Tracing Videos to their Social Network with Robust DCT Analysis

Ben Lewis (Ben.Lewis@warwick.ac.uk) and Victor Sanchez (V.F.Sanchez-Silva@warwick.ac.uk)

Department of Computer Science, University of Warwick, Coventry, U.K.

Abstract. Videos are increasingly becoming a prominent form of multimedia information readily available online, primarily through social networks. However, misinformation can easily be spread through videos and has the potential to go viral, with severe social consequences. Being able to identify the source of a video can add authenticity to it and help detect and track misinformation. Although different approaches have been proposed to identify the social network used to share a video, each one has flaws such as being dependent on the spatial domain of the video or vulnerable to laundering. One of the most robust techniques is based on the detection of the unique traces left by the compression process applied by the social network by analysing the DCT coefficients of a compressed video. Different social networks compress a video differently, such as using different coding parameters, leading to distinct differences in the DCT coefficients. This work improves upon previous DCT coefficient-based methods by using a novel feature vector created with the interval histograms of the DCT coefficients of all the color components of the I- and P-frames. By training a random forest classifier with this feature vector, significant improvements are achieved, even when videos are shared multiple times or edited.

Keywords: Video forensics · Platform provenance · Machine learning.

1 Introduction

Every day thousands of hours of videos are uploaded to social networks, with 30000 hours of videos being uploaded to YouTube alone every hour [1]. With this vast amount of multimedia content readily available online, automatic methods must be developed to help detect misinformation and prevent its spread to a mass audience. Reconstructing the sharing history of a video can help identify the video's source and verify its authenticity. This sharing history is commonly known as the platform provenance of the video, which has been identified as important in the forensics community [2].

There has been significant research on platform provenance for images with multiple approaches being proposed. Some rely on using the metadata within a classification framework [3], while others train machine learning models to detect artefacts introduced by the social network in the image's sensor pattern noise [4].

One of the most common approaches is to identify the compression performed by the social network, which is similar to detecting double JPEG compression [5]. Image and video compression relies on transforming the data using a transform (e.g., for JPEG, it is the Discrete Cosine Transform (DCT)) that allows for the transformed data to be compressed. It is also common practice to predict the raw data first and transform only the prediction error, an approach called predictive transform coding (PTC). Modelling the DCT coefficients of compressed images has been used to detect compression traces of that are unique to a social network [6]. Some of the techniques designed for images have been extended to videos, with container (metadata) based approaches achieving good performance [7, 8]. However, the approaches designed for videos often have undesirable properties, e.g., metadata-based approaches are vulnerable to laundering [9].

A recent state-of-the-art approach extends the modelling of DCT coefficients to videos [9]. Such modelling is desirable because it accounts for the compression processes applied by the social network and relies on the content of the video without using the raw pixel values, which may cause the model to not generalize well to unseen videos. Different social networks apply different compression processes (e.g., by using different coding parameters [9]), which leave distinct and detectable traces in the DCT coefficients.

This paper introduces several improvements to the current DCT coefficient analysis for platform provenance of videos. Specifically, we design a strong feature vector based on the interval histogram of the DCT coefficients of not only key frames of a compressed video (i.e., I-frames), but also of those frames compressed by motion estimation and compensation (P-frames). The improved feature vectors are used to train a random forest (RF) classifier to detect the video's social network as a classification task. Our results on the PREMIER A1 and A2 datasets show that the RF trained on the proposed feature vectors outperforms the state-of-the-art for the single sharing case, i.e., a video is shared on one social network, and the multiple sharing case. i.e., the video is shared on multiple social networks. Our results also show that our model achieves strong performance on the single sharing case when the videos are edited to modify their visual content. The performance on edited videos is promising because it suggests that our model is robust to changes by social networks in their uploading processes, i.e., by changing the coding parameters.

2 Related Work

The task of platform provenance has been first considered for images [2] for both the single sharing case [10] and the multiple sharing case. However, there is significantly less research on platform provenance for videos [2]. In general, there are four main approaches in the literature to tackle the case of identifying the social network of a video for the single sharing case.

The first approach relies on metadata, where the container [7] or the encapsulation characteristics [8] of the video are analysed to train a classifier to identify the social network. Approaches using metadata can achieve very impressive re-

sults; however, it has been shown that if the metadata is tampered with, their accuracy can drastically decrease [9]. While some approaches [8] try to limit the effect of tampering by using encoding parameters that would require re-encoding to change the metadata, it is more desirable for the classifier to just rely on the visual content to increase robustness.

The second approach [11] involves training a model to learn to distinguish the unique traces introduced by the compression applied by the social network. This is done by analysing the I-frames and P-frames. However, if the pixels of the decompressed frames are used to distinguish these traces, the classifier may not generalize well to other videos with very different visual content. Therefore, it is desirable to detect a video's social network without relying on the pixel values.

The third approach uses transfer learning to re-train a model designed for images to one that can work with videos [12]. Since the signal processing applied to images and videos by a social network is likely to be different, it is desirable to design and train a classifier that targets videos, rather than relying on the similarities between the image and video signal processing.

The last approach [9] avoids many of the problems highlighted before by using the DCT coefficients of the I-frames to identify the traces left by the compression applied by the social network. This is a desirable approach because a classifier trained on information extracted from the DCT coefficients relies on the content of the video without using the pixel values. This makes the trained model robust to unseen videos and editing applied to the videos. This paper follows this approach and focuses on the H.264 codec, which is one of the most used video codecs [9, 13, 14].

3 Proposed Feature Vectors

It has been shown that analyzing the DCT coefficients of compressed images within a classification framework can be used to detect the sharing platform [6, 10]. The same idea can be applied to videos [9]. Although this work focuses on the H.264 video codec, the ideas presented here can be easily adapted to any codec that uses PTC. In general, video codecs based on PTC process a video as a set of groups of pictures (GOPs) and encode each GOP independently. Each frame in a GOP is encoded as either an I-, P-, or a B-frame, with a different level of compression achieved by each coding type. Regardless of the type, each frame is split into blocks and each block is compressed independently. Specifically, I-frames are compressed using intra-frame coding, where blocks are first sequentially predicted using other blocks already predicted and compressed within the same frame. The prediction errors, i.e., the difference between the original block and the predicted one, are then transformed using the DCT, and the resulting coefficients are finally entropy-encoded. P-frames are compressed using inter-frame coding, where blocks are sequentially predicted first using another block in any previous P-frame or the I-frame within the same GOP, a process called motion compensation and estimation. The prediction errors are also transformed using

the DCT and the resulting coefficients are finally entropy-encoded. B-frames are compressed similarly to P-frames, however, the block that is chosen for the prediction can either come from a previous or future P-frame or the I-frame within the same GOP.

Differently from other works that consider exclusively the DCT coefficients of the I-frames, we consider the DCT coefficients of the I- and P-frames. We do not use the B-frames as their DCT coefficients are mostly zero-valued and hence do not provide discriminative information. By considering I- and P-frames, one can train a classifier that can generalize better to unseen videos. Moreover, we use the luma component and both chrominance components because these components are often handled differently by the video codec. By including all three components a higher classification accuracy can be attained.

Although applying the DCT on the decompressed frames allows obtaining DCT coefficients, these coefficients may not include the traces left by the compression process applied by the social network as they contain the distortions introduced by the extra decompression process. We then opt for extracting the DCT coefficients directly from the compressed bitstream, as done by other methods [9]. To this end, we modify the open source JM H.264/AVC Codec [15] to output the histograms of DCT coefficients for each color component of I- and P-frames, i.e., six different histograms .

The bin of the histograms that represents the zero-valued coefficient is discarded because these coefficients provide negligible information about the compression process used by the social network. Six different interval histograms are then computed by using intervals of size 4. Equation 1 shows how the frequency for value v, denoted by $I_H(v)$, is calculated based on the frequency $H(v)$ for value v in the DCT histogram:

$$I_H(v) = H(v) + H(v+1) + H(v+2) + H(v+3). \tag{1}$$

It is important to note that other methods [9] use *disjoint* histograms, where the frequencies of the DCT coefficients not seen within the training set are skipped. However, unseen videos may contain DCT coefficients that are not seen in the training set. Hence, not accounting for these values can potentially discard critical information. Therefore, our interval histograms can improve the model's ability to generalize to unseen data by extrapolating the frequency of DCT coefficient values not present in the training set by using neighbouring DCT coefficients. Interval-based histograms have been used before in forensics, with quantized histograms being used for spatially rich models of digital images to make the features more sensitive to changes [16]. Figure 1 shows the two different representations for the same histogram in the interval [20, 60]. This figure shows that the disjoint histogram has several bins missing due to those coefficient values not appearing in the training set. The interval histogram, on the other hand, consistently has bins with a frequency value greater than 0 for every 4 coefficient values.

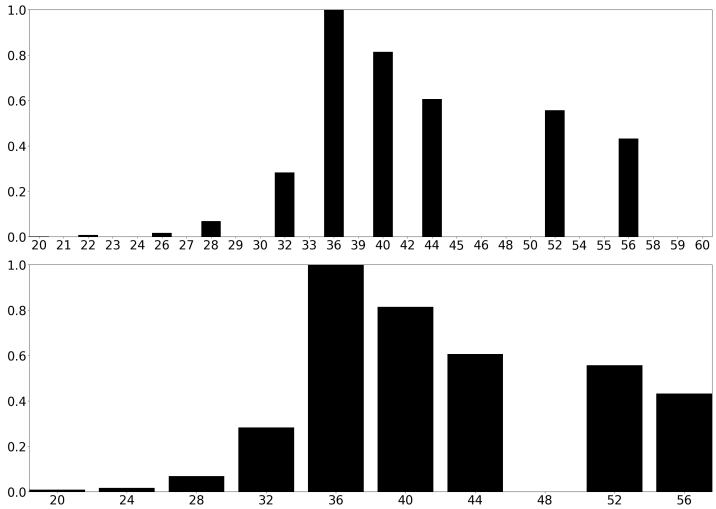

Fig. 1: The disjoint (top) and interval (bottom) histogram for the same video in the interval [20, 60].

By using the six different interval histograms representing one of the components of the I-or P-frames (each with 2,500 bins), we create a feature vector by concatenating them into one final representation (with 15,000 bins). This feature vector is normalized to account for different video lengths, i.e., short videos are expected to have lower frequency values while long videos are expected to have higher frequency values. As a classifier, we use an RF because its bagging process provides strong generalization capabilities and reduces overfitting. This classifier is particularly useful for small training sets, e.g., the PREMIER A2 dataset [9] which comprises only 160 videos, as bagging uses different subsets of the training data. Additionally, each decision tree within the RF can learn from different parts of the feature vector, i.e., from different components and frame types, thus making independent errors that make the whole ensemble an effective classifier.

4 Experimental Results

We compare the performance of the RF trained with our proposed feature vectors against the method in [9], which is referred to as the baseline method. We used the PREMIER A1 and PREMIER A2 datasets for these tests. The "Facebook", "YouTube", "Weibo", and "Tik Tok" classes are chosen from the PREMIER A1 dataset because these social networks are diverse enough for the results to be significant. The PREMIER A2 dataset is used for the multiple sharing case with each class representing the chain of social networks the video has been shared through.

The baseline method constructs a histogram of the frequency of the values of the DCT coefficients of the I-frames. Next, it removes the bins that represent the coefficients that do not appear in the training set, thus creating a disjoint histogram. The bins that are required to construct the disjoint histogram are then stored so that when testing the constructed disjoint histograms use the same bins as those used for training.

Two modifications are introduced to the feature vectors used by the baseline to have a fair comparison against our RF. Namely, the DCT coefficients extracted from the bit-streams are limited to values in the range $[-5000, 5000]$. This means that coefficients outside of this region are not used. However, this is not expected to significantly affect the results as very few coefficients are outside this range. Additionally, the feature vectors are normalized to have a magnitude of 1.

The available videos are randomly split into a training and test set with a 90:10 ratio. Each test is repeated 100 times with different splits to emphasize the statistical significance of the results, with every test using the same 100 splits. Results are presented as confusion matrices averaged over the 100 tests. Empirically, we find that using 200 different decision trees in the RF and either a linear or RBF kernel for the SVC in [9] provides the best results.

Single sharing case on unedited videos: For this test, only the unedited videos of the PREMIER A1 dataset are used, i.e., each class has 140 videos, for a total of 560 videos. Table 1 shows the performance of the baseline and our RF. These results show that the RF trained with the proposed feature vectors achieves a higher accuracy for each of the classes, demonstrating that it can effectively learn the traces left by the compression process used by each social network more effectively than the baseline. The overall accuracy of our RF is 98.59% vs. 83.54% for the baseline.

Note that for the "Weibo" class, our RF always classifies the videos correctly, which shows that the traces left behind by this platform are very distinct. These results are promising because the proposed method can achieve a very high accuracy while using only the DCT coefficients. This is desirable due to DCT analysis having many useful properties (as highlighted earlier) compared to other approaches.

Table 1: The mean confusion matrix of the proposed and baseline model on unedited videos from the PREMIER A1 dataset.

SN vs	**Facebook**		**Weibo**		**YouTube**		**Tik Tok**	
Class. (%)	Prop.	Base.	Prop.	Base.	Prop.	Base.	Prop.	Base.
Facebook	99.71	97.64	0.0	0.0	0.29	1.42	0.0	0.93
Weibo	0.0	2.5	100.0	87.07	0	8.93	0	1.5
YouTube	0.0	12.57	0.0	0.29	98.71	83.21	1.29	3.92
Tik Tok	2.43	10.86	0.0	0.79	1.64	22.14	95.92	66.21

Single sharing case on edited videos: In this set of experiments, three different editing techniques from the PREMIER A1 dataset are considered:

1. **Cut down:** the video is cut to 5-7 seconds. A cut out video may be seen out of context, thus manipulating its original narrative.
2. **Speed up:** the video is sped up by 4 times by dropping some frames. This kind of edit may seem trivial, but there have been real-world scenarios of it being used as pointed out in [7].
3. **Cut down and downscale:** the video is cut down to 15 seconds and downscaled to a 320×420 resolution.

Each video is tampered with before being uploaded to the social network. The untampered videos are used for training, while tampered ones are used for testing. Each class then has 140 untampered videos and each of those videos has a tampered version for each type of edit. By training on untampered videos and testing on tampered videos, the experiment also tests the robustness of these classifiers to any significant changes in the uploading process that a social network may perform in the real world. While these edits may be more extreme than any real-life change, these tests still demonstrate the classifier's robustness to content edits. Note that DCT analysis is robust to changes to the container as the container is not used.

Table 2 demonstrates that our RF is more robust to Edit 1, with an overall accuracy of 77.44% compared to the baseline's accuracy of 65.09%. The results show that editing videos can significantly affect the distribution of the DCT coefficients, affecting the accuracy of DCT-based methods. Note that the "YouTube" class is classified poorly by the baseline, i.e., 41.93% accuracy vs. the 83.21% accuracy achieved in the previous set of experiments. This suggests that the distribution of DCT coefficients from different social networks is affected differently by the different edits.

Table 2: The mean confusion matrix of the proposed and baseline model on videos from the PREMIER A1 dataset tampered using Edit 1.

SN vs	**Facebook**		**Weibo**		**YouTube**		**Tik Tok**	
Class. (%)	Prop.	Base.	Prop.	Base.	Prop.	Base.	Prop.	Base.
Facebook	95.79	95.29	0.0	0.0	2.71	2.71	1.5	2.0
Weibo	0.0	0.0	88.64	83.71	0.0	7.0	11.36	9.29
YouTube	21.36	56.43	0.0	0.0	75.5	41.93	3.14	1.64
Tik Tok	14.57	25.14	0.0	9.07	33.57	26.36	49.86	39.43

Table 3 shows the performance after using Edit 2. Our RF is more robust to this type of edit than the baseline with an overall accuracy of 77.98% vs. 54.07% for the baseline. The overall accuracy of the RF for the case of using Edit 2 is similar to the case of using Edit 1. This further shows that the RF is robust to a larger range of edits.

Table 3: The mean confusion matrix of the proposed and baseline model on videos from the PREMIER A1 dataset tampered using Edit 2.

SN vs	**Facebook**		**Weibo**		**YouTube**		**Tik Tok**	
Class. (%)	Prop.	Base.	Prop.	Base.	Prop.	Base.	Prop.	Base.
Facebook	88.21	84.64	0.0	0.0	0.93	1.07	10.86	14.29
Weibo	5.5	10.21	75.79	42.14	1.64	15.07	17.07	32.57
YouTube	4.71	49.21	0.0	0.0	89.07	43.71	6.21	7.07
Tik Tok	18.0	27.21	0.0	2.21	23.14	24.79	58.86	45.79

Finally, Table 4 shows the results for the case using Edit 3. Note that while our RF does outperform the baseline, it has a significantly lower accuracy compared to the other types of edits. Edit 3 is more aggressive as it is a combination of two edits, where the downscaling significantly degrades the visual quality of the videos. The overall accuracy of our RF is 66.2% compared to the baseline's accuracy of 62.66%. Also, note that both methods can classify two of the social networks with high accuracy, but poorly classify the other two. This shows that this edit can be used to intentionally fool these classifiers; however, this edit would impair the visual quality of the videos significantly.

Table 4: The mean confusion matrix of the proposed and baseline model on videos from the PREMIER A1 dataset tampered using Edit 3.

SN vs	**Facebook**		**Weibo**		**YouTube**		**Tik Tok**	
Class. (%)	Prop.	Base.	Prop.	Base.	Prop.	Base.	Prop.	Base.
Facebook	97.57	87.14	0.0	0.0	2.43	12.71	0.0	0.14
Weibo	0.0	0.0	98.86	95.93	0.0	0.0	1.14	4.07
YouTube	54.64	67.79	0.0	0.0	41.14	31.79	4.21	0.43
Tik Tok	43.0	40.86	0.0	5.07	29.79	18.26	27.21	35.79

Multiple sharing case: Our RF and the SVC in [9] are trained on the PREMIER A2 dataset, which has 4 classes each with 40 videos. Each class represents a different sharing chain between Facebook and YouTube, with the name of the class representing the sharing order. Table 5 shows that our RF outperforms the baseline in every class, with an overall accuracy of 84.38% compared to the baseline's accuracy of 58.88%. Compared to the experiments on PREMIER A1 for the single sharing case - unedited videos (see Table 1), the accuracy of the baseline significantly decreases demonstrating that this is a challenging task. Videos that have been shared on multiple social networks have compression traces of all networks, therefore our more diverse and stronger feature vectors allow for the RF to learn these multiple traces more effectively. As mentioned in Section 2, research on platform provenance for videos is still in its infancy [2]. This is

particularly true for the multiple sharing case, therefore the results of this set of experiments are very promising.

Table 5: The mean confusion matrix of the proposed and baseline model on videos from the PREMIER A2 dataset.

SN vs	**FB**		**FB-YT**		**YT**		**YT-FB**	
Class. (%)	Prop.	Base.	Prop.	Base.	Prop.	Base.	Prop.	Base.
FB	82.0	65.5	0.0	11.75	0.75	0.5	17.25	22.25
FB-YT	0.0	6.25	88.25	77.0	11.75	13.25	0.0	3.5
YT	0.5	6.5	13.75	41.75	85.5	47.25	0.25	4.5
YT-FB	16.5	40.0	1.0	9.0	0.75	5.25	81.75	45.75

Ablation studies: To confirm the advantages of using interval histograms to create the feature vectors, we evaluate the performance of our RF with disjoint histograms (as used by [9]) for the I- and P-frames and all the color components. Table 6 tabulates the overall accuracy of our RF averaged over 100 tests. These results confirm that the use of interval histograms improved performance for every case. Within the supplementary material, an ablation study is included justifying the design decisions made and highlighting where the improvement in performance over the baseline comes from.

Table 6: Average accuracy of the RF classifier when using disjoint and interval histograms to compute the feature vectors.

Dataset	**Disjoint**	**Interval**
Unedited PREMIER A1	98.44	98.59
PREMIER A1 with Edit 1	76.69	77.45
PREMIER A1 with Edit 2	77.48	77.98
PREMIER A1 with Edit 3	66.11	66.2
PREMIER A2	84.13	84.38

5 Conclusions and Future Work

We proposed a novel feature vector based on the DCT coefficients of a compressed video to train a model for detecting the platform provenance. Our feature vector uses the interval histograms of the luma and chrominance components of I- and P-frames. An RF classifier trained with our feature vectors was capable of outperforming a previously proposed model for the single and multiple sharing cases on the PREMIER A1 and PREMIER A2 datasets, even when videos are edited. Our future work focuses on using the proposed feature vectors within a deep learning framework on larger datasets.

References

1. L.Ceci. Hours of video uploaded to youtube every minute as of february 2022, 2022. https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/.
2. Cecilia Pasquini, Irene Amerini, and G. Boato. Media forensics on social media platforms: a survey. *EURASIP Journal on Information Security*, 2021, 05 2021.
3. Oliver Giudice, Antonino Paratore, Marco Moltisanti, and Sebastiano Battiato. A classification engine for image ballistics of social data. In *Image Analysis and Processing - ICIAP 2017*, pages 625–636. Springer International Publishing, 2017.
4. Roberto Caldelli, Irene Amerini, and Chang Tsun Li. Prnu-based image classification of origin social network with cnn. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1357–1361, 2018.
5. Junfeng He, Zhouchen Lin, Lifeng Wang, and Xiaoou Tang. Detecting doctored jpeg images via dct coefficient analysis. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 423–435, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
6. 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)*, pages 1–6, 2017.
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*, 14(5):947–954, 2020.
8. Enes Altinisik, Hüsrev Taha Sencar, and Diram Tabaa. Video source characterization using encoding and encapsulation characteristics. *IEEE Transactions on Information Forensics and Security*, 17:3211–3224, 2022.
9. Dasara Shullani, Daniele Baracchi, Massimo Iuliani, and Alessandro Piva. Social network identification of laundered videos based on dct coefficient analysis. *IEEE Signal Processing Letters*, 29:1112–1116, 2022.
10. Irene Amerini, Chang-Tsun Li, and Roberto Caldelli. Social network identification through image classification with cnn. *IEEE Access*, 7:35264–35273, 2019.
11. Irene Amerini, Aris Anagnostopoulos, Luca Maiano, and Lorenzo Ricciardi Celsi. Learning double-compression video fingerprints left from social-media platforms. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2530–2534, 2021.
12. 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*, 7(8):140, aug 2021.
13. 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*, 2017:15, 10 2017.
14. Brian C. Hosler, Xinwei Zhao, Owen Mayer, Chen Chen, James A. Shackleford, and Matthew C. Stamm. The video authentication and camera identification database: A new database for video forensics. *IEEE Access*, 7:76937–76948, 2019.
15. Video Quality Experts Group. Modified jm h.264/avc codec. https://vqeg.github.io/software-tools/encoding/modified-avc-codec/. Accessed: 26-03-2023.
16. Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.