ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447089

# STRUCTURE MATTERS: ANALYZING VIDEOS VIA GRAPH NEURAL NETWORKS FOR SOCIAL MEDIA PLATFORM ATTRIBUTION

*Andrea Gemelli, Dasara Shullani, Daniele Baracchi, Simone Marinai, and Alessandro Piva*

Department of Information Engineering, University of Florence

## ABSTRACT

Detecting the origin of a digital video within a social network is a critical task that aids law enforcement and intelligence agencies in identifying the creators of misleading visual content. In this research, we introduce an innovative method for identifying the original social network of a video, even when the video has been altered through actions like group of frames removal and file container reconstruction. The proposed method takes advantage of the video encoding's temporal uniformity, leveraging motion vectors to characterize the specific features associated to various social media platforms. Each video is represented by a graph where nodes correspond to macroblocks. These macroblocks are interconnected by following the inter-prediction rules outlined in the H.264/AVC codec standard. Such a structure can be then classified using a graph neural network to predict the platform on which the video has been shared. Experimental results demonstrate that this approach outperforms both codec- and content-based approaches, underscoring the effectiveness of a structural approach in attributing the social media platform from which videos originated.

***Index Terms*—** Graph Neural Networks, Video Forensics, Social Network Identification, H.264/AVC videos, Motion Vectors.

## 1. INTRODUCTION

Over the last decade, the proliferation of user-friendly editing tools available to a wider audience, combined with the emergence of advanced artificial intelligence techniques capable of creating highly convincing counterfeit images and videos known as *DeepFakes*, has raised significant concerns about the reliability of visual media. The demand for multimedia forensics tools to protect images and videos on social media platforms is growing, with research focusing on practical scenarios like sharing on social networks [1]. Technological challenges, such as platform modifications, complicate traditional forensic methodologies and introduce identifiable patterns. In this scenario, the ability to extract details about a media item's digital history becomes a valuable resource.

This work was supported in part by the Italian Ministry of Universities and Research (MUR) under Grant 2017Z595XS, and in part by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112090136. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

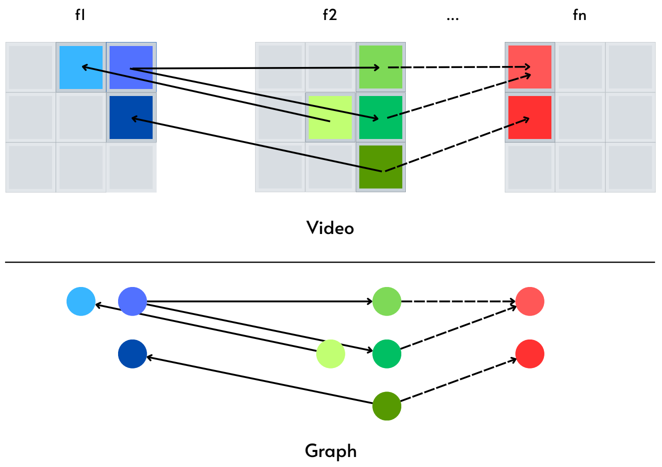

**Fig. 1**. Video to graph representation. Nodes represent macroblocks connected through motion vectors. Dotted lines replace hidden frames; Frames $f2$ and $fn$ may not be physical connected.

This problem has been thoroughly investigated in relation to images. Within the field of forensics, professionals have gradually developed a number of methodologies for estimating the primary social media platform from available evidence. These approaches leverage information derived from quantization tables [2], the container tree [3], and the distribution of Discrete Cosine Transform (DCT) coefficients [4], sometimes in conjunction with characteristics obtained from convolutional neural networks [5, 6].

Recently, there has been a growing interest for broadening the scope of social platform identification to include video content as well. A highly effective approach involves analyzing the video container tree [7], which has proven to be extremely accurate in characterizing the final processing steps applied to a video. Despite its remarkable effectiveness, the video container tree remains a fragile trace that can be altered even through basic operations like stream copying without additional compression. Hence, even unintentional actions have the potential to lead a classifier astray. To address these limitations, more robust features such as the statistics of DCT coefficients can be utilized to classify videos shared on popular social platforms [8]. This technique, while not as effective as container-based methods for unaltered videos, exhibits remarkable resilience against laundering operations, unlike the rapid decline of container-based methods.

Lately, there has been quite an increase in interest in applying innovative deep learning approaches to forensic applications. Models that combine information from both images and videos have demonstrated their effectiveness in classifying the platforms through which videos are shared [9]. Among emerging architectures in this field, Graph Neural Networks (GNNs) have gained significant attention. Even though GNNs have made recent breakthroughs, it is worth noting that they have a long-standing history. The concept of GNNs was introduced, among others, in [10], where the authors proposed a novel architecture that harnessed graph structures for ranking web pages. Over time, GNNs have evolved and diversified, leading to the coining of the *umbrella* term "Geometric Deep Learning" to encompass various graph-based approaches [11]. The versatility of GNNs is evident from their wide range of applications [12], spanning from enhancing connectivity in social networks like Snapchat [13] to detecting fake news on platforms like Twitter [14]. In the realm of video forensics, GNNs have found utility in tasks such as detecting partial copies of videos [15], identifying visual anomalies in surveillance videos [16], and detecting deepfakes videos [17]. An interesting application of GNNs consists of classifying the container tree of videos [18]. This approach has yielded promising results in various tasks, including attributing social network and editing tools to videos. However, it is worth noting that this method relies on the same fragile features (i.e. container tree) introduced in [7], making it susceptible to manipulation if the container structure is tampered with.

In this paper, we propose a novel approach employing a GNN to analyze videos modeled as graph structures, with the primary objective of classifying the social network from which videos originate. The main contributions of this paper are: (i) we devise an algorithm to build a graph representation of a video exploiting the structure induced by the H.264/AVC codec, as depicted in Fig. 1; (ii) we adopt a GNN to classify the graphs; (iii) we evaluate the proposed approach by comparing with state-of-the-art techniques and by means of suitable ablation studies.

The paper is structured as follows. In Section 2, we describe in detail the proposed method for the graph-based representation of videos and the GNN architecture used for the experiments. In Section 3, we report the ablation studies, the results achieved by the GNN, and its limitations. Finally, in Section 4 we draw the conclusions and possible future extensions of this work.

## 2. METHOD

Our approach comprises two fundamental phases: firstly, the construction of a graph representation based on features extracted from the video bitstream, and secondly, the classification of these features using a deep-learning-based method. Specifically, we employ macroblocks (MBs) as fundamental elements in the graph representation; the MBs correspond to nodes, carrying information like position, type, and partition. Edges in the graph correspond to motion vectors. Subsequently, we leverage a graph neural network to acquire a deeper understanding of this graph representation and ultimately classify the video according to its social network of origin. In the following, we delve into a detailed and comprehensive description of these two steps.

### 2.1. H.264/AVC codec and video features

The predominant video encoding and decoding algorithm used in smartphones is the H.264/AVC [19]. This codec was specifically designed to reduce the storage requirements for high-resolution videos. In pursuit of this goal, each video frame is divided into fixed-size macroblocks: $16\times16$ pixels for the luma component and $8\times8$ pixels for the chroma components. Luma macroblocks can be further divided into partitions (or splits) of various sizes, such as $16\times8$, $8\times16$, $8\times8$, or $4\times4$ pixels. To optimize the information needed for encoding video content, each macroblock is stored as a difference (or residual) in relation to a predicted content derived from another macroblock. This residual undergoes a series of transformations, including Discrete Cosine Transformation, quantization, and entropy coding, before reaching its final form. The above encoding process for each macroblock varies based on its prediction mode. Macroblocks of type I store the residual concerning intra-prediction, based on their neighboring pixels in the same frame. On the other hand, macroblocks of type P are predicted by selecting the most similar macroblock that has already been decoded in a previous frame. Furthermore, macroblocks of type B employ motion compensation predictions, taking into account macroblocks from multiple previously decoded frames. In all instances where a macroblock needs to reference another one for prediction (P and B), the encoder stores a motion vector that points to this referenced macroblock.

The H.264 standard grants broad discretionary powers to implementations regarding how to divide frames and macroblocks into various types of predictions. Consequently, it is reasonable to expect that different encoders may generate distinct prediction patterns, influenced by specific choices made by the implementer. This implies that such variations could be leveraged as indicators of the social platform from which a video originates, much like previous research has demonstrated with video file containers [7] and the residuals stored in macroblocks within I-frames [8]. To harness this informa-

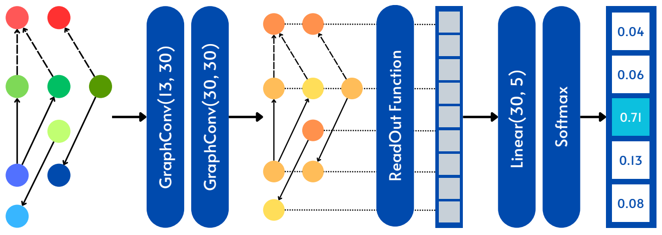

**Fig. 2**. Each graph representing a video passes through a 2-layer GNN, using *GraphConv* as hidden layer and updating hidden nodes' representations. Node features are aggregated with a max *ReadOut* function and the resulting vector representing the whole graph is fed into a *Linear* classifier layer. Hidden layer dimension is set to 30 through hyperparameter tuning.

tion, we build a graph from each video stream representing the structure induced by the predictions. In this graph, nodes represent macroblocks, and edges are formed by motion vectors connecting MBs[1]. It is possible to follow the motion vector of a specific MB to an area of the frame that is not aligned with the macroblock grid. In such cases, all MBs in the destination frame that intersect this area are connected by an edge to the macroblock of origin. Each node is labeled with its normalized XY coordinates, prediction type (e.g., I, B, P), and split size. Finally, we exclude all nodes that are not connected by any edge (i.e., that are neither the origin nor the destination of a motion vector). We highlight that this representation has a much looser dependency on video content with respect to other features such as the DCT coefficients distribution, as most information is given by the structural choices performed by the encoder. The process of representing a video as a graph is illustrated in Fig. 1.

## 2.2. Graph Neural Network

Graphs can be processed by GNNs at node, edge, and graph level. We are interested in the latter case and, in particular, in a graph classification task. Each node in the graph corresponds to a frame macroblock, carrying a feature vector containing: position features (x,y coordinates) scaled within 0 and 1 given the video resolution and two different one-hot encoding vectors used separately for type (I, P, B, SI, Skip, Other) and split features ($[16 \times 16]$, $[16 \times 8]$, $[8 \times 16]$, $[8 \times 8]$, $[4 \times 4]$), concatenated in a 13-long features vector. Then, the motion vectors establish the connections and hence guide the exchange of information within neighbors using a permutation invariant aggregation function. Finally, to classify the entire graph, all the collected and learned hidden node representations are aggregated through a *ReadOut* function and used to classify across the different social networks. In details, as depicted in Fig. 2, we employed a two-layer deep GNN updating node features given neighboring information (using the implementation of *GraphConv* [21] from the PyTorch Deep Graph Library as the core layer), a *ReadOut* max function to aggregate all nodes information into one vector representing the whole graph, a final *Linear* layer, and *Softmax* function activation. Each node vector, through the *GraphConv* layers, is updated as follows:

$$h_i^{(l+1)} = \sigma\left(b^{(l)} + \sum_{j \in N_i} \frac{1}{c_{ji}} h_j^{(l)} W^{(l)}\right) \tag{1}$$

where $h_i^{(l+1)}$ refers to the feature vector of the $i^{th}$ node updated after the $l^{th}$ layer, $h_j$ are its neighbor ones (belonging to the set $N_i$) and $c_{ji}$ is the product of the square root of node degrees. $W^{(l)}$ and $b^{(l)}$ are the shared matrix of weights and the bias term, respectively, at layer $l$, and $\sigma$ is a ReLU activation function.

## 3. EXPERIMENTAL VALIDATION

In this section, we describe the dataset used for our evaluation. Then, we outline the experimental setup employed for conducting the ablation study and for comparing our approach with other state-of-the-art methods. Finally, we provide an analysis and discussion of the results obtained.

### 3.1. Dataset and settings

In order to have an appropriate comparison with existing approaches, we adopted the same dataset and experimental scenarios outlined in [8]. The dataset (Premier-A4) contains native videos collected from 28 different smartphones and shared through Facebook, Instagram, Twitter, and YouTube. The dataset also contains versions of both native and shared videos laundered via *FFmpeg* and *Avidemux* editing softwares. Videos edited with *FFmpeg* have been built by extracting the original video and audio streams and then by combining them in a new container, while videos edited with *Avidemux* were created by removing the first group of pictures from the original video, copying the audio stream, and reconstructing the video container. The final collection resulted in 386 videos in the Base scenario, 383 in FFmpeg, and 267 videos in Avidemux [2]. In the first evaluation scenario (Base), the classifier is trained and tested on two different subsets of non-edited videos. In the second (FFmpeg) and third (Avidemux) scenarios, the classifier is trained on the same set of videos from the Base scenario, but tested on *FFmpeg*- or *Avidemux*-edited versions of test videos, respectively. All the experiments are performed in a leave-one-device-out cross validation setting, where the model is trained and evaluated

[1]An extended version of JM 16.1 [20] has been used to collect also inter-predicted information for each macroblock.

[2]Differences in the total number of videos compared to Shullani et al. [8] are related to unprocessed feature extraction.

**Table 1**. Graphs' statistics in all scenarios.

| **Scenario** | **Graphs** | **Nodes** | | | **Edges** | | |
|---|---|---|---|---|---|---|---|
| | total | min | median | max | min | median | max |
| Base | 386 | 5.5K | 160K | 1.3M | 5.5K | 164K | 1.3M |
| FFmpeg | 383 | 15K | 160K | 1.3M | 15K | 165K | 1.3M |
| Avidemux | 267 | 1.9K | 228K | 1.2M | 2K | 236K | 1.2M |

28 times, each using videos from a different device for testing and videos from all the other devices for training. In Table 1 we report the statistics of videos transformed into graphs to illustrate the dimensions we have been working with. Experiments have been carried out on an Intel(R) Core(TM) i9-7940X CPU at 3.10GHz with 125 GB of CPU RAM and a Quadro P6000 equipped with 24 GB of GPU RAM. The source code of the algorithm and additional experimental results are available online [3].

## 3.2. Experiments and results

We designed our experiments with the aim of addressing several key research questions: (i) whether the structure truly matters, (ii) to what extent do node features affect the ultimate outcomes, and (iii) if performance improvements scale with the quantity of frames. To do so, we examine the impact of using random connections instead of motion vectors, varying the number of frames utilized and selecting different subsets of node features. As for the frame count, we adopted two settings: a restricted one involving only the first 10 frames from each video and a standard one using the initial 112 frames, as this quantity represents the minimum number of frames common to all Premier-A4 videos. This selection was made to ensure an equitable comparison among the videos while taking into account computational constraints. In terms of feature selection, we explored various combinations of normalized MB coordinates (XY), MB type (Type), and MB partitioning (Split). We report the outcomes of these ablations in Table 2.

To begin, we emphasize the significance of employing motion vectors as edges. Notably, we observe a consistent decline in performance across all experiments where random connections are used instead. In the base scenario, exploiting motion vectors, we achieve a balanced accuracy [22] of 0.89. This surpasses the results obtained from random connections and represents an average improvement of 21 points. We also observe that the combination of all the considered features is informative, even in the absence of motion vectors. However, when motion vectors are employed to construct edges, there is a remarkable 20-point enhancement, underscoring the validity of our assertion that structure matters.

As for the second question, the ablation experiments reveal that the most favorable results are consistently achieved through the combined utilization of all the considered features in all tested scenarios. It's worth noting that the use of just the XY feature with MVs can yield relatively strong performance, especially in the context of the 112-frame setting.

[3]`https://github.com/IAPP-Group/StructureMatters`

**Table 2**. Ablation study of graphs' structure in terms of balanced accuracy [22]. Base scenario.

| **Features** | | | **10 frames** | | **112 frames** | |
|---|---|---|---|---|---|---|
| XY | Type | Split | Random | MVs | Random | MVs |
| ✓ | | | 0.20 | 0.42 | 0.20 | 0.55 |
| ✓ | ✓ | | 0.40 | 0.54 | 0.40 | 0.70 |
| ✓ | | ✓ | 0.51 | 0.67 | 0.44 | 0.78 |
| ✓ | ✓ | ✓ | *0.64* | **0.72** | *0.69* | **0.89** |

**Table 3**. Balanced accuracy [22] results. The proposed GNN uses 112 frames per video and MVs as node connections.

| **Method** | **Base** | **FFmpeg** | **Avidemux** |
|---|---|---|---|
| Yang et al. [7] | **0.99** | 0.24 | 0.27 |
| Shullani et al. [8] | 0.74 | 0.72 | 0.67 |
| Proposed GNN | *0.89* | **0.87** | **0.81** |

To address the third and final question, we experimented with a subset of frames, and we observed an overall decrease in performance. Interestingly, increasing the number of frames does not significantly improve the results when random connections are employed. In contrast, when motion vectors are utilized as edges, an increase in the number of frames consistently yields better outcomes.

After establishing the efficacy of our structural approach, we compared it with two other state-of-the-art methods, as detailed in Table 3. The proposed method exhibits a substantial improvement of nearly 15 points over the results reported in [8] across all scenarios. Significantly, not only do we narrow the performance gap in the base case by 10 points (as opposed to the previous 25-point margin), but our method also demonstrates robustness in the other two scenarios, maintaining strong performance as indicated by balanced accuracy.

## 4. CONCLUSIONS

In this paper, we introduce a novel approach for video social media classification based on a graph representation of video macroblocks processed by a Graph Neural Network. We demonstrate the effectiveness of this approach, improving the accuracy with respect to state-of-the-art methods and maintaining strong performance even on laundered videos. Future research directions include investigating the aggregation of larger portions of video frames to include a wider range of information and meet computational memory and time constraints, as well as combining content and structural features to improve results.

## 5. REFERENCES

[1] Cecilia Pasquini, Irene Amerini, and Giulia Boato, "Media forensics on social media platforms: a survey," *EURASIP Journal on Information Security*, vol. 2021, no. 1, pp. 1–19, 2021.

[2] Oliver Giudice, Antonino Paratore, Marco Moltisanti, and Sebastiano Battiato, "A classification engine for image ballistics of social data," in *19th International Conference on Image Analysis and Processing*. Springer, 2017, pp. 625–636.

[3] Sebastiano Verde, Cecilia Pasquini, Federica Lago, Alessandro Goller, Francesco De Natale, Alessandro Piva, and Giulia Boato, "Multi-clue reconstruction of sharing chains for social media images," *IEEE Transactions on Multimedia*, 2023.

[4] Irene Amerini, Tiberio Uricchio, and Roberto Caldelli, "Tracing images back to their social network of origin: A CNN-based approach," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 2017, pp. 1–6.

[5] Irene Amerini, Chang-Tsun Li, and Roberto Caldelli, "Social network identification through image classification with CNN," *IEEE access*, vol. 7, pp. 35264–35273, 2019.

[6] Simone Magistri, Daniele Baracchi, Dasara Shullani, Andrew D Bagdanov, and Alessandro Piva, "Towards continual social network identification," in *2023 11th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2023, pp. 1–6.

[7] Pengpeng Yang, Daniele Baracchi, Massimo Iuliani, Dasara Shullani, Rongrong Ni, Yao Zhao, and Alessandro Piva, "Efficient video integrity analysis through container characterization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 947–954, 2020.

[8] Dasara Shullani, Daniele Baracchi, Massimo Iuliani, and Alessandro Piva, "Social network identification of laundered videos based on DCT coefficient analysis," *IEEE Signal Processing Letters*, vol. 29, pp. 1112–1116, 2022.

[9] Luca Maiano, Irene Amerini, Lorenzo Ricciardi Celsi, and Aris Anagnostopoulos, "Identification of social-media platform of videos through the use of shared features," *Journal of Imaging*, vol. 7, no. 8, pp. 140, 2021.

[10] F. Scarselli, Sweah Liang Yong, M. Gori, M. Hagenbuchner, Ah Chung Tsoi, and M. Maggini, "Graph neural networks for ranking web pages," in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 2005, pp. 666–672.

[11] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[12] "Everything is connected: Graph neural networks," *Current Opinion in Structural Biology*, vol. 79, pp. 102538, 2023.

[13] Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah, "Graph neural networks for friend ranking in large-scale social platforms," in *Proceedings of the Web Conference 2021*, 2021, pp. 2535–2546.

[14] Yi Han, Shanika Karunasekera, and Christopher Leckie, "Graph neural networks with continual learning for fake news detection from social media," *arXiv preprint arXiv:2007.03316*, 2020.

[15] Xiyue Liu, Xin Feng, and Pan Pan, "GANN: a graph alignment neural network for video partial copy detection," in *7th IEEE Intl. Conference on Big Data Security on Cloud, High Performance and Smart Computing, and Intelligent Data and Security*. IEEE, 2021, pp. 191–196.

[16] Kai Cheng, Yang Liu, and Xinhua Zeng, "Learning graph enhanced spatial-temporal coherence for video anomaly detection," *IEEE Signal Processing Letters*, vol. 30, pp. 314–318, 2023.

[17] Fatima Khalid, Ali Javed, Hafsa Ilyas, Aun Irtaza, et al., "DFGNN: an interpretable and generalized graph neural network for deepfakes detection," *Expert Systems with Applications*, vol. 222, pp. 119843, 2023.

[18] Ziyue Xiang, Amit Kumar Singh Yadav, Paolo Bestagini, Stefano Tubaro, and Edward J Delp, "MTN: forensic analysis of MP4 video files using graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 963–972.

[19] Iain E Richardson, *The H. 264 advanced video compression standard*, John Wiley & Sons, 2011.

[20] Video Quality Experts Group, "Modified jm h.264/avc codec," 2018.

[21] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[22] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 3121–3124.