LEARNING DOUBLE-COMPRESSION VIDEO FINGERPRINTS LEFT FROM SOCIAL-MEDIA PLATFORMS

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

Sapienza University of Rome[1] ELIS Innovation Hub[2]

ABSTRACT

Social media and messaging apps have become major communication platforms. Multimedia contents promote improved user engagement and have thus become a very important communication tool. However, fake news and manipulated content can easily go viral, so, being able to verify the source of videos and images as well as to distinguish between native and downloaded content becomes essential. Most of the work performed so far on social media provenance has concentrated on images; in this paper, we propose a CNN architecture that analyzes video content to trace videos back to their social network of origin. The experiments demonstrate that stating platform provenance is possible for videos as well as images with very good accuracy.

Index Terms— Social networks, video forensics, deep learning, multitask learning, platform provenance analysis.

1. INTRODUCTION

In recent years multimedia content has become one of the predominant ways for exchanging information. Every day people watch over a billion hours of video on YouTube [1] and share more than a billion stories on Facebook [2]. The expressiveness of visual content makes multimedia a powerful means of communication. Therefore, it becomes increasingly important to be able to verify the source of this information.

When uploaded and shared across social networks and messaging apps, multimedia content undergoes a processing step in which the platforms perform a set of operations on the input. Indeed, to optimize transfer bandwidth as well as display quality, most platforms apply specific compression and resizing methods. These methods, which tend to be unpublished, differ among the different social platforms [3]. All these operations inevitably leave some traces on the media content itself [4, 5, 6]. The social media identification problem has been widely studied for image files with promising results [3, 7, 8], employing machine learning classifiers. Recently, Quan et al. [9] showed that by using convolutional methods it is possible to recognize Instagram filters and attenuate the sensor pattern noise signal in images. Amerini et al. [10] introduced a CNN for learning distinctive features among social networks from the histogram of the discrete cosine transform (DCT) coefficients and the noise residual of the images. Phan et al. [11] proposed a method to track multiple image sharing on social networks by using a CNN architecture able to learn a combination of DCT and metadata features. Nevertheless, the identification of the traces left by social networks and messaging apps on video contents remains an open problem. Recently, Iuliani et al. [12] presented an approach that relies on the analysis of the container structure of a video through the use of unsupervised algorithms to perform source-camera identification for shared media with high performance; their method is strictly dependent on the file structure, whereas in our work we are interested in approaches that are based on the content of a video, independently of the file type. Kiegaing and Dirik [13] showed that fingerprinting the I-frames of a flat content native video can be used to accurately identify the source of YouTube videos. Moreover, although the research community has treated video and image forensics as separate problems, a recent work from Iuliani et al. [14], demonstrates that it is possible to identify the source of a digital video by exploiting a reference sensor pattern noise generated from still images taken by the same device, suggesting that it could be possible to link social media profiles containing images and videos captured by the same sensor.

In this work, we propose a multistream neural network architecture that can capture the double compression traces left by social networks and messaging apps on videos. According to our knowledge, this is the first work that investigates whether it is possible to recognize videos from different social networks by analyzing the traces of compression left by these sites when loading content. The possibility of reconstructing information on the sharing history of a certain object is highly valuable in media forensics. In fact, it could help in monitoring the visual information flow by tracing back the initial uploads, thus aiding source identification by narrowing down the search. This could be helpful in different applications such as, for example, cyberbullying, where we want to be able to investigate who and where this individual has shared a certain content. Similarly, this tool could be help-

* Supported by the ERC Advanced Grant 788893 AMDROMA, the EC H2020 RIA project "SoBigData++" (871042), and the MIUR PRIN project ALGADIMAR.

ful to trace the sharing of videos of military propaganda or other criminal activity back to the source, as well as for fact checking and countering fake news.

The problem of classifying photos and videos from social networks has been typically treated separately. To overcome this limitation, here we investigate the possibility to test the robustness of our implementation with respect to images once the network is trained on videos. The rest of the paper is organized as follows: Section 2 describes our approach. Section 3 discusses different experimental results. Finally, Section 4 draws the conclusions of our work.

2. PROPOSED METHOD

In video coding, a video is represented as a sequence of *groups of pictures* (GOP)s, each of which begins with an *I-frame*. I-frames are not predicted from any other frame and are independently encoded using a process similar to JPEG compression. Apart from the I-frames, the rest of each GOP consists of *P-frames* and *B-frames*. These frames are predictively encoded using motion estimation and compensation. Thus, these frames are derived from segments of an anchor I-frame and represent lower quality frames. In this section we describe the proposed architecture (see Figure 1) composed by a two-stream network, inspired by the work by Nam et al. [15]. However, the application of this particular network to the problem that we study is novel and it requires some important modifications to the method in [15]. First, we modified the third convolutional block of the Ind-Net removing a stack of Convolutional, Batch Normalization, and ReLU operations and we added one more convolutional block (Block 6) at the end of the CNN. This deeper configuration helps the network to capture more subtle details in the input. Next, we modified the Pred-Net by doubling the number of operations in each block and increased the number of output channels of each block in order to learn a richer representation. Finally, we changed the dimensionality of the flattened feature maps from 128 to 256 for the P-frames stream and from 16,384 to 4,096 for the IF-stream. This helps to limit the importance of I-frames over the P-frames. We choose not to include B-frames in our analysis because of the lower quality of these kind of frames. Finally, we introduce a two-stream network (MultiFrame-Net), which learns the inter-modal relationships between features extracted from both types of frames. In the rest of this section, we use the notation $W \times H$ to denote the resolution of a video v. Each video can also be represented by N frames denoted as $f_0, \ldots, f_{N-1}$, where $f_j \in \mathbb{Z}^{3\times W\times H}$. Moreover, we use the notation $f_{Ii}^{(v)}$ and $f_{Pi}^{(v)}$ to denote the ith I-frame or P-frame, respectively, of a video v.

2.1. Ind-Net

In this section we propose a network that analyzes the I-frames of a video. The network is depicted in the bottom part of Figure 1. We designed a network that consists of six convolutional blocks that act as a feature extractor and a fully connected network that takes the input feature vector and produces an output classification. The first tree convolutional blocks made of (1) two consecutive stacks of convolution (Conv2D), batch normalization (BatchNorm), and ReLU operations, and (2) a final max pooling (MaxPool) layer. The last three convolutional blocks are organized in three consecutive stacks of (1) Conv2D, BatchNorm, and ReLU operations, and (2) a final MaxPool layer. Apart from the first convolutional layer, which has a 5×5 kernel, all other convolutional layers have a 3×3 kernel. The feature extracted by the last MaxPool layer becomes eventually flattened and passed through two stacks made by a 512-dimensional fully connected layer and a ReLU, and a final 512-dimensional fully connected layer followed by a softmax one. The network outputs a $|C|$-dimensional vector, where $|C|$ is the number of output classes.

Before being fed into the network, the decompressed I-frames are initially transformed through a preprocessing module. To highlight the traces left by double compression, we employ the high-pass filter introduced by Fridrich and Kodovsky [16, operator S5a], and used in [15] and apply it to the Y-channel of the input after RGB-to-YUV conversion. Therefore, we denote as $X_{Ii} = \{f_{Ii}^{(v)}\} \in \mathbb{Z}^{3\times W\times H}$ the input ith frame of video v and compute $X'_{Ii} = \{HPF(Y(f_{Ii}^{(v)}))\} \in \mathbb{Z}^{W\times H}$ to obtain the preprocessed input of the network, where $HPF(\cdot)$ indicates the high pass filter and $Y(\cdot)$ indicates the Y-channel of the input frame. Because we assume that each video could come from a single social media platform, we train the model using a cross-entropy loss function, thus training the model to output a probability over the $|C|$ classes for each video.

2.2. Pred-Net

Now we present Pred-Net, a new network architecture that analyzes the P-frames of a video to detect double compression fingerprints. The network (depicted in the top of Figure 1) is made of five convolutional blocks and a fully connected network. All the convolutional blocks consist of two stacks of (1) Conv2D, BatchNorm, and ReLU operations, and (2) a final average pooling (AvgPool) layer. The AvgPool and GlobalAvgPool levels help to preserve the statistical properties of feature maps that could otherwise be distorted with the MaxPool. All the Conv2D layers in the first two blocks have a 5×5 kernel, and the last three blocks have a a 3×3 kernel. Finally, the feature maps extracted from the last convolutional block are flattened and passed through a 256-dimensional fully connected layer that outputs a $|C|$-dimensional vector and a softmax operation that calculates the output prediction.

Similarly to the Ind-Net, we add a preprocessing step to the input frames in which a high-frequency–component extraction operation is applied to eliminate the influence

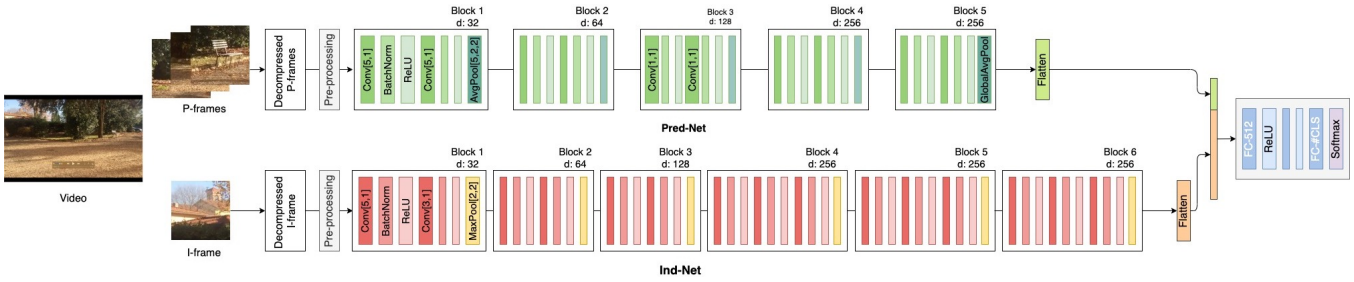

Fig. 1. The proposed two-stream network (MultiFrame-Net) architecture. The network is constructed by concatenating the feature maps of the Ind-Net and the Pred-Net. The I-frame and P-frame streams are trained separately. Next, we concatenate the flattened output of the two-streams and train a fully connected classifier.

of diverse video contents. Further, because the P-frames represent predicted low-quality frames, we compensate for the loss of information by stacking consecutive frames. In fact, given a stack of three consecutive P-frames denoted as $X_{Pi} = \{f_{Pi-1}^{(v)}, f_{Pi}^{(v)}, f_{Pi+1}^{(v)}\} \in \mathbb{Z}^{3\times3\times W\times H}$, we compute $X'_{Pi} = \{Y(f) - G(f) | f \in X_{Pi}\} \in \mathbb{R}^{3\times W\times H}$, where the function $G(\cdot)$ denotes a Gaussian filter. Like the Ind-Net, the network is trained with a cross-entropy loss function.

2.3. MultiFrame-Net

Multistream architectures have been successfully applied by multimedia forensics researchers for both forgery detection and source identification tasks [17, 18, 19, 10]. Therefore, we combine the feature maps of both Ind-Net and Pred-Net to feed the fully connected classifier with inter-modal relationships between different types of frames. As shown in Figure 1, we concatenate the output features maps of the two CNNs and feed them to the classifier. The concatenated features vector is a $4,352$-dimensional vector obtained by integrating the $4,096$-dimensional output vector of the Ind-Net and the 256-dimensional output vector of the Pred-Net.

In our setting, we train the the Ind-Net and Pred-Net separately and exploit the weights of the pretrained convolutional blocks of these networks to train the fully connected classifier. As for the Ind-Net and Pred-Net, we train the model according to a cross-entropy loss function.

3. EXPERIMENTAL EVALUATION

This section describes the experimental setup and the tests that have been carried out to evaluate the robustness of the proposed approach. We begin describing the dataset and configurations used for this work, then, in sections 3.1 and 3.2 we discuss the results that we obtained on several tests.

All the experiments discussed in this section were conducted on a Google Cloud Platform n1-standard-8 instance with 8 vCPUs, 30GB of memory, and an NVIDIA Tesla K80 GPU. The networks have been implemented using Pytorch[20] v.1.6. We trained all the networks with the learning rate set to $1e-4$, weight decay of the L2-regularizer set to $5e-5$, and Adam optimizer with an adaptive learning rate. In our experiments we trained the networks for 80 epochs with batches of size 32 and early stopping set to 10.

To train our model and evaluate its performance, we relied on the VISION dataset [21]. The dataset comprises of 34,427 images and 1,914 videos, both in the native format and in their social media version (i.e., Facebook, YouTube, and WhatsApp), captured by 35 portable devices of 11 major brands. The dataset has been collected recording 648 native single-compressed (SC) videos, mainly registered in landscape mode with *mov* format. For each device, the videos depict flat, indoor, and outdoor scenarios and different acquisition modes. The resolution varies from 640×480 up to 1920×1080 depending on the device. Furthermore, the dataset contains 622 videos that were uploaded on YouTube (YT), and 644 shared through WhatsApp (WA). Similarly to videos, the dataset also contains images captured in multiple orientations and scenarios and shared via Facebook and WhatsApp.

In our experiments, we previously process the dataset with the *ffprobe*[22] analyzer from the *FFmpeg* software to extract the I-frames and P-frames from a subset of 20 devices. Next, we crop each frame into nonoverlapping patches of size $H \times W$, where $H = W = 256$, obtaining 153,843 I-frame patches and 209,916 P-frame patches. Finally, we balance all classes and split the dataset for training, validation, and test with a proportion of 70%, 15%, and 15%, respectively.

3.1. Results on Shared Videos

To estimate the performance of our method, we initially compared the system with respect to a baseline model. Then, we moved forward to assess the performance of our two-stream architecture, namely to validate the increase in performance obtained combing the Ind-Net and Pred-Net.

1) Baseline comparison: In our first set of experiments we measured the performance of the single components of MultiFrame-Net (the Ind-Net and Pred-Net streams) with respect to the baseline model introduced by Nam et al. [15], for their classification efficacy when using only I-Frames and P-

Frames, respectively. To limit model training time, we chose to conduct these experiments on a subset of 10 devices from the VISION dataset. In fact, in this test, we are not interested in obtaining the absolute best performances, but we limit ourselves to proving that there is a boost in performance compared to the baseline. For these experiments we produce an 80%-10%-10% split of the dataset of the input patches for training, validation, and test, respectively.

Input	[15]	Proposed method
I-Frame	67.71%	**88.42%** (Ind-Net)
P-Frame	67.23%	**76.84%** (Pred-Net)

Table 1. Accuracy on a subset of 10 devices from the VISION [21] dataset. The proposed method is confirmed to be more precise than the baseline at recognizing traces left by social networks and apps on frames patches.

The results reported in Table 1 confirm the significantly improved performance of our method respect to the baseline. In fact, the deeper architectures help to distinguish with higher accuracies (88.42% and 76.84% for the Ind-Net and Pred-Net, respectively) between different types of double compressions left by social media and messaging apps. Indeed, the model must be able to distinguish not only between single and double compression, but also between different types of double-compression fingerprints. In this sense, a deeper architecture is capable of extracting more complex information.

2) MultiFrame-Net evaluation: In this test, we evaluate whether and to what extent our two-stream architecture (MultiFrame-Net) improves even more in terms of accuracy compared to the single streams. For this experiment, we trained and evaluated the models on a subset of 20 devices with a dataset split of 70%, 15%, and 15% for training, validation, and test, respectively. First, we train the Ind-Net and Pred-Net in an end-to-end fashion on a subset of 15 devices. Next, applying transfer learning, we froze the convolutional layers of both networks and retrained the fully connected classifier on a subset of 5 devices that have not been used on the previous training. We measure the performance of each network with respect to its accuracy and its area under the curve (AUC) score. Table 2 reports the results of these experiments and Table 3 represents the confusion matrix of MultiFrame-Net. The experiment confirms that by combining the classification of different types of frames, the model achieves better performance, with the MultiFrame-Net gaining up to 95.51% of accuracy and 96.44% of AUC score on patches from SC, WA, and YT. Moreover, the confusion matrix (see Table 3) of the MultiFrame-Net on 3,749 patches from 234 unique videos from WA, YT, and SC suggests that the errors are very small and slightly more numerous in the case of SC patches.

Model	Accuracy	AUC
Ind-Net	92.32%	94.24%
Pred-Net	91.87%	93.12%
MultiFrame-Net	**95.51%**	**96.44%**

Table 2. Model accuracies and AUCs on a subset of 20 devices from the VISION dataset [21]. The MultiFrame-Net shows higher performance with respect to Ind-Net and Pred-Net.

	YT	WA	SC
YT	**1238 (96.41%)**	20(1.65%)	32(2.55%)
WA	31(2.41%)	**1161 (95.79%)**	49(3.91%)
SC	15(1.16%)	31(2.55%)	**1172 (93.53%)**

Table 3. Confusion matrix of the MultiFrame-Net over YT, WA and SC patches from 234 unique videos of the VISION dataset [21].

3.2. Results on Shared Images

In our last experiment, we measure the robustness of the Ind-Net with respect to images. Specifically, we moved from the intuition that I-frames are independently encoded using a process similar to JPEG compression, such that it could be possible to detect images as well as videos coming from the same social media platform. For this reason we test the Ind-Net trained on videos, on native and WhatsApp images available on the VISION dataset. Unfortunately, the VISION dataset contains images uploaded only on WA and Facebook. Therefore, we can apply this test only on WA images. We began the experiment by training the Ind-Net on native and WA video patches obtaining 92.74% of accuracy. Next, by applying transfer learning, we froze the convolutional blocks of the network to act as feature extractors and retrained the fully connected classifier on images from the same classes. With minimal retraining of the classifier, it achieves 86.83% of accuracy. This result suggests that a mixed method to trace both kinds of media is actually possible. Therefore, we leave this problem for future research and extensive experiments.

4. CONCLUSIONS

In this paper, we introduced a CNN architecture to detect videos downloaded from social media and messaging apps, based on their content. We evaluated the advantages of using a deep neural network architecture and inter-modal relationships between features extracted from different types of frames. We also explored the possibility of applying multitask learning to quickly adapt the network from videos to images obtaining promising results. Future work will take into consideration new datasets together with multimodal media assets as well as multitask learning and meta-learning.

5. REFERENCES

[1] "Youtube for press," https://www.youtube.com/about/press/.

[2] "Facebook, company info," https://about.fb.com/company-info/.

[3] Oliver Giudice, Antonino Paratore, Marco Moltisanti, and Sebastiano Battiato, "A classification engine for image ballistics of social data," *Lecture Notes in Computer Science*, p. 625–636, 2017.

[4] Weihong Wang and Hany Farid, "Exposing digital forgeries in video by detecting double mpeg compression," in *Proceedings of the 8th Workshop on Multimedia and Security*, New York, NY, USA, 2006, MM&Sec '06, p. 37–47, Association for Computing Machinery.

[5] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, 2012.

[6] C. Long, E. Smith, A. Basharat, and A. Hoogs, "A c3d-based convolutional neural network for frame dropping detection in a single video shot," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1898–1906.

[7] Roberto Caldelli, Rudy Becarelli, and Irene Amerini, "Image origin classification based on social network provenance," *Trans. Info. For. Sec.*, vol. 12, no. 6, pp. 1299–1308, June 2017.

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

[9] Yijun Quan, Xufeng Lin, and Chang-Tsun Li, "Provenance analysis for instagram photos," in *Data Mining*, Rafiqul Islam, Yun Sing Koh, Yanchang Zhao, Graco Warwick, David Stirling, Chang-Tsun Li, and Zahidul Islam, Eds., Singapore, 2019, pp. 372–383, Springer Singapore.

[10] I. Amerini, C. Li, and R. Caldelli, "Social network identification through image classification with cnn," *IEEE Access*, vol. 7, pp. 35264–35273, 2019.

[11] Q. Phan, G. Boato, R. Caldelli, and I. Amerini, "Tracking multiple image sharing on social networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8266–8270.

[12] M. Iuliani, D. Shullani, M. Fontani, S. Meucci, and A. Piva, "A video forensic framework for the unsupervised analysis of mp4-like file container," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 635–645, 2019.

[13] Emmanuel Kiegaing and Ahmet Emir Dirik, "Prnu-based source device attribution for youtube videos," *Digital Investigation*, vol. 29, 03 2019.

[14] Massimo Iuliani, Marco Fontani, Dasara Shullani, and Alessandro Piva, "Hybrid reference-based video source identification," *Sensors*, vol. 19, pp. 649, 02 2019.

[15] S. Nam, J. Park, D. Kim, I. Yu, T. Kim, and H. Lee, "Two-stream network for detecting double compression of h.264 videos," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 111–115.

[16] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[17] Irene Amerini, Tiberio Uricchio, Lamberto Ballan, and Roberto Caldelli, "Localization of JPEG double compression through multi-domain convolutional neural networks," *CoRR*, vol. abs/1706.01788, 2017.

[18] S. Verde, L. Bondi, P. Bestagini, S. Milani, G. Calvagno, and S. Tubaro, "Video codec forensics based on convolutional neural networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 530–534.

[19] Ghazal Mazaheri, Niluthpol Chowdhury Mithun, Jawadul H. Bappy, and Amit K. Roy-Chowdhury, "A skip connection architecture for localization of image manipulations," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[20] "Pytorch, an open source machine learning framework that accelerates the path from research prototyping to production deployment," https://pytorch.org/.

[21] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Alshaya, and Alessandro Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, pp. 15, 10 2017.

[22] "Ffmpeg, a complete, cross-platform solution to record, convert and stream audio and video," https://ffmpeg.org/ffprobe.html.