*APSIPA Transactions on Signal and Information Processing, 2022, 11, e43*

## Original Paper

# Detecting Deepfake Videos in Data Scarcity Conditions by Means of Video Coding Features

Jun Wang*, Omran Alamayreh, Benedetta Tondi, Andrea Costanzo and Mauro Barni

*Department of Information Engineering and Mathematics, University of Siena, Via Roma, 53100 Siena, Italy*

ABSTRACT

The most powerful deepfake detection methods developed so far are based on deep learning, requiring that large amounts of training data representative of the specific task are available to the trainer. In this paper, we propose a feature-based method for video deepfake detection that can work in data scarcity conditions, that is, when only very few examples are available to the forensic analyst. The proposed method is based on video coding analysis and relies on a simple footprint obtained from the motion prediction modes in the video sequence. The footprint is extracted from video sequences and used to train a simple linear Support Vector Machine classifier. The effectiveness of the proposed method is validated experimentally on three different datasets, namely, a synthetic street video dataset and two datasets of Deepfake face videos.

*Keywords:* Deepfake detection, video forensics, deep learning for forensics, video coding.

*Corresponding author: Jun Wang, j.wang@student.unisi.it.

Received 27 May 2022; Revised 05 August 2022
ISSN 2048-7703; DOI 10.1561/116.00000032

## 1 Introduction

Modern Artificial Intelligence (AI) techniques for synthetic media generation allow us to generate images and videos from scratch and manipulate them with an extremely high level of realism. Some techniques allow synthesizing a new image from scratch [15] or transferring the style from an image to another [32], while others implement object swapping, i.e., they swap objects in an image while keeping the background unchanged [19]. In this category, face-swapping is a common manipulation, where a person's face is swapped with the face of another person [21]. Due to the high realism of synthetic and manipulated media, it is getting harder and harder to distinguish altered media from pristine ones, even for humans. These contents, called deepfakes, can be maliciously used as a source of misinformation. For this reason, detecting AI-generated media is becoming a more and more pressing need, which attracted the attention of researchers in the last years [26].

The goal of video deepfake detection is to distinguish AI-manipulated videos from real ones. Powerful video deepfake detection methods developed so far are based on Deep Learning (DL) [24], and require that a large amount of training data representative of the task at hand are available. To facilitate the progress of deepfake detection, many efforts have been made to collect deepfakes, like for instance the FaceForensics++ [21], DeeperForensics-1.0 [14], VideoForensicsHQ [10] datasets. However, DL techniques trained on large datasets require that the data used for training matches the data analyzed when the model is deployed in operative conditions. The generalization capabilities of these techniques are in fact often poor, with the performance dropping significantly on related – but unseen – manipulations or when different deep learning architectures are used for the generation [8, 9]. In addition to the burden of collecting such a large amount of data, in some scenarios, collecting a large number of samples to train a DL model may simply be impossible. This is the case, for instance, of privacy-sensitive applications, like in the healthcare domain. In such situations, only very limited amounts of training data, collected under strict conditions, may be available. In other cases, the investigator may not have access to the model which has been used to generate the synthetic media, thus making it impossible to build a sample dataset which is large enough to train a DL architecture. Still, the investigator may rely on a few synthetic samples collected during his/her investigations [28]. Therefore, the availability of a tool that can be trained with limited available data would be of great help.

With the above ideas in mind, in this paper, we propose a feature-based method for video deepfake detection that can work in data scarcity conditions, that is, when only very few examples of the same kind are available to the forensic analyst. The proposed method relies on the analysis of motion-related features, that have been already exploited in some works in order to detect deepfake videos by means of deep learning [2, 3, 12, 23]. However, instead of relying on deep learning architectures and intensive training procedures to extract discriminative features, we base the classification on a simple footprint that relies on the frequency of motion prediction modes in the video sequence. The footprint is inspired by the one used in [7] for the detection of re-encoded (double encoded) videos. Based on our analysis of the behavior of this footprint on real and synthetic videos, we argue that, for deepfake videos, the distribution of motion prediction modes is different with respect to real videos. Moreover, in many cases, deepfake videos tend to be less predictable than real videos, thus requiring the use of a large number of *Intra* predicted frames. The footprint extracted from video sequences is used to train a Support Vector Machine (SVM) classifier in charge of discriminating between fake and real videos. Due to the simplicity of the adopted features, a simple linear SVM can reach high accuracy with very few video samples. The effectiveness of the proposed method is experimentally demonstrated on three datasets: a dataset of fake street views, DeepStreets [1], corresponding to the case of fully synthetic videos generated by Generative Adversarial Models (GANs), and two datasets of fake face videos, namely, DeeperForensics-1.0 [14] and VideoForensicsHQ [10], where the video contents are locally manipulated by means of Autoencoders.

The rest of this paper is organized as follows. In Section 2, we briefly review the state of the art in deepfake detection methods and the use of video coding-based features in Multimedia Forensics. In Section 3, we present the proposed method. Then, in Section 4, we describe the methodology we followed in our experiments. The results of the experiments we carried out to validate the proposed detector are described in Section 5. Finally, in Section 6, we conclude the paper with some final remarks and hints for future research.

## 2 Related Work

In this section, we review the main state-of-the-art about video deepfake detection. The use of video coding in multimedia forensics is also briefly discussed.

### *2.1 Video Deepfake Detection*

Plenty of methods to detect video deepfakes have been proposed in recent years. Most of these methods rely on a frame-based analysis and resort to Convolutional Neural Network (CNN) classification and automatically learned features [24, 26]. Methods combining model-based with deep learning features have also been proposed, e.g. in [16, 29].

Recently, techniques have been proposed that also exploit the temporal correlation and motion artefacts by resorting to Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM)-based CNN classifiers [12, 23]. A mixture of hand-crafted and deep-learning temporal features is used in [2, 3]. Specifically, features based on Inter-frame prediction errors have been investigated in [2] jointly with a LSTM-based network cable to learn the temporal correlation among consecutive frames. In [3], typical motion-related features, that is, the optical flow fields have been extracted and used as input of a CNN classifier. However, the performance of such approaches relying on LSTM architectures are often not superior to those achieved by frame-based methods [24]. To reduce the expensive cost of training in terms of time and resources, non-DL-based methods have also been recently proposed, to devise efficient detectors with small trained model sizes [4, 5, 33].

### *2.2 Video Coding-based Forensics*

The process of editing a video sequence always ends with the re-encoding of the edited video (possibly with a distinct codec or different coding parameters). For this reason, footprints introduced by video codecs during recompression have been extensively studied in the past years for the detection of traditional manipulations and video editing [18]. In many cases, the traces left by the double encoding process can be exposed by looking at the distribution of the macroblock prediction types in the frames of the re-encoded videos [7, 25, 30, 31]. In [7] and [31], statistics of the prediction modes have been used to detect fake high-quality videos. In particular, in [7], a very simple footprint obtained by counting the number of macroblocks of type *Intra*, *Inter* and *Skip* was successfully employed to distinguish low quality H.264 (AVC) videos, re-encoded in H.265 (HEVC), from native HEVC videos, i.e., obtained from an uncompressed video sequence.

## 3 DeepFake Detection based on Video Coding Features

Before presenting the proposed method, we introduce the main features of the H.264 video coding standard [20], which is the standard more commonly used for encoding deepfake videos, by confining the discussion to the notions that are necessary to understand the footprint adopted. It is worth observing that these features are general ones, common to all macroblock-based video coding standards, e.g., MPEG-2, MPEG-4 or H.263, and also the modern H.265 (HEVC) standard.

### *3.1 Basic Notions of (H.264) Video Coding*

During video encoding, each frame of a video stream is partitioned into macroblocks. In the following, we refer to the processing unit as macroblock. The typical size of the macroblocks is 16 × 16 pixels. The prediction types are selected on a macroblock basis rather than being the same for the entire frame. Main types of macroblocks are: *Intra*-coded (I-type) macroblocks, and *Inter*-coded macroblocks, that can be forward predicted (P-type), and bi-directional predicted (B-type). I-type macroblocks are encoded by *Intra* prediction, independently of other macroblocks, while P and B-type macroblocks are predicted via *Intra*-prediction. In the P-type, the previous frame is used for the prediction, while in the B-type case, the prediction is bi-directional, then both the previous and future frame is used. Therefore, for *Inter*-coded macroblocks, the motion vector pointing to the best matching macroblock in the reference frame is encoded, along with the prediction error (residuals), for motion compensation. A frame can belong to three different categories, defining the prediction types available to that frame: P frames, that contain I- or P-type macroblocks; B frames, that contain I-, P- or B-type macroblocks; I frames, that contain I-type macroblocks only. In early standards, motion compensation was performed with one motion vector per macroblock. In H.264/AVC, like in other modern standards, a macroblock can be split into multiple variable-sized prediction blocks (up to 4 × 4), called partition units, based on the content of the frame. Then, a separate motion vector is specified for each partition of the *Inter*-predicted macroblock.

To further improve the coding efficiency in P- and B-frames, the H.264 and H.265 standards resort to the *Skip* mode. A *Skip* macroblock consists of a single prediction unit whose motion data is derived from the neighboring macroblocks. No residuals are transmitted in *Skip* mode.

In H264 and H265 video coding standards, the quality of the compressed video (or compression rate/level) is set by means of the Constant Rate Factor (CRF). The Raw format refers to the case CFR = 0. In High Quality (HQ) and Low Quality (LQ) video formats, the videos are compressed by the video codec with a CFR respectively of 23, and 40 [22].

### *3.2 The Proposed Video Coding Feature*

As we mentioned, the common procedure of generation of fake videos using generative models, like Generative Adversarial Networks (GANs) or Autoencoders, introduce both spatial (pixel-level) as well as motion artefacts. By inspecting fake and real videos with similar contents, we can observe that the movements in fake videos tend to be less fluent and natural with respect to the case of real (original) videos. Therefore, we hypothesize that, in many cases, fake videos tend to be less predictable than real videos. We argue that such differences can be captured by video coding features. In particular, our conjecture is that the different behavior can be reflected by looking at the prediction modes of the macroblocks. More specifically, we focused on the prediction modes in P- and B-frames and computed their frequency on a video (sub)sequence or batch of frames. For each frame, we extract the following footprint: $F = [f_{Intra}, f_{Inter}, f_{Skip}]$, where $f_x$ denote the frequency/percentage of macroblocks of mode $x$ in the frame. Frequency $f_{Inter}$ counts for both the P-type and the B-type macroblocks. Notice that a similar feature has been used in [7] for the detection of low quality H.264 video contents re-encoded H.265 with higher quality.

To verify our hypothesis, we computed $F$ for real and fake H.264 encoded video sequences. Comparison is carried out for the same video quality, e.g., for the same rate of compression or CRF. We observed that, as argued, when the fake videos are fully GAN synthesized, the frequencies of the prediction modes are very different with respect to the case of real videos, and the distribution of the three features is very different. Figure 1 reports the distribution of $f_{Intra}$, $f_{Inter}$, and $f_{Skip}$ averaged on all the frames of the raw videos for the three sets of the DeepStreets dataset [1], respectively for real (left) and fake (right) videos with similar content. In particular, by looking at the figure, we see that, for fake videos, *Intra*-coded macroblocks represent the large majority of the macroblocks, and their percentage is by far larger for fake (74.7%) than for real (43.8%) videos. This confirms our conjecture that fake videos are less predictable, requiring more *Intra*-coded macroblocks. Notably, the same behavior is not observed when the fake videos are generated by object swapping method, in which case we verified that the percentage of *Intra*-coded macroblocks is similar for fake and real. Since those methods only modify a limited part of the frames via autoencoders, e.g., the faces or only the facial expressions, while the rest of the frame remains the same, deepfake videos generated in this way are not fully synthetic videos.[1] However, even in this case, discrimination is still possible based on the proposed feature, by looking at the joint behavior of the three features, and in particular at their distribution over batches of $N$ consecutive frames. Figure 2 shows the feature distribution after dimensionality reduction via Principal Component Analysis (PCA) for real and fake videos coming from the DeeperForensics-1.0 and VFHQ dataset. In order to apply PCA, for each video, the $(N \times 3)$-dimensional tensor with the $N$ 3D footprints computed on the $N$ frames, is reshaped to a vector of length $3N$. In the plot, $N$ is set to 30. The reduced dimension is equal to 3.

Motivated by the above analysis, we used the proposed feature to train an SVM classifier to discriminate between real and deepfake videos under data limited conditions. The details of the SVM training are provided in Section 4.2.

[1] Since the footprint relies on (spatially) global features, it is not surprising that the best discrimination is achieved in the fully synthetic case.

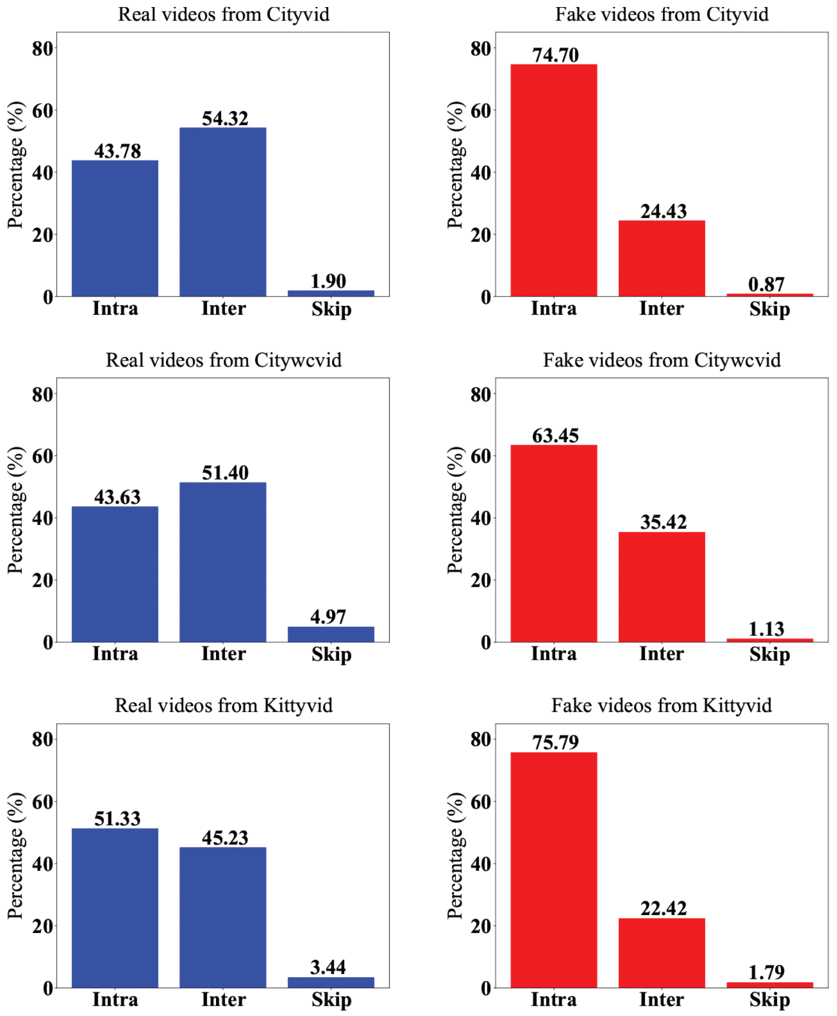

Figure 1: Frequency of *Intra*, *Inter* and *Skip* for raw videos in the DeepStreets dataset.

## 4 Methodology

In this section, we describe the methodology we followed in our experiments.

### *4.1 Datasets*

To investigate the effectiveness of the proposed detector, we considered a dataset of fake street views (fully synthetic) and two datasets of Deepfake face videos, described in the following.

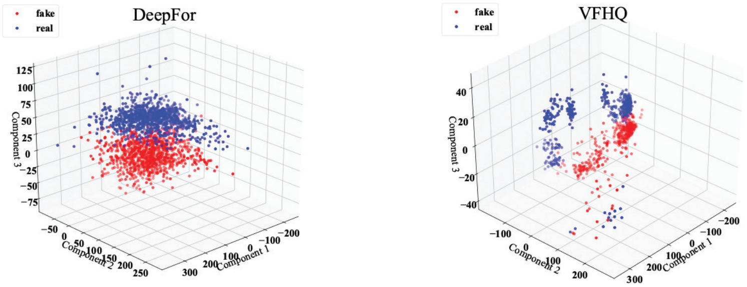

Figure 2: Feature distribution for 1000 videos in the DeeperForensics-1.0 dataset and 394 videos in the VFHQ dataset.

**DeepStreets** [1] is a GAN-synthesized street video dataset. The fake videos are generated from a sequence of semantic segmentation masks. The dataset consists of 3 subsets; Cityvid, Citywcvid and Kittyvid. The fake videos are generated using the Vid2vid [27] model (for Cityvid and Kittyvid) and the Wc-vid2vid [17] models (for Citywcvid), both trained on the Cityscapes dataset. For the real videos, in the Kittyvid case, they are taken from Kitti dataset [11], while in the other cases they are taken from the Cityscapes [6] dataset.

Each subset contains 200 real videos and 200 fake videos and the average length of the videos is 3 s. The subsets are made available for three different qualities or compression levels of the videos, namely Raw, HQ and LQ.

**DeeperForensics-1.0 (DeepFor)** [14] is a new large scale face swapping dataset consisting of 60,000 manipulated videos in total. For our tests, we considered 1000 raw manipulated videos and 1000 corresponding real videos, that are taken from YouTube. The minimum video length is 6 s.

**VideoForensicsHQ (VFHQ)** [10] is a high quality face reenactment dataset, where only the face is manipulated. Specifically, the facial expression of the source video is transferred to the target video. The dataset is comprised of 1737 videos in total (1141 real videos and 596 fake videos) representative for 8 different emotions. The video duration varies from 3 s to more than 5 min. To get balanced data, we considered 596 real and fake videos from this dataset.

Some examples of videos from each of the three datasets are reported in Figure 3.

### *4.2 SVM Detection, Training and Setting*

From our discussion in Section 3.2 we argue that, in some cases (e.g., with fully GAN-synthesized videos), a simple feature could be extracted from $F$ and directly used for the detection, e.g. the difference $(f_{Intra} - f_{Inter})$. However,

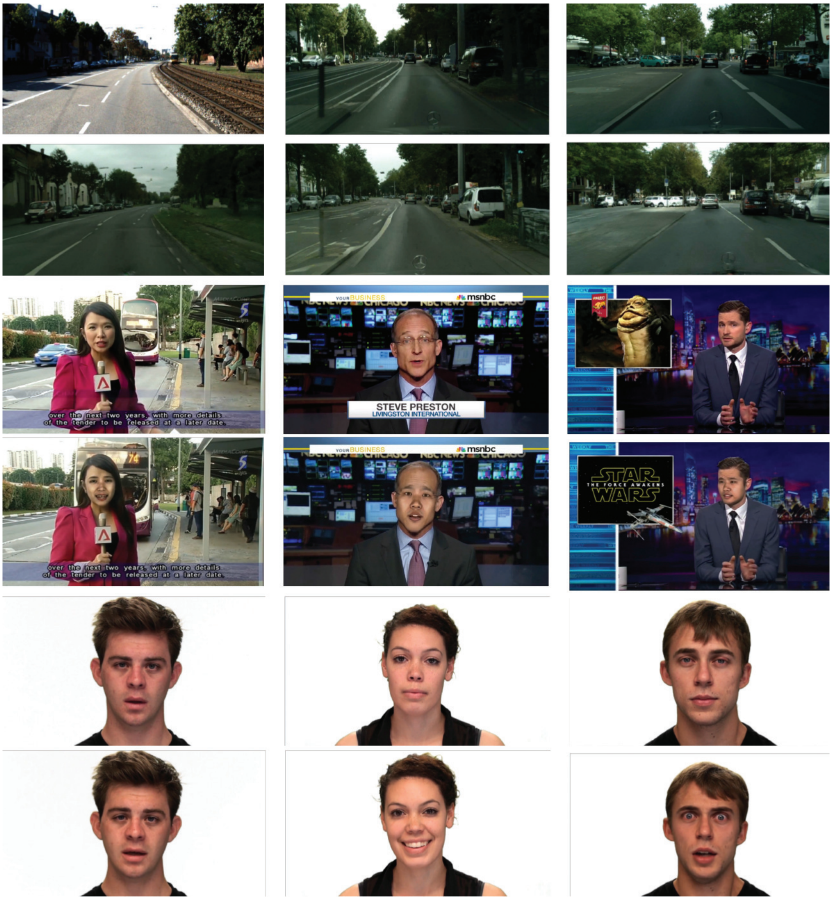
over the next two years with more details of the tender to be released at a later date
msnbc
STEVE PRESTON
LIVINGSTON INTERNATIONAL
STAR WARS
THE FORCE AWAKENS

Figure 3: Examples of real and fake videos from the three datasets. From top to bottom row: 3 real and 3 fake from DeepStreets, 3 real and 3 fake from DeepFor, and 3 real and 3 fake from VFHQ.

this approach would not work with non fully-synthetic fakes, e.g., in the swapping case. Since the information on the synthetic nature of the fake is unknown a priori, we resort to a more general approach and train an SVM classifier.

Given a video $v_i$, the footprint $F_i$ is extracted from the first $N$ frames of the video, skipping the first frame that is always Intra-coded.[2] Let $f_{ij} = [f_{Intra_{ij}}, f_{Inter_{ij}}, f_{Skip_{ij}}]$ be the footprint extracted from frame $j$. We denote

[2] For all videos, the $N$ frames following the first one are always P- or B-type, for the values of $N$ considered in our experiments.

with $F_i = [f_{i1}, f_{i2}, \dots, f_{iN}]$ the concatenation of the $N$ footprints. Then, $F_i$ is the resulting feature vector extracted from video $v_i$, having dimensionality $N \times 3$. This vector is extracted from each video of the dataset $\mathcal{C}$, consisting of $c$ videos, and used to train a linear SVM classifier. The regularization parameter $C$ defining the separation margin adopted for the classification is set to 0.1. The procedure is reported in Algorithm 1. Notation $Y$ is used to indicate the label vector associated to the videos in $\mathcal{C}$, that is, $Y = (y_1, \cdots, y_c)$. We denote the trained SVM model with $F$-SVM.

**Algorithm 1:** $F$-SVM

```
Input: $\{(v_i, y_i)_{i=1}^{c}\}$, $v_i \in \mathcal{C}$, $y_i \in$ {'real','fake'}
Output: $F$-SVM model
$F \leftarrow 0$
for $i \leftarrow 0$ to $\mathcal{C}$ do
    for $j \leftarrow 1$ to $N$ do
        extract $f_{Intra_{ij}}, f_{Inter_{ij}}, f_{Skip_{ij}}$    // Using Elecard StreamEye software
        $f_{ij} \leftarrow [f_{Intra_{ij}}, f_{Inter_{ij}}, f_{Skip_{ij}}]$
    $F_i \leftarrow [f_{i1}, f_{i2}, \dots, f_{iN}]$
$F$-SVM $\leftarrow$ train LinearSVM( C = 0.1, $F$, $Y$ )    // Using LibSVM
return $F$-$SVM$
```

An SVM model was trained and tested for all the 3 datasets described above. For DeepStreets, 50 out of 200 per class videos are used for testing, while the remaining 150 are used for training. Actually, only 128 are made available for training, while the remaining ones are left out for validation, that is performed for the DL model used for comparison. For DeepFor, the number of videos reserved for testing is 201 out of 1000 per class, while 703 are made available for training (the remaining 96 for validation). For VFHQ, testing is performed with 89 out of 596 videos per class, with 394 made available for training (113 for validation). In each case, the SVM is trained using different numbers of videos from the training set.

Since videos have different lengths and some of them are very short, for simplicity, for every video only the first $N = 30$ frames of the stream are considered (with the exclusion of the first frame which is Intra-coded), discarding the rest of the video. In the scenario addressed in this paper, we assume that the quality of the videos is matched in testing and training. In practice, this corresponds to reading the video quality from the encoded stream and using a tool trained for the same quality. To consider always the same codec format, all the videos have been re-encoded H.264 with maximum quality (CRF = 0) before performing our video coding analysis. Other information about the framework considered in our experiments is in order: the FFmpeg software is adopted to encode a video to H.264 codec format, while for the analysis of the video, we used the Elecard StreamEye software.[3]

Being a feature-based method, that is, a method based on handcrafted features, the proposed method should (expectedly) require less training data with respect to data-driven methods, that have also to learn suitable features for the classification during the training process. Moreover, due to the simplicity of the footprint and its low dimensionality, a simple linear SVM can be used for the classification (with the two parameters defining the separating hyperplane being the only parameters to be trained). Hence, we expect that very few videos are enough to train our model, with a significant gain in terms of the required amount of training videos.

### *4.3 Comparison with the State-of-the-art*

To prove that the proposed detector achieves better results than deep learning-based methods in data scarcity conditions, the comparison is carried out with a frame-based XceptionNet detector, that is the state-of-the-art for DeepStreet detection [1], a ResNetLSTM network [10] [13] and a very recent non DL-based method named DeFakeHop [5], proposed for face manipulation detection. For a proper comparison, the same splitting of the three datasets has been used to define the training, validation and test data.

- **XceptionNet** is a network with Depthwise Separable Convolutions and it represents the state-of-the-art for DeepStreets video detection [1]. The network consists of three modules: entry flow, middle flow and exit flow. In our work for comparison, we trained the model from scratch, considering the first $N$ frames from each video. The input face image is resized to 299 × 299. The network was trained using Adam optimizer with learning rate $\alpha = 0.0002$ and the default values for the first and second-order moments, that is, $(\beta_1 = 0.9, \beta_2 = 0.999)$. The batch size was set to 24 frames and the model was trained for 50 epochs. During testing, the decision on each video is taken by means of soft majority voting, i.e., considering the average of the detection scores of all the frames.

- **DeFakeHop** is a light-weight deepfake detection method that can achieve high deepfake video detection accuracies with a small model size and a fast training procedure. The detection is conducted based on features extracted from patches (e.g., the left eye, right eye and mouth) by PixelHop++ module and further processed by a feature distillation module with the output of the probability for the patches. Finally, the

[3] https://www.elecard.com/products/video-analysis/streameye.

probabilities for all patches are integrated to get the final description of whether the face is fake. For the comparison, we trained the model by taking $N$ frames per video. The detection results are reported in video-level.

- **ResNetLSTM** is a network that combines ResNet and LSTM thus enabling the learning of temporal clues for deepfake detection. More specifically, the ResNet50 network is used to extract spatial features from face images. Afterwards, the extracted features from the last convolution layer are further passed through one convolutional layer (C = 128, kernel = $(1, 1)$), adaptive pooling layer, and LSTM module with 512 hidden units to learn temporal discrepancy between frames. Finally, a fully connected layer is used for fake prediction. Similarly, for each video, we take $N$ sequences with the length of 7 from $N$ frames per video to train the model, and the input face is resized to $224 \times 224$. The loss is calculated for all the frames during the training. And only the features of the last frame are used for prediction during the test. For the other parameters, We used the same settings as for XceptionNet training.

## 5 Experimental Results

We first demonstrate the effectiveness of the proposed method on the DeepStreets dataset of fully-synthetic videos. Then, we also test the effectiveness of the proposed method on the DeepFor and VFHQ datasets, where only a limited area of the video frames is manipulated and then has synthetic nature.

### *5.1 Comparison Results*

Table 1 reports the accuracy values achieved by the state-of-the-art XceptionNet [1] and the proposed method for different numbers of videos $c$ used to train the classifier (the number reported refers to each class). Results are reported for different compression levels (Raw, HQ, LQ). We see that, for the Cityvid and Citywcvid subsets, in all the cases, the proposed detector can achieve an accuracy of around 98% by training on just 4 videos, while XceptionNet needs 64 videos (32, in very few cases). For Kittyvid subset, the proposed method can not achieve the same level of accuracy as the state-of-the-art for a large number of training videos, which means that the features are not that discriminative to detect the fake videos from the reals. The lower discrimination capability, in this case, can also be argued by looking at the feature distribution of Kittyvid dataset in Figure 1. A reason can be that in the case of Kittyvid sub-dataset, the pristine videos come from a completely different dataset with different

Table 1: Detection accuracy for the three different subsets in the DeepStreets for both our method and the state-of-the-art XceptionNet method in [1]. Results are reported for different video qualities (Raw, HQ and LQ).

|  | Cityvid |  |  |  |  |  | Citywcvid |  |  |  |  |  | Kittyvid |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Raw |  | HQ |  | LQ |  | Raw |  | HQ |  | LQ |  | Raw |  | HQ |  | LQ |  |
| train $c$ | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept | $F$-SVM | Xcept |
| 2 | **96.9** | 50.0 | **97.5** | 50.0 | **95.8** | 50.0 | **96.7** | 50.0 | **98.6** | 50.0 | **99.0** | 50.0 | **80.0** | 50.0 | **75.0** | 50.0 | **67.8** | 50.0 |
| 4 | **97.2** | 50.0 | **98.5** | 50.0 | **98.3** | 50.0 | **98.5** | 55.0 | **98.9** | 67.0 | **99.6** | 52.0 | **81.1** | 50.0 | **76.2** | 50.0 | **71.0** | 50.0 |
| 8 | **98.5** | 88.0 | **98.4** | 78.0 | **99.5** | 84.0 | **98.5** | 70.0 | **99.8** | 71.0 | **99.7** | 84.0 | 82.1 | **99.0** | 81.2 | **100** | 72.1 | **100** |
| 16 | **98.6** | 90.0 | **99.0** | 94.0 | **99.5** | 81.0 | **99.7** | 90.0 | **100** | 87.0 | **99.8** | 92.0 | 85.0 | **100** | 82.8 | **100** | 74.2 | **100** |
| 32 | **98.8** | 97.0 | **99.1** | 66.0 | **99.7** | 89.0 | **99.9** | 96.0 | **100** | 97.0 | **99.9** | 97.0 | 86.8 | **100** | 86.8 | **100** | 76.6 | **85.0** |
| 64 | **99.4** | 87.0 | 99.4 | **100** | **99.7** | 99.0 | **100** | **100** | **100** | **100** | **100** | 86.0 | 87.9 | **100** | 85.3 | **100** | 79.5 | **99.0** |
| 128 | 99.0 | **100** | **100** | **100** | **100** | 96.0 | **100** | **100** | **100** | **100** | **100** | **100** | 91.6 | **100** | 91.3 | **100** | 81.5 | **100** |

background and scene (temporal) variability with respect to the fake images. The different temporal variability might affect our video coding features. On the contrary, the significantly different content and background in real and fake videos in the Kittyvid case is beneficial for XceptionNet which can look at the diversity of the background to discriminate between real and fake videos. This explains why better performance can be achieved by XceptionNet on Kittivid with respect to the other cases. Nevertheless, the superior performance of the method under limited training data is confirmed also in this case, with the proposed detector reaching an accuracy of around 80% with 2 videos, for high quality videos.

We also performed a cross-data analysis for the three subsets in DeepStreets, whose results are reported in Table 2 for raw videos. The results refer to the case of training with 120 videos when both our and XceptionNet methods can achieve good performance. From the table we see that the proposed method generalizes extremely well, improving the poor generalization capability of the state-of-the-art method, with the exception of the Kittyvid case where, however, the results of our methods are low also in the matched case, see Table 1.

Table 2: Cross dataset results for DeepStreets on raw videos (120 videos are used for training).

| | | Train set | | |
|---|---|---|---|---|
| Test set | Method | Cityvid | Citywcvid | Kittyvid |
| Cityvid | Xcept | – | 73.0 | 50.0 |
| | *F*-SVM | – | **99.3** | **80.0** |
| Citywcvid | Xcept | 49.0 | – | 50.0 |
| | *F*-SVM | **99.0** | – | **53.8** |
| Kittyvid | Xcept | **99.0** | **75.0** | – |
| | *F*-SVM | 85.5 | 69.0 | – |

The performance achieved on the DeepFor and VFHQ dataset is reported in Table 3. In the case of DeepFor, we see that accuracy of 91.4% can be achieved with our method by training with only 8 videos, while XceptionNet, ResNetLSTM and DeFakeHop need respectively 128, 16 and more than 256 videos to reach a similar value of accuracy (92%). Notably, the performance of our method is already close to the maximum around $c = 32$, and the accuracy does not increase further using many more videos, while the performance of other methods increases while increasing $c$. On the VFHQ dataset, the difference is lower, however, the good performance of the proposed method with few videos is confirmed. The proposed F-SVM has much better performance than XceptionNet and DeFakeHop, reaching an accuracy of 90.1% with 16

Table 3: Detection accuracy for DeepFor and VFHQ datasets for our method and the compared methods.

| No. train | DeepFor | | | | VFHQ | | | |
|---|---|---|---|---|---|---|---|---|
| videos ($c$) | $F$-SVM | Xcept | DeFakeHop | ResNetLSTM | $F$-SVM | Xcept | DeFakeHop | ResNetLSTM |
| 2 | 67.5 | 50.0 | 61.3 | 65.9 | 61.7 | 50.0 | 60.6 | 58.4 |
| 4 | 74.8 | 50.0 | 62.8 | 70.6 | 75.5 | 50.0 | 63.8 | 77.5 |
| 8 | **91.4** | 50.0 | 62.0 | 85.9 | 84.8 | 50.0 | 69.9 | **93.3** |
| 16 | **95.5** | 53.0 | 72.8 | **91.7** | **90.1** | 73.6 | 72.6 | **90.8** |
| 32 | **96.5** | 56.0 | 79.8 | **97.1** | **92.3** | **91.0** | 77.8 | 88.6 |
| 64 | **96.8** | 83.0 | 85.6 | **99.0** | **93.5** | **94.9** | 80.9 | 88.3 |
| 128 | **96.8** | **92.3** | 88.5 | **98.0** | **94.4** | **95.5** | 79.8 | **93.2** |
| 256 | **97.0** | **98.5** | 89.7 | **99.9** | **96.2** | **94.4** | 82.7 | **94.3** |

videos. However, in this case, ResNetLSTM can get an accuracy larger than 90% with just 8 videos. Since in the case of VFHQ dataset only the facial expressions are manipulated, with the head poses remaining the same, it is not surprising that the detection based on the proposed footprint is more difficult. Moreover, in the VFHQ dataset, there is significant motion in the background. However, since the background is not manipulated, the prediction modes from the macroblocks in the background negatively affect the footprint estimation for fake videos.

A noticeable advantage of the proposed methods over the state-of-the-art methods is also in terms of running (training) time and size of the trained model. Table 4 reports the training time for the various methods computed for the case of 16 videos for the DeepFor and VFHQ datasets. For our method, the time needed for the feature extraction is also considered.[4] By relying on a simple linear SVM classifier trained on the extracted low dimensional footprints, the training of our method is extremely fast (with only two parameters to be learnt from the data), much faster than the other methods. The size of the

Table 4: Comparison in terms of training time, no. of parameters and model size on DeepFor and VFHQ datasets for the case of 16 training videos.

| Methods | F-SVM | Xcept | DeFakeHop | ResNetLSTM |
|---|---|---|---|---|
| Number of parameters | 2 | 22.8M | 42.8K | 25.1M |
| Training time (DeepFor) | 0.7281s | 18m34s | 23.7251s | 1h15m24s |
| Training time (VFHQ) | 0.8584s | 17m45s | 27.9549s | 1h17m45s |
| Model size (DeepFor&VFHQ) | 10KB | 79.7MB | 76KB | 96.0MB |

[4] For the other methods, only the training time is considered since the data pre-processing time (that is, the time for face detection and extraction - for XceptionNet and ResNetLSTM - and the time for landmark detection and patch - eyes and mouth - extraction for DeFakeHop) is negligible compared to the training time.

trained models, which is the same for the two datasets, is also reported in the table, showing that our SVM trained model has a very small size compared to the other models.

## 6 Conclusions

We proposed a feature-based method for video deepfake detection that can work in data scarcity conditions, that is, when only very few examples are available to the forensic analyst. The method is based on a simple video coding feature, counting the frequency of motion prediction modes in the video sequence. Results carried out on three different datasets show the effectiveness of the proposed method, that only needs very few video samples to train the detector. Future works will focus on further improving the performance in the case of manipulated videos (non fully synthetic), where only a region of the frames is modified, e.g., the face swapping case, by confining the extraction of the footprint to the foreground region. A more comprehensive assessment of the generalization capability of the proposed detector based on video coding analysis will also be performed. Finally, the robustness of the detector against video re-compression and more in general intentional attacks, e.g., adversarial attacks, is also worth investigation.

## Acknowledgement

This work has been partially supported by the China Scholarship Council (CSC), file No. 202008370186, by the PREMIER project under contract PRIN 2017 2017Z595XS-001, funded by the Italian Ministry of University and Research, and by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

## References

[1] O. Alamayreh and M. Barni, “Detection of GAN-Synthesized Street Videos,” in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, 811–5, DOI: 10.23919/EUSIPCO54536.2021.9616262.

[2] I. Amerini and R. Caldelli, "Exploiting Prediction Error Inconsistencies through LSTM-based Classifiers to Detect Deepfake Videos," in *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, 2020, 97–102.

[3] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake Video Detection through Optical Flow based CNN," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, 0–0.

[4] H.-S. Chen, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop++: An Enhanced Lightweight Deepfake Detector," *arXiv preprint arXiv:2205.00211*, 2022.

[5] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "Defakehop: A Light-Weight High-Performance Deepfake Detector," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 3213–23.

[7] A. Costanzo and M. Barni, "Detection of Double AVC/HEVC Encoding," in *2016 24th European Signal Processing Conference (EUSIPCO)*, IEEE, 2016, 2245–9.

[8] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "Forensictransfer: Weakly-Supervised Domain Adaptation for Forgery Detection," *arXiv preprint arXiv:1812.02510*, 2018.

[9] M. Du, S. Pentyala, Y. Li, and X. Hu, "Towards Generalizable Deepfake Detection with Locality-Aware Autoencoder," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, 325–34.

[10] G. Fox, W. Liu, H. Kim, H.-P. Seidel, M. Elgharib, and C. Theobalt, "Videoforensicshq: Detecting High-quality Manipulated Face Videos," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[12] D. Güera and E. J. Delp, "Deepfake Video Detection using Recurrent Neural Networks," in *2018 15th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, IEEE, 2018, 1–6.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 770–8.

[14] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, “Deeperforensics-1.0: A Large-scale Dataset for Real-world Face Forgery Detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 2889–98.

[15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of Stylegan,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8110–9.

[16] Y. Li, M.-C. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2018, 1–7.

[17] A. Mallya, T.-C. Wang, K. Sapra, and M.-Y. Liu, “World-Consistent Video-to-Video Synthesis,” *arXiv preprint arXiv:2007.08509*, 2020.

[18] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, “An Overview on Video Forensics,” *APSIPA Transactions on Signal and Information Processing*, 1, 2012.

[19] S. Mo, M. Cho, and J. Shin, “Instance-aware Image-to-Image Translation,” in *International Conference on Learning Representations*, 2019, https://openreview.net/forum?id=ryxwJhC9YX.

[20] I. E. Richardson, *The H. 264 Advanced Video Compression Standard*, John Wiley & Sons, 2011.

[21] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Faceforensics++: Learning to Detect Manipulated Facial Images,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 1–11.

[22] V. Sze, M. Budagavi, and G. J. Sullivan, “High Efficiency Video Coding (HEVC),” in *Integrated Circuit and Systems, Algorithms and Architectures*, Vol. 39, Springer, 2014.

[23] S. Tariq, S. Lee, and S. S. Woo, “A Convolutional LSTM based Residual Network for Deepfake Video Detection,” *arXiv preprint arXiv:2009.07480*, 2020.

[24] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection,” *Information Fusion*, 64, 2020, 131–48.

[25] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesaña, A. Piva, and M. Barni, “Detection of Video Double Encoding with GOP Size Estimation,” in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2012, 151–6.

[26] L. Verdoliva, “Media Forensics and Deepfakes: An Overview,” *IEEE Journal of Selected Topics in Signal Processing*, 14(5), 2020, 910–32.

[27] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-Video Synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[28] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a Few Examples: A Survey on Few-shot Learning," *ACM Computing Surveys (CSUR)*, 53(3), 2020, 1–34.

[29] X. Yang, Y. Li, and S. Lyu, "Exposing Deep Fakes using Inconsistent Head Poses," in *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 8261–5.

[30] H. Yao, S. Song, C. Qin, Z. Tang, and X. Liu, "Detection of Doublecompressed H. 264/AVC Video Incorporating the Features of the String of Data Bits and Skip Macroblocks," *Symmetry*, 9(12), 2017, 313.

[31] Y. Yu, H. Yao, R. Ni, and Y. Zhao, "Detection of Fake High Definition for HEVC Videos based on Prediction Mode Feature," *Signal Processing*, 166, 2020.

[32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-image Translation using Cycle-consistent Adversarial Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2223–32.

[33] Y. Zhu, X. Wang, H.-S. Chen, R. Salloum, and C.-C. J. Kuo, "A-PixelHop: A Green, Robust and Explainable Fake-Image Detector," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 8947–51.