, is calculated by Equation (1):

$$X_{k,l} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{n,m} \cdot e^{(-\frac{i2\pi}{N} k_n)} \cdot e^{(-\frac{-i2\pi}{M} l_m)} \quad (1)$$

Convolutional Neural Networks (CNN), also known as ConvNet, is a deep learning algorithm comprised of neurons that self-optimize through learning. Each neuron receives an input and performs an operation, such as a scalar product, followed by a non-linear function [7]. Technically, in CNN, each input photo will pass through a series of layers, in order to train and test the model. There are three types of layers: convolutional layers, pooling layers, and fully connected layers. CNN processing takes an input photo, processes it, and classifies it under certain pre-defined categories, such as fake or genuine. The input photo is seen as an array of pixels, and it depends on the photo resolution.

Jafar et al. [8] applied a CNN-based method to detect deepfake by using DFT in previously extracted mouth features (DFT-MF). Deepfake videos extraction is made by `moviePy` tool and takes into account the occurrences of certain words. By using the identified face landmarks, the frames in which the person has their mouth closed are removed.

Several surveys on the use of deep learning methods for digital forensics have been published recently [9,10]. The results obtained with CNN on image forensics are impressive and outperform those obtained with other machine learning methods. However, the processing time and the computational resources allocated are far beyond the admissible for standalone digital forensic stations [5].

Support Vector Machines (SVM) is an ML kernel-based method and has been successfully used in a wide set of classification problems, namely those applied to a binary classification between two distinct classes. It has been employed on manipulated photos and videos detection with promising results and reduced processing times [3].

ML methods are being incorporated into real-world digital forensics applications, as standalone applications or as third-party modules in widely used tools, such as Autopsy. When properly automated, ML classification and detection tasks can have a great impact on the daily routine of criminal investigation, namely on cybercrimes involving the detection of tampered photos and videos. However, realistic datasets should be made available to benchmark and challenge ML methods to detect tampered multimedia content.

The aim of this paper is to describe a compound dataset of photos and videos built on top of already published state-of-the-art datasets. It is a realistic and up-to-date dataset composed of about 52,000 examples of genuine and manipulated photos and videos, which incorporates the most common manipulation techniques. The dataset is available at a GitHub repository under an MIT license, and the researchers have at their disposal a set of scripts written in Python to preprocess, extract the features from the original multimedia files, and process the dataset files with ML methods through already existing frameworks. The dataset was evaluated with SVM by extracting 50 simple features with DFT, and with a CNN-based method, by applying a set of scripts that are also available for that purpose.

The remaining of the paper is organized as follows. Section 2 describes the data that is contained in the dataset, its format, and how it can be read and interpreted. Section 3 details the methods developed to preprocess and process the dataset, as well as how the data can be reused. Section 4 describes the technical validation of the dataset, namely by using SVM and CNN-based methods.

## 2. Data Description

The dataset presented in this paper is a compilation of genuine and manipulated photos and videos already published and available in state-of-the-art datasets. These datasets have been used to benchmark ML methods for classification and manipulation detection purposes. Table 1 summarizes the original datasets that were gathered in the resulting dataset and are described in this Section. The proposed dataset incorporates both objects and people's faces, being possible to detect distinct types of manipulations aside deepfake.

**Table 1.** Composition of the dataset.

Name	Fake	Real	Content	Manipulation Type	Source
CelebA-HQ dataset	-	10,000	photos	-	[3,11]
Flickr-Faces-HQ dataset	-	10,000	photos	-	[3,12]
“100K Facesproject”	10,000	-	photos	Deepfake	[13]
“this person does not exist”	10,000	-	photos	Deepfake	[14]
COVERAGE dataset	97	97	photos	Copy-move	[15]
Columbia Image Splicing Dataset	180	183	photos	Splicing	[16]
Created by us	14	14	photos	Copy-move	[17]
Celeb-DFv1	795	158	videos	Deepfake	[4]
	21,086	20,452			

Several works have already processed a compound dataset [3], namely by compiling photos available in CelebA-HQ dataset [11], Flickr-Faces-HQ dataset [12], “100K Faces project” (<https://generated.photos>, accessed on 4 August 2021) and “this person does not exist” project (<https://thispersondoesnotexist.com>, accessed on 4 August 2021). The datasets described on Table 1 were tested and benchmarked individually in a wide set of published research works [2–4,8]. Notwithstanding the richness of the published datasets, some of them only have deepfake-based manipulations examples. To overcome this limitation, datasets with distinct manipulations types, such as copy-move, were added. To do this, additional datasets that contain not only faces but also everyday objects were added. COVERAGE dataset [15] is a copy-move forgery database with similar but genuine objects that contains 97 legitimate photos and 97 manipulated ones.

Columbia Uncompressed Image Splicing Detection Evaluation Dataset [16] was also added, which consists of high-resolution images, 183 authentic (taken using just one camera and not manipulated), and 180 spliced photos. Additional 14 legitimate and 14 fake ad hoc photos were also added, containing splicing and copy-move manipulations. In [18], the authors proposed a technique that utilizes a fully convolutional network (FCN) to localize image splicing attacks training with the Columbia dataset.

Celeb-DF [4] has 795 fake and 158 real videos, extracted from Youtube. To combine these videos with the rest of the dataset, three frames per second were extracted from each video, in a total of 6200 extracted frames from real videos, and 31,551 from fake ones. In [19], the authors proposed a method to edit physiological signals in facial videos, and the experiments were conducted using the Celeb-DF dataset.

The final dataset is balanced, as more machine learning models could be used to train and test the models. To achieve that, if at some point there are more real photos than fake ones, only a minimum amount between them is used. To be more specific, as there are 31,551 fake frames extracted from videos and 6200 real ones, there will only be used 6200 photos from the fake ones, totaling 12,400 photos extracted from videos.

Therefore, the compound dataset proposed in this paper has a similar number of examples for both fake and genuine photos and videos. It is composed of 52,988 examples, which corresponds to 40,588 photos and 12,400 videos, as detailed in Table 2.

**Table 2.** Number of examples available on the compound dataset.

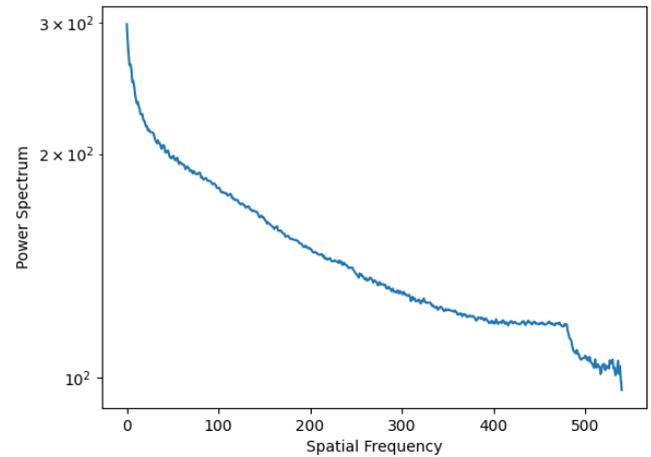
	Number of Examples
Photos	40,588
Videos	12,400
Total	52,988

Figure 4 illustrates a manipulated photo (Figure 4a) and its original version (Figure 4c), as well as the corresponding power spectrum obtained by DFT method described on

Section 1 (Figure 4b,d). It is possible to identify the variations of the power spectrum between both photos.



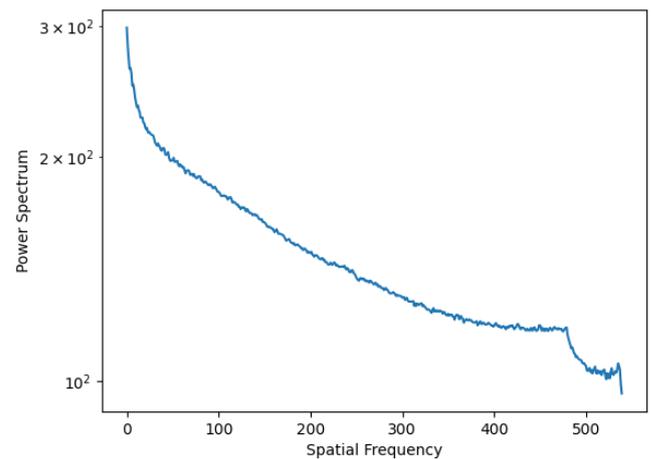
(a) Manipulated photo submitted for testing.



(b) DFT power spectrum.



(c) Real photo submitted for testing.



(d) DFT power spectrum.

**Figure 4.** Photo features extraction by using DFT [4].

Figure 5 illustrates an entry of the resulting uni-dimensional vector. Each entry starts with the classification label, followed by a list of numeric values, which corresponds to the features extracted from the photo through DFT. The label can have two values, namely 0 and 1, which correspond to a genuine or manipulated photo, respectively. The following values are obtained applying an azimuthal averaging to compute a robust one-dimensional representation of the DFT power spectrum. It can be seen as a compression, gathering and averaging similar frequency components into a vector of features.

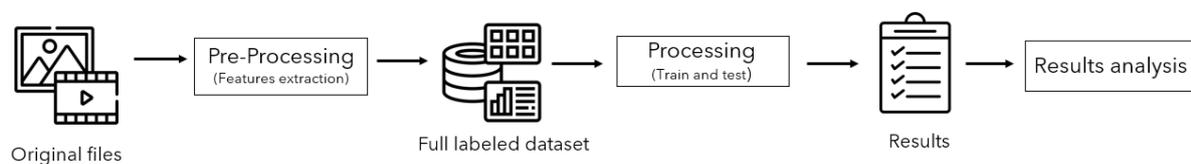
The dataset was validated (Section 4) with 50 simple features extracted through the DFT method. Different features sets can be extracted, being their length intrinsically related with CPU/memory configuration and processing time required. Researchers can preprocess the dataset with different features sets, as described in Section 3.

```
[1.          0.82355024 0.77355254 0.7250774  0.69703216 0.66774856
 0.65085273 0.62850308 0.61371226 0.59162878 0.58443304 0.56818412
 0.55030864 0.54080633 0.52237036 0.50521914 0.49109923 0.47546215
 0.46539512 0.45412343 0.44550405 0.43447022 0.42734298 0.41989387
 0.41687332 0.40845023 0.40379619 0.40252673 0.39726456 0.38867093
 0.38868427 0.38281923 0.37291213 0.36753934 0.36654633 0.36439966
 0.3594111  0.36067989 0.35511738 0.34150034 0.33475406 0.32893409
 0.332545   0.33369142 0.32661819 0.3237767  0.32802495 0.32731123
 0.32439296 0.33110984]
```

**Figure 5.** Photo features extraction by using DFT.

### 3. Methods

This Section describes the methods available to preprocess and process the dataset. The experimental setup pipeline is depicted in Figure 6.



**Figure 6.** The pipeline of the experimental setup of the dataset.

The overall architecture is comprised of three main phases, namely preprocessing and features extraction (Section 3.1), processing (Section 3.2) and results analysis (Section 3.3). Two complementary methods were developed to convert the datasets to other formats, namely CSV and TXT (Section 3.4).

The dataset files and the developed methods to preprocess and process the photos and videos are available on the following GitHub repository: <https://github.com/saraferreirascf/Photos-Videos-Manipulations-Dataset> (accessed on 4 August 2021). The software development and experiments were conducted on a PC with Windows 10, 8 GB RAM, and AMD Ryzen 52,600. The following software applications are required: Python version 3.9.2, Python module NumPy version 1.19.4, OpenCV version 4.4.0.46, Matplotlib version 3.3.3, SciPy version 1.5.4, and SciKit-learn version 0.23.2.

#### 3.1. Preprocessing and Features Extraction Phase

The preprocessing phase aims to transform the original photos and video frames into a labeled dataset. The files are converted into a uni-dimensional array, which is the result of the DFT simple features extraction. Regarding video files, three frames per second were extracted, which corresponds to an admissible and common value used in digital forensics. The features extraction and the setup of training and testing sets are implemented by the following corresponding scripts:

```
./create_train_file.py <dir> <features> <max_files> <output_filename>
./create_test_file.py <dir> <features> <max_files> <output_filename>
```

Where:

- <dir> corresponds to the directory containing the original dataset, which has the sub-directories *fake* and *real*, respectively, for tampered and genuine photos;
- <features> is the number of simple features to extract from each file by applying the DFT method;
- <max\_files> is the maximum number of files used for the classes *fake* and *real*;
- <output\_filename> is the output filename for the training or testing dataset.

The output of the scripts is a PKL file, which is created by the Python module named “pickle” (<https://docs.python.org/3/library/pickle.html>, accessed on 2 July 2021). The PKL

file contains a byte stream that represents the serialized objects, which can be deserialized back into the runtime Python program. Each PKL file record has a label and a numeric array composed of a set of simple features extracted by DFT.

### 3.2. Processing Phase

A set of ML methods can be used by the researchers to process and benchmark the proposed dataset. A Python script is available on GitHub (directory *Scripts*) to automate the dataset processing with an SVM-based method. The script is able to process an input file or split the dataset into a K-fold (5 or 10) or 67% for training and 33% for testing.

```
./svm_model.py <training_file> <testing_file> <run_mode>
```

Where:

- <training\_file> receives the training input file to train the SVM model;
- <testing\_file> receives the testing file, namely those that should be classified;
- <run\_mode> receives a numeric value with the mode to process the SVM model.

The parameter <run-mode> can have one of the following values:

- -1: classifies each entry in the <testing\_file>;
- 0: splits the dataset into two parts: 67% for training and 33% for testing;
- 5: splits the dataset to be used in a 5-fold cross validation;
- 10: splits the dataset to be used in a 10-fold cross validation;

The script `cnm_model.py` is also available to process the dataset with CNN. It uses tensorflow and keras and can be used as described below:

```
./cnm_model.py <training_folder> <testing_folder> <run_mode>
```

Where:

- <training\_folder> receives the folder containing files to train the CNN model. This folder must have two sub-directories: "fake" and "real";
- <testing\_folder> receives the folder containing the files to be classified. This folder needs to have one sub-directory named "predict";
- <run\_mode> can be one of the following two values: 0 to test with 10% of the files into the training folder; 1 to test with the files that are in the testing folder.

### 3.3. Results Analysis

The performance evaluation is made by calculating a set of classification metrics. The metrics used to evaluate the results obtained during the dataset validation (Section 4) were Precision (P), Recall (R), F1-score, and Accuracy (A). Table 3 depicts the confusion matrix [20], which inputs the calculations of the evaluation metrics summarized in Table 4.

**Table 3.** The confusion matrix.

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Precision measures the number of photos and videos predicted as manipulated that actually were manipulated. The recall is the percentage of manipulated examples that were predicted from the total number of manipulated examples. Accuracy measures the rate of correct classifications out of all the examples in the dataset. Finally, F1-score is a weighted average of Precision and Recall. It ranges between [0, 1] and measures the preciseness and robustness of the classifier.

**Table 4.** Metrics used to evaluate the dataset.

Metric	Equation
Precision	$P = \frac{TP}{(TP+FP)}$
Recall	$R = \frac{TP}{(TP+FN)}$
F1	$F1 = 2 * \frac{P*R}{(P+R)}$
Accuracy	$A = \frac{TP+TN}{TP+TN+FP+FN}$

The scripts developed to calculate the evaluation metrics takes advantage of the `sklearn` library for Python (<https://scikit-learn.org/stable/>, accessed on 23 June 2021). By observing the Listing 1, with the `y_test` from the testing set and `x_pred` from the predictions given by the SVM model, it is possible to get the evaluation metrics, namely precision, recall, F1-score, and the resulting confusion matrix.

Listing 1: Python code to calculate the evaluation metrics.

```

from sklearn.metrics import classification_report , confusion_matrix
print(classification_report(y_test , x_pred))

print("confusion_matrix")
print(confusion_matrix(y_test , x_pred))

print("True_Positives:" ,confusion_matrix(y_test , x_pred)[0][0])
print(" False_Negatives:" ,confusion_matrix(y_test , x_pred)[0][1])
print(" False_Positives:" ,confusion_matrix(y_test , x_pred)[1][0])
print(" True_Negatives:" ,confusion_matrix(y_test , x_pred)[1][1])

```

Listing 1 is an excerpt of the script used to calculate the performance obtained by processing the testing dataset against the SVM model learned with the training dataset.

### 3.4. Complementary Methods

A set of complementary functions was developed to give researchers the flexibility to process the files into different formats. The training and testing files are originally in PKL format. Two distinct scripts were made available to convert PKL to CSV and TXT files formats. The conversion of a PKL file to CSV format is obtained by executing the following script:

```
./pkl_to_csv.py <pkl_file> <csv_file>
```

Where:

- `<pkl_file>` receives a pkl file to be converted to a CSV format;
- `<csv_file>` is the output and corresponds to a file in the CSV format.

This script creates two different files in CSV format: one with the features and one with the corresponding labels.

Regarding TXT format, the conversion of a PKL input file is obtained through the execution of the following script:

```
./pkl_to_txt.py <pkl_file> <txt_file>
```

Where:

- `<pkl_file>` receives a pkl file to be converted to a txt format;
- `<txt_file>` is the output and corresponds to a txt file.

#### 4. Technical Validation

This section details the technical validation of the dataset. The dataset was initially processed by applying a Support Vector Machine (SVM)-based machine learning method. The results obtained were then benchmarked against a deep learning Convolutional Neural Network (CNN) based method.

It has been chosen to validate the dataset with exactly 50 features extracted from each multimedia file. However, we have made experimental tests with sets with different lengths and collected the corresponding processing time and F1-score. Table 5 summarizes the results obtained, where it is possible to observe that between 50 and 100 features, the processing time almost doubled, and the F1-score slightly increases. The dataset validation was made with 50 features; however, it can be processed with different features sets of different lengths.

**Table 5.** The influence of the number of features on the total processing time and F1-score.

Features	Preprocessing Time (s)	Processing Time (s)	F1-Score
20	1437	14.84	0.7010
50	1708	22.64	0.7160
100	3026	43.16	0.7570
200	4103	94.84	0.7819
500	5339	185.71	0.8462

The SVM processing and dataset evaluation were made through two distinct approaches: 5-fold cross-validation [5] and 10-fold cross-validation, which are detailed below in this Section. The dataset described in Section 2 was divided into three parts: one part with only photos, one part with only frames taken from videos, and a third part with the mixture of the other two parts. For each part, a 10-fold cross-validation was performed, and the results are shown in Tables 6–8.

For each split, corresponding to the evaluation of a  $K$  part of the dataset, the values of TP, TN, FP, and FN were obtained. With the values obtained, the precision, recall, F1-score, and accuracy were calculated using the formulas explained in Section 3.

**Table 6.** The results obtained with 10-fold cross-validation against the dataset containing only photos.

	TP	TN	FP	FN	Precision	Recall	F1-Score	Accuracy
Split 1	2016	2015	7	20	0.9965	0.9902	0.9933	0.9933
Split 2	1983	2056	12	7	0.9940	0.9965	0.9952	0.9953
Split 3	2057	1980	7	14	0.9966	0.9932	0.9949	0.9948
Split 4	2032	2005	11	10	0.9946	0.9951	0.9949	0.9948
Split 5	2012	2027	4	15	0.9980	0.9926	0.9953	0.9953
Split 6	2079	1954	9	16	0.9957	0.9924	0.9940	0.9938
Split 7	2004	2039	9	6	0.9955	0.9970	0.9963	0.9963
Split 8	1971	2070	4	13	0.9980	0.9934	0.9957	0.9958
Split 9	2026	2018	5	9	0.9975	0.9956	0.9966	0.9966
Split 10	1989	2052	6	11	0.9970	0.9945	0.9957	0.9958
<b>Mean</b>	<b>2017</b>	<b>2022</b>	<b>7</b>	<b>12</b>	<b>0.9963</b>	<b>0.9941</b>	<b>0.9952</b>	<b>0.9952</b>

Table 6 describes the results obtained with 10-fold cross-validation to the dataset containing only photos. The table highlights the partial results obtained in each split, namely the number of FP, FN, TN, and TP, as well as the calculated values for Precision, Recall, F1, and Accuracy. The corresponding mean scores obtained with the 10-fold cross-validation are also indicated. The results obtained show a mean F1-score above 99.52%, which outperforms the state-of-the-art documented work [2]. The mean value obtained for accuracy ( $A$ ) is 99.52%, which surpasses the result of 93.52% achieved in [2]. The number of incorrectly classified examples, namely FP and FN, is low, having a mean value of 7 and 12, respectively.

The results attained with 10-fold cross-validation against the dataset containing only videos are presented in Table 7. The mean values for F1-score and accuracy are 79.8% and 78.3%, respectively. When comparing with previously documented experiments [8], it is possible to note that, using the Celeb-DF dataset [4] as part of the input dataset, the results outperform those obtained with the DFT-MF approach, which achieved an accuracy of 71.25%. Regarding misclassified examples, the average values for FP and FN are, respectively, 180 and 89 in a total amount of 1240 examples.

**Table 7.** The results obtained with 10-fold cross-validation against the dataset containing only videos.

	TP	TN	FP	FN	Precision	Recall	F1-Score	Accuracy
Split 1	544	442	174	80	0.7577	0.8718	0.8107	0.7952
Split 2	553	447	135	105	0.8038	0.8404	0.8217	0.8065
Split 3	548	420	188	84	0.7446	0.8671	0.8012	0.7806
Split 4	510	441	198	91	0.7203	0.8486	0.7792	0.7669
Split 5	520	443	184	93	0.7386	0.8483	0.7897	0.7766
Split 6	554	448	159	79	0.7770	0.8752	0.8232	0.8081
Split 7	522	426	202	90	0.7210	0.8529	0.7814	0.7645
Split 8	505	464	177	94	0.7405	0.8431	0.7884	0.7815
Split 9	524	421	196	99	0.7278	0.8411	0.7803	0.7621
Split 10	532	453	182	73	0.7451	0.8793	0.8067	0.7944
<b>Mean</b>	<b>531</b>	<b>441</b>	<b>180</b>	<b>89</b>	<b>0.7476</b>	<b>0.8568</b>	<b>0.7983</b>	<b>0.7836</b>

**Table 8.** The results obtained with 10-fold cross-validation against the dataset containing both photos and videos.

	TP	TN	FP	FN	Precision	Recall	F1-Score	Accuracy
Split 1	2689	1962	641	7	0.8075	0.9974	0.8925	0.8777
Split 2	2689	2005	600	5	0.8176	0.9981	0.8989	0.8858
Split 3	2633	2040	623	3	0.8087	0.9989	0.8938	0.8819
Split 4	2627	2021	641	10	0.8039	0.9962	0.8898	0.8771
Split 5	2631	2012	651	5	0.8016	0.9981	0.8892	0.8762
Split 6	2656	2000	640	3	0.8058	0.9989	0.8920	0.8787
Split 7	2647	2015	630	7	0.8077	0.9974	0.8926	0.8798
Split 8	2596	2083	612	8	0.8092	0.9969	0.8933	0.8830
Split 9	2639	2023	632	5	0.8068	0.9981	0.8923	0.8798
Split 10	2627	2043	621	8	0.8088	0.9969	0.8931	0.8813
<b>Mean</b>	<b>2643</b>	<b>2020</b>	<b>629</b>	<b>6</b>	<b>0.8078</b>	<b>0.9978</b>	<b>0.8927</b>	<b>0.8801</b>

Considering that videos are composed of a set of frames, a third experiment was made to accommodate both multimedia content types. Table 8 presents the results obtained with the whole dataset composed of 52,990 examples, applying 10-fold cross-validation.

It is possible to observe that the mean values for precision, recall, and F1-score are, respectively, 80.78%, 99.78%, and 89.23%. The calculated mean accuracy is 88.01%, and the overall results outperform those attained and documented in [3].

Table 9 summarizes the dataset evaluation for photos processing made with different methods, while Table 10 summarizes the results obtained with the processing of video frames. The results obtained with 5-fold cross-validation and the CNN-based method are described in [5].

**Table 9.** Dataset evaluation for photos.

ML Method	Features Extraction	Precision	Recall	F1-Score	Accuracy
SVM - 5-fold CV	DFT	0.9965	0.9941	0.9953	0.9951
SVM - 10-fold CV	DFT	0.9963	0.9941	0.9952	0.9952
CNN	Original files	0.9970	0.9966	0.9968	0.9967

**Table 10.** Dataset evaluation for videos.

ML Method	Features Extraction	Precision	Recall	F1-Score	Accuracy
SVM - 5-fold CV	DFT	0.7438	0.8548	0.7955	0.7794
SVM - 10-fold CV	DFT	0.7476	0.8568	0.7983	0.7836
CNN	Original files	0.8820	0.8045	0.8415	0.8387

As depicted in Table 7, compared with Table 6, it is possible to note that videos has lower accuracy. These results can be justified with the number of frames extracted from each video. Since only 3–4 frames per second were extracted, the frames with manipulations may go unnoticed. The quality of the videos present in the dataset can also partially justify the results obtained.

Benchmarking ML methods is crucial to investigate innovative learning methods that could be successfully applied in the detection of tampered multimedia files in a digital forensics analysis context. By observing Table 11, it is possible to note the DFT-SVM-based method has quicker processing times comparing to the CNN-based method. As the aiming is usually implementing these ML methods in digital forensic tools to automate the process of detecting tampered multimedia content, time is a important factor. Even though the CNN-based method achieved better results, their preprocessing and processing times can be unbearable in real-time processing scenarios. Additional research should be made to reduce the processing time on using CNN in standalone digital forensics tools.

**Table 11.** Processing time spent for videos and photos, in the format hh:mm:ss.

	Photos	Videos
DFT-SVM-based method	00:00:51	00:02:00
CNN-based method	06:36:00	02:40:00

Deep learning based methods have been widely used and are considered state-of-the-art in image and video forensics [9,10]. Notwithstanding, the features extraction methods and the overall functioning of deep learning based models, such as CNN and RNN, are time-consuming to process and less flexible to be embedded into a standalone digital forensics application, such as Autopsy. Regarding the DFT-SVM-based method used to

process the proposed dataset, the results achieved are competitive with the CNN model for both photos and videos with a significantly lower processing time, as depicted in Table 11. The trade-off between the processing time and the evaluation performance obtained by DFT-SVM method [3] should thus be taken in account in the creation of forensic tools to support and help criminal investigator's digital forensics daily routine.

## 5. Conclusions

This paper described a dataset of genuine and manipulated photos and videos to be used by ML methods in the detection of tampered multimedia content. A classified dataset of about 40,000 photos is proposed, composed of both faces and objects, where it is possible to find examples of copy-move, splicing, and deepfake manipulations. Technical validation of the dataset was made by benchmarking it with CNN and SVM ML methods.

The DFT features extraction method was used to process the dataset with SVM. A set of 50 features was used for technical validation of the dataset, being however possible to extract a different number of features. Regarding CNN, the original multimedia files were processed. The results obtained are in line with those documented in the literature, namely on the use of SVM and CNN methods to detect tampered files. Generally, it was possible to achieve a mean F1-score of 99.68% on the detection of manipulated photos, while a mean F1-score of 84.15% was attained for videos.

The dataset is delivered with a set of tools that give flexibility to the researchers, namely by using it in different ML frameworks and with distinct formats. The use of realistic and well-structured datasets, such as the one presented in the paper, give the ML practitioners and researchers the ability to test a vast set of methods and models that can be further applied to solve digital forensics real-world problems. By incorporating these methods into well-known digital forensics tools, such as Autopsy ([www.autopsy.com](http://www.autopsy.com), accessed on 23 June 2021), the daily routine of criminal investigation could benefit enormously [5].

Future work has the following major topics: to continuously improve the dataset by integrating more genuine and manipulated photos, namely by enhancing the quality and resolution; to incorporate videos with high-quality manipulations that may challenge the ML methods even more.

**Author Contributions:** Conceptualization, S.F., M.A., and M.E.C.; data curation, S.F. and M.A.; formal analysis, S.F. and M.A.; funding acquisition, M.E.C.; investigation, S.F., M.A., M.E.C.; methodology, S.F. and M.A.; software, S.F.; supervision, M.A. and M.E.C.; validation, M.A. and M.E.C.; visualization, S.F.; writing—original draft: S.F., M.A., M.E.C.; writing—review and editing: S.F., M.A., and M.E.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is financed by National Funds through the Portuguese funding agency, FCT-Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data are publicly available under a MIT License in the following GitHub repository: <https://github.com/saraferreirascf/Photos-Videos-Manipulations-Dataset> (accessed on 4 August 2021).

**Acknowledgments:** The authors acknowledge the facilities provided by INESC TEC, Faculty of Sciences, and University of Porto, for the support to this research. This work is financed by National Funds through the Portuguese funding agency, FCT-Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

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

## Abbreviations

The following abbreviations are used in this manuscript:

A	Accuracy
CNN	Convolutional Neural Networks
CV	Cross Validation
DFT	Discrete Fourier Transformation
DoG	Difference of Gaussian
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
ML	Machine Learning
ORB	Oriented Rotated Brief
P	Precision
R	Recall
RNN	Recurrent Neural Networks
SVM	Support Vector Machines
TN	True Negative
TP	True Positive

## References

1. Nguyen, T.T.; Nguyen, C.M.; Nguyen, D.T.; Nguyen, D.T.; Nahavandi, S. Deep learning for deepfakes creation and detection. *arXiv* **2019**, arXiv:1909.11573.
2. Niyishaka, P.; Bhagvati, C. Digital image forensics technique for copy-move forgery detection using dog and orb. In Proceedings of the International Conference on Computer Vision and Graphics, Warsaw, Poland, 17–19 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 472–483.
3. Durall, R.; Keuper, M.; Pfrendt, F.J.; Keuper, J. Unmasking deepfakes with simple features. *arXiv* **2019**, arXiv:1911.00686.
4. Li, Y.; Yang, X.; Sun, P.; Qi, H.; Lyu, S. Celeb-df: A large-scale challenging dataset for deepfake forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 3207–3216.
5. Ferreira, S.; Antunes, M.; Correia, M.E. Exposing Manipulated Photos and Videos in Digital Forensics Analysis. *J. Imaging* **2021**, *7*, 102. [CrossRef]
6. Ferreira, S.; Antunes, M.; Correia, M.E. Forensic analysis of tampered digital photos. In Proceedings of the 25th Iberoamerican Congress on Pattern Recognition (CIARP), IARP, Porto, Portugal, 10–13 May 2021; pp. 402–411.
7. O’Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
8. Jafar, M.T.; Ababneh, M.; Al-Zoube, M.; Elhassan, A. Forensics and Analysis of Deepfake Videos. In Proceedings of the IEEE 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 53–58.
9. Castillo Camacho, I.; Wang, K. A Comprehensive Review of Deep-Learning-Based Methods for Image Forensics. *J. Imaging* **2021**, *7*, 69. [CrossRef]
10. Yang, P.; Baracchi, D.; Ni, R.; Zhao, Y.; Argenti, F.; Piva, A. A survey of deep learning-based source image forensics. *J. Imaging* **2020**, *6*, 9. [CrossRef]
11. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
12. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
13. 100k Faces Generated. Available online: <https://generated.photos> (accessed on 4 August 2021).
14. This Person Does Not Exist Website. Available online: <https://thispersondoesnotexist.com> (accessed on 4 August 2021).
15. Wen, B.; Zhu, Y.; Subramanian, R.; Ng, T.T.; Shen, X.; Winkler, S. COVERAGE—A novel database for copy-move forgery detection. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 161–165.
16. Hsu, Y.F.; Chang, S.F. Detecting image splicing using geometry invariants and camera characteristics consistency. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 549–552.
17. Photos-Videos-Manipulations-Dataset. Available online: <https://github.com/saraferreirascf/Photos-Videos-Manipulations-Dataset> (accessed on 4 August 2021).
18. Salloum, R.; Ren, Y.; Kuo, C.C.J. Image splicing localization using a multi-task fully convolutional network (MFCN). *J. Vis. Commun. Image Represent.* **2018**, *51*, 201–209. [CrossRef]

- 
19. Chen, M.; Liao, X.; Wu, M. PulseEdit: Editing Physiological Signal in Facial Videos for Privacy Protection. 2021. Available online: [https://www.techrxiv.org/articles/preprint/PulseEdit\\_Editing\\_Physiological\\_Signal\\_in\\_Facial\\_Videos\\_for\\_Privacy\\_Protection/14647377](https://www.techrxiv.org/articles/preprint/PulseEdit_Editing_Physiological_Signal_in_Facial_Videos_for_Privacy_Protection/14647377) (accessed on 4 August 2021).
  20. Hossin, M.; Sulaiman, M. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manag. Process.* **2015**, *5*, 1.