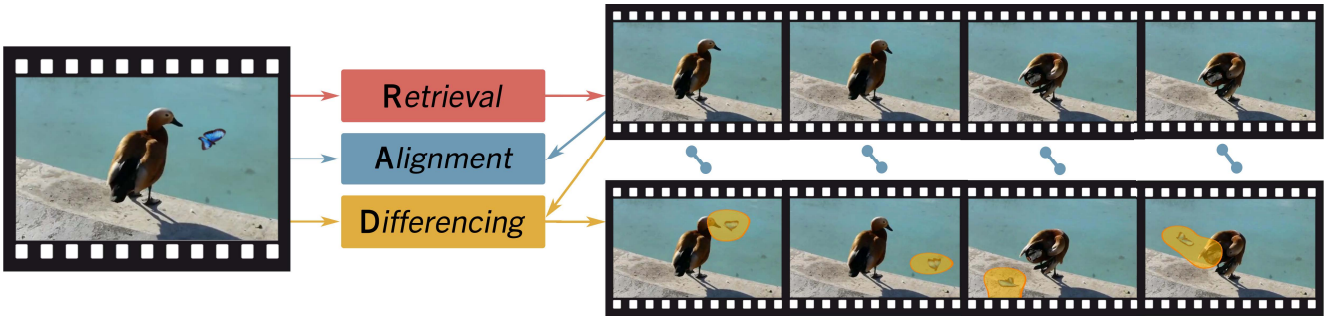
VADER: Video Alignment Differencing and Retrieval

Alexander Black[1] Simon Jenni[2] Tu Bui[1] Md. Mehrab Tanjim[2] Stefano Petrangeli[2]
Ritwik Sinha[2] Viswanathan Swaminathan[2] John Collomosse[1,2]
[1]CVSSP, University of Surrey [2]Adobe Research
{alex.black,t.v.bui}@surrey.ac.uk {jenni,tanjim,petrange,risinha,vishy,collomos}@adobe.com



Abstract

We propose VADER, a spatio-temporal matching, alignment, and change summarization method to help fight misinformation spread via manipulated videos. VADER matches and coarsely aligns partial video fragments to candidate videos using a robust visual descriptor and scalable search over adaptively chunked video content. A transformer-based alignment module then refines the temporal localization of the query fragment within the matched video. A space-time comparator module identifies regions of manipulation between aligned content, invariant to any changes due to any residual temporal misalignments or artifacts arising from non-editorial changes of the content. Robustly matching video to a trusted source enables conclusions to be drawn on video provenance, enabling informed trust decisions on content encountered. Code and data are available at https://github.com/AlexBlck/vader

1. Introduction

Fake news and misinformation are major societal problems, much of it caused by manipulated videos shared to misrepresent events. Detecting manipulation presents only a partial solution; most edited content is not misinformation [22]. Many emerging solutions, therefore, focus on *provenance* - determining the origins of content, what was changed, and by whom [43, 6]. Securely embedding such an audit trail inside video metadata can help consumers make better trust decisions on content, and is the basis of recent open standards (*e.g.* C2PA) [11, 1]. However, such metadata can be trivially stripped or replaced.

We propose VADER; a robust technique matching video fragments to a trusted database of original videos (such as those maintained by journalism or publishing organizations, per C2PA). Once matched and temporally aligned, differences between the query-result pair are visualized to highlight any manipulation of the video (ignoring artifacts due to benign, *i.e.*, non-editorial transformation due to renditions). Any provenance metadata associated with the original may also be displayed. Our technical contributions are:

1. Scalable video fragment (R)etrieval. We learn a robust self-supervised video clip descriptor invariant to various transformations of visual content. Video content is adaptively chunked and represented via these descriptors, which are aggregated and ranked to match and coarsely localize short video fragments within a large corpus of videos.

2. Fine-grained temporal query (A)lignment. We propose a Transformer based alignment method that performs fine-grained (*i.e.*, frame level) localization of the matched video with the query fragment.

3. Space-time (D)ifferencing to visualize manipulation. We learn a spatio-temporal model that 'compares' the aligned query and matched video in order to identify regions of manipulation. The resulting heatmap ignores any discrepancies introduced due to residual frame misalignments or visual artifacts introduced by non-editorial changes such as quality, resolution, or format changes.

4. Video manipulation dataset. We contribute a dataset of 1042 professionally mANipulated videos and mAsK annotatIoNs (ANAKIN) to train and evaluate VADER.

We show VADER's modular design meets three desirable properties we propose for practical video attribution: scalability in search; fine-grained localization of the query in the returned clip; and visualization of areas of manipulation. We evaluate all three properties on our new dataset of manipulated video content (ANAKIN) in addition to standard video benchmarks (VCDB [33] and Kinetics-600 [37]). We believe VADER is a promising step toward addressing fake news by helping users make more informed trust decisions on the video content they view online.

2. Related Work

Studies into video content authenticity generally fall into two camps: detection and attribution. Detection methods primarily evolve from the digital forensics perspective in detecting synthetic (or 'deep fake') content or the presence of artifacts resulting from digital manipulation. The Deep Fake Detection Challenge (DFDC) [14] catalyzed many works [41, 63, 64, 23] exploring the detection of facial manipulation in videos, analogous to the FaceForensic++ dataset [47] in the image domain. Departures from natural image statistics could be identified via noise level analysis in the pixel domain [56] or via neural networks in the frequency domain [21]. These artifacts can be used to localize tampered regions [15, 54] or even determine which GAN architecture synthesized an image [8]. Alternatively, content authenticity can also be studied as an anomaly detection problem, exploiting limitations in synthesis models such as lack of blinking in GAN-based deepfake videos [39] or local anomalous features introduced during the forgery process [59]. However, not all misinformation is created via manipulation; imagery may be misattributed to tell a false story.

Attribution approaches, therefore, employ provenance-tracing methods to determine the origins of content and how it has been transformed from the original. Attribution is the focus of cross-industry coalitions (*e.g.*, CAI [46], Origin [1]) and emerging standards (*e.g.*, C2PA [11]). A robust identifier may be embedded via metadata, watermarking [13, 2], steganography [60, 57]) or obtained via perceptual hash of content (fingerprinting [43, 61, 4]). The identifier serves as a key to lookup provenance information in a database operated by a trusted source [1] or in a decentralized trusted database such as a blockchain [12, 7]. Challenges of fingerprinting include sensitivity to subtle manipulations and robustness to various perturbation sources, such as noise, compression, padding, enhancement, etc., during the re-sharing of media. These challenges can be addressed via object-centric tamper-sensitive hashing [43], or multi-stage approaches [6, 55] that learn an identifier robust to both benign and editorial changes, opting instead to detect or highlight manipulation regions once matched. In all these works, provenance information can be exposed when a match is found via a pairwise comparison between the query and reference content, giving an advantage over blind detection methods.

Content attribution for video is a largely unexplored topic due to challenges in fingerprinting arbitrary-length videos; the presence of noise and possible manipulations along the temporal dimension; video alignment and re-localization; and lacking real-world datasets. Often, the query video is temporally cropped, and the video attribution task requires retrieving the original video and localizing the query within the matched candidate. A closely related problem is near-duplicate retrieval and video copy detection [3, 34, 18, 58]. Early approaches perform fingerprinting with handcrafted features [34, 58, 17], often omitting temporal information and resulting in false positives. In the era of deep learning, long video sequences have been modeled via recurrent networks for re-localization [20], hierarchical attention for attribution [7] or metric learning for moment retrieval and partial copy detection [3, 34]. In a recent work of TransVCL [29], video copy localization is treated as an object detection task on the learnable frame-level similarity matrix between a pair of videos. However, these works assume that the reference video is already identified. Several methods work on pairwise comparisons at inference time [38, 26] for fine-grained retrieval or alignment at the cost of scalability. Drop-DTW [19] supports video alignment however against text description rather than partial videos. VPN [5] performs coarse-level video retrieval via a simple chunking algorithm followed by differencing with the best matched candidate, but does not take temporal cue into account. To our best knowledge, the most closely related state-of-art are [49] for near-duplicate video search and [3] for temporal alignment, both are later shown to under-perform our approach. Uniquely, our method supports video search, fine-grained alignment, and spatio-temporal manipulation visualization. Additionally, we contribute ANAKIN - a dataset of *professionally edited* videos with rich annotations, unlike existing near-duplicate/copy detection video benchmarks containing only simulated (*e.g.*, TRECVID2008 [36]) or simple real-world transformations (*e.g.*, VCDB [33]).

3. Methodology

Following a coarse-to-fine video matching strategy, our method consists of three stages: retrieval, alignment, and differencing. First, we identify the candidate video chunks (retrieval), then arrive at a sub-chunk fine-grained localization (alignment) while also adopting a space-time frame differencing approach that mitigates any remaining fine-grain misalignment. Section 3.1 describes our video retrieval approach. We train an encoder in a self-supervised manner to extract robust video descriptors and use an adaptive chunking strategy to construct an index in a scalable manner. We also introduce an approach for aggregating the retrieval results from multiple temporal chunks into a score for re-ranking, verification, and coarse temporal localization for downstream

alignment. Section 3.2 describes the video alignment module. We propose a sequence matching transformer architecture, which is used to improve the alignment between the query and the retrieved video from chunk-level to a precise frame-to-frame matching. After alignment, the videos undergo pairwise comparison, described in Section 3.3. Using a 3D CNN architecture, we produce a per-frame prediction of the presence of manipulations in query video via a heatmap.

3.1. Scalable Video Retrieval

We aim to robustly match a video fragment to its corresponding entry in a large database, even when the query fragment only consists of a short segment (partial temporal matching) and undergoes various benign degradations and manipulations. To address these challenges and to enable efficient retrieval and storage of long videos, our method (Figure 1) consists of (i) a robust self-supervised video clip descriptor, (ii) an adaptive chunking strategy to reduce the number of descriptors required to represent long videos, and (iii) aggregating chunk-level retrieval results into a single score for ranking and verification.

Robust Video Descriptor. Given an arbitrary-length video $\mathbf{v}$, we split into a non-overlapping sequence of chunks $v_i \in \mathbb{R}^{T\times H\times W\times 3}$. We empirically sample once every 4th frame and set $T = 16$. Thus, each chunk represents about 3 seconds of video at 24fps. We train a 3D-CNN model [51], F, to encode each chunk to a compact descriptor. Concretely, $F(\mathbf{v}) = [\nu_1, \ldots, \nu_n]$ where all ν_i are descriptors of fixed-sized chunks, $\nu_i = F(v_i) \in \mathbb{R}^{512}$. We require an F robust to common visual degradations (benign changes) and manipulations (c.f. subsec. 4.1) and so employ three self-supervised training objectives as proposed in [31]: 1) contrastive learning between augmented video clips, 2) contrastive learning between video and the corresponding audio track, and 3) temporal pretext tasks, *e.g.*, the prediction of video playback speed and direction. The Kinetics-600 dataset [37] is used to pre-train F with these self-supervised objectives.

Adaptive Long-Form Video Indexing. Representing videos as a sequence of per-chunk features would lead to a linear scaling of the index size as a function of video duration. However, as videos have high temporal redundancy, *i.e.*, consecutive video chunks often have similar semantics, we propose an adaptive chunking strategy to reduce the number of descriptors for each video. Our approach defines a threshold τ for consecutive chunk similarity above which we aggregate similar chunks. Adaptive chunking results in shorter video descriptors

$$\hat{F}(\mathbf{v}) = [\bar{\nu}_{0:i_1}, \ldots, \bar{\nu}_{i_m:n}], \tag{1}$$

where aggregated descriptors are given by $\bar{\nu}_{j:k} = \frac{1}{k-j}\sum_{i=j+1}^{k}\nu_i$. The chunk boundaries i_j in Eq. 1 are given by all the indices $i \in \{1, \ldots, n-1\}$ where $d(\nu_i, \nu_{i+1}) < \tau$, *i.e.*, where the cosine similarity between consecutive chunks is below the chosen threshold τ. In practice, we chose τ by computing consecutive chunk similarities for a large and representative set of long videos and set τ to be the k-th percentile of the resulting distribution. Note that a lower τ corresponds to more aggressive chunking and that the chosen percentile roughly corresponds to the compression rate. We also explore the use of a pre-computed fixed k-means clustering of the feature space to define the chunk boundaries as the indices where the cluster assignments ρ change, *i.e.*, $\rho(\nu_i) \neq \rho(\nu_{i+1})$.

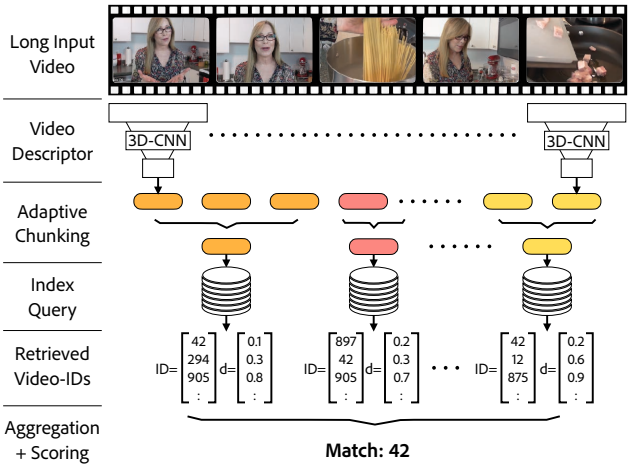


Figure 1. **Robust Scalable Retrieval of Long Videos.** Given a long video, we extract robust descriptors for multiple non-overlapping temporal chunks. We exploit temporal redundancy via our adaptive chunking approach to reduce the number of descriptors required to represent a long video. Finally, we propose an approach to aggregate all the chunk-level retrieval results into a per-video score for re-ranking and verification.

Aggregating and Ranking Search Results. With each video being represented by a sequence of adaptively chunked descriptors (Eq. 1), a question arises of how to store, query, and rank results for videos containing multiple temporal chunks. We store each descriptor $\bar{\nu}$ along with its video ID and temporal position in an efficient inverted file index with Product Quantization (IVFPQ) [35]. Given an adaptively chunked sequence of query descriptors at query time, we propose a method to aggregate multiple chunk-level retrieval results to rank the retrieved videos according to how likely they are to contain the query fragment. To this end, let $\mathbf{S} \in \mathbb{N}^{m\times r}$ and $\mathbf{D} \in \mathbb{R}^{m\times r}$ represent the chunk-level retrieval results for a query $F(\mathbf{v}) \in \mathbb{R}^{m\times 512}$, *i.e.*, S_{ij} is the video-ID for chunk i at rank j and D_{ij} the cosine similarity between the query and retrieved chunk descriptor. Let us further define $\text{rank}_i(s) = \min\{j | S_{ij} = s\}$ as the rank of the first chunk belonging to video s for query chunk i, and $\sigma_i = \text{std}(\{D_{i1}, \ldots, D_{ir}\})$ as the standard deviation of the chunk similarities for query chunk i. We then assign a weight to each combination of query chunk $\bar{\nu}_i$ and video-ID s through

$$w_i(s) = \begin{cases} \frac{\sigma_i}{\text{rank}_i(s)^\lambda} & \text{if } s \in \{S_{i1}, \ldots, S_{ir}\} \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

Per Eq. 2, we assign higher weights when video s appears at a low rank and when the chunk similarities have higher standard deviation (indicating more distinctive chunk descriptors). The parameter λ controls the relative importance of higher rank retrievals and is set to $\lambda = 2$ by default. Finally, the retrieved videos are ranked according to the score

$$\text{score}(s) = \frac{1}{m} \sum_{i=1}^{m} w_i(s) \cdot d_i(s), \quad (3)$$

where $d_i(s)$ is the cosine similarity between query chunk i and the most similar retrieved chunk from video s.

Coarse Temporal Alignment. Next, we describe how to coarsely localize a query segment in a retrieved video and reduce the search space for downstream temporal alignment. Let $t_i(s)$ represent the starting time of the first retrieved chunk of video s for query i. To coarsely localize the query in the retrieved videos, we propose to use the weighted sum of the retrieved chunk starting times

$$\text{start}(s) = \sum_{i=1}^{m} \hat{w}_i(s)\big(t_i(s) - \Delta t_i\big), \quad (4)$$

where Δt_i is the time between the first and i-th query chunk, and the weights $\hat{w}_i(s)$ are per-video normalized versions of the weights in Eq. 2, *i.e.*, $\hat{w}_i(s) = \frac{w_i(s)}{\sum_{j=1}^{m} w_j(s)}$.

3.2. Fine-Grained Video Alignment

We perform fine-grained video alignment to find corresponding matches between frames of the query video fragment, and the frames of the video clip identified and coarsely aligned by the prior retrieval step. This finer-grained alignment is necessary for subsequent manipulation detection, described in Section 3.3. We propose a deep neural network architecture for alignment robust to manipulations and temporal augmentations (Figure 2).

Model Architecture. Inspired from [32, 9, 16], we implement a transformer network $\mathcal{T}$ that, given a query video V, a retrieved video V' and a normalized time coordinate $t \in [0, 1]$ is tasked to output time $t' \in [0, 1]$ such that frame $V(t)$ and $V'(t')$ are temporally aligned. Unlike [32, 9, 16], which work on images, we uniquely leverage a transformer for the pair-wise video alignment task.

For each video frame, we extract a robust 256 dimensional feature embedding using a frame encoder ε. A video consisting of T frames can be represented as a $T \times 256$ sequence of frame descriptors $\varepsilon(V) = [\varepsilon(V_1), \varepsilon(V_2), \ldots, \varepsilon(V_T)]$. Since our approach is agnostic of the feature choice, we follow [3, 18, 45] and use RMAC [50] features and explore other options in the supmat.

We subsample to the length of 20 and concatenate together the feature sequences of two videos V and V'. A vector of positional encodings P is added to the 40×256 sequence before it is used as an input for the transformer encoder $\mathcal{T}_\mathcal{E}$. The output of the transformer encoder, together with the positional encoding of the query time $\mathcal{P}(t)$, is passed into the transformer decoder $\mathcal{T}_\mathcal{D}$. Then, a Multi-Layer Perceptron (MLP) D is used to interpret the output of the transformer decoder and obtain a prediction of the corresponding time $\hat{t}'$. To summarize, inputs to the model are $c = [\varepsilon(V), \varepsilon(V')] + P$ and predictions computed as

$$\hat{t}' = \mathcal{T}(t|V, V') = D\Big(\mathcal{T}_\mathcal{D}\big(\mathcal{P}(t), \mathcal{T}_\mathcal{E}(c)\big)\Big). \quad (5)$$

As in [32], we use a linear increase in frequency for the positional encoding instead of log-linear to yield a more stable optimization. The positional encoding for a given time t is defined as $\mathcal{P}(t) = [\sin(1\pi t), \cos(1\pi t), \ldots, \sin(k\pi t), \cos(k\pi t)]$, where $k = 128$ results in a sequence of length 256.

Learning Objectives. The network is trained to minimize two losses: mean squared error (MSE) $\mathcal{L}_{MSE}$ and a cycle loss $\mathcal{L}_{cycle}$. The first loss is defined as the error between the ground truth time t' and the network's prediction $\hat{t}'$ as: $\mathcal{L}_{MSE} = \|t' - \hat{t}'\|_2^2$. The cycle loss ensures that the network is able to reconstruct the query time t using its own prediction of the corresponding time $\hat{t}'$: $\mathcal{L}_{cycle} = \|t - \mathcal{T}(\hat{t}'|V', V)\|_2^2$. The total loss is $\mathcal{L} = w_m\mathcal{L}_{MSE} + w_c\mathcal{L}_{cycle}$ where loss weights $w_m = 0.6, w_c = 0.4$ are set empirically.

Frame Sampling. During training, we subsample 20 frames from each video. From the non-manipulated video V of length N we select a set of frames $[V_{s+0k}, V_{s+1k} \ldots V_{s+19k}]$ where the starting frame s and stride k can be any combination of integers that satisfies the condition $0 \leq s < s + 19k \leq N$, chosen at random. For the second video, the starting frame is shifted by $19k * \alpha$ rounded to the nearest integer, where $\alpha \in [0, 0.25]$. This ensures that both sequences' coverage of the video overlaps by at least 50%. Additionally, the indices of the second video are subjected to random perturbations $\mathbf{g}_i \sim \mathcal{N}(\mu, \sigma^2)$ rounded to the nearest integer (we set $\mu = 0$ and $\sigma = 4$): $[V'_{s'+0k+\mathbf{g}_0}, V_{s'+1k+\mathbf{g}_1} \ldots V'_{s'+19k+\mathbf{g}_{19}}]$. The set of pairs of query times t and corresponding times t' consists of normalized time coordinates of V that overlap with V' and thus can vary in size between 10 and 20 pairs. During inference, to ensure no loss in precision, no subsampling is performed but rather each consecutive set of 20 frames is evaluated in a sliding window fashion.

Implementation Details. The transformer encoder $\mathcal{T}_\mathcal{E}$ is implemented as 6 layers of 8-head self-attention modules, and the transformer decoder $\mathcal{T}_\mathcal{D}$ as 6 layers of 8-head encoder-decoder attention modules. There is no self-attention in the decoder to make queries independent of each other. The

MLP module D consists of $d_{mlp} = 6$ fully-connected layers with a hidden dimension size of $h_{mlp} = 2048$. These values were determined using grid-search with $d_{mlp} \in [1, 10]$ and $h_{mlp} \in [2^5, 2^{12}]$.

3.3. Video Manipulation Detection

We leverage the network inflation method [10] to extend the image comparator network (ICN) [6] to video domain. We later show that encoding multiple frames via a 3D-CNN architecture provides more temporally stable differencing visualizations than single-frame methods like ICN. Additionally, introducing slight misalignments between the frames during training results in a model resilient to alignment errors that may occur during the video alignment step.

Model Architecture. Our model (Figure 3) takes as input two video clips of size T×C×W×H = 16×3×224×224. The model output is a 16×7×7 grid which, when interpolated to the original frame size, serves as a heatmap that highlights the edited region of the video. We inflate ICN using the approach described in [10], where we substitute 2D convolutions with weights repeated along the time dimension to produce 3D convolutions. In our case, the stride along the time dimension is always kept at 1, resulting in an output with the same temporal resolution as the input - one heatmap per frame.

We train a video differencing 3D-CNN $\mathcal{D}$ that, given a query video V' and a retrieved and aligned video V, outputs a heatmap $h \in \mathbb{R}^{16\times7\times7}$ and a binary classification score c, signifying whether the query video has been manipulated. First, both videos are passed separately through a shared feature extractor $\mathcal{D}_E$:

$$z_0 = \mathcal{D}_E(V);\ z'_0 = \mathcal{D}_E(V') \in \mathbb{R}^{T\times H'\times W'\times C}. \quad (6)$$

$\mathcal{D}_E$ contains three 3D convolution layers separated by ReLU, batch norm, and max pooling. It outputs features at $\frac{1}{4}$ resolution ($H' = H/4, W' = W/4$ and we set $C = 64$). Features are concatenated to form an input of size $T\times H'\times W'\times 2C$ to the next module $\mathcal{D}_S$ producing $z = \mathcal{D}_S([z_0, z'_0]) \in \mathbb{R}^{T\times256}$. $\mathcal{D}_S$ contains 4 residual blocks [28] then an FC layer.

Learning Objective. To correctly recognize whether the query video was manipulated and produce accurate localization masks of the manipulated regions, we train the differencing module with two losses. First, the similarity loss $\mathcal{L}_S$ minimizes the cosine distance between the manipulation heatmaps derived from z and the ground truth heatmaps. We produce the heatmap at resolution $T \times n \times n$ from z via a FC layer $\mathcal{D}_H(z) \in \mathbb{R}^{T\times n^2}$ and compute the loss:

$$\mathcal{L}_S = \frac{1}{T}\sum_{t=1}^{T} 1 - \frac{\mathcal{D}_H(z_t)\cdot G_t}{|\mathcal{D}_H(z_t)|\,|G_t|}) \quad (7)$$

where G_t is the ground truth manipulation heatmap of the t-th frame. We define the output heatmap resolution $n = 7$ during training. At test time, the 7×7 heatmaps are interpolated using bicubic interpolation to the original resolution $H \times W$ and super-imposed on the query video. The heatmaps are continuous but can be thresholded for more intuitive visualization. Additionally, we apply a classification loss $\mathcal{L}_C$ computed as binary cross-entropy loss:

$$\mathcal{L}_C = \frac{1}{T}\sum_{t=1}^{T} -\log\frac{e^{c_{t,y_t}}}{\sum_{i=1}^{2} e^{c_{t,i}}} \quad (8)$$

where $c = \mathcal{D}_C(z) \in \mathbb{R}^{T\times2}$ with a FC layer $\mathcal{D}_C$, and y_t is the classification target, indicating if the t-th frame of the query contains manipulations.

The total loss is $\mathcal{L}_D = w_s\mathcal{L}_S + w_c\mathcal{L}_C$ where loss weights $w_s = w_c = 0.5$ are set empirically.

4. Experiments and Discussion

We evaluate the performance of the VADER retrieval, alignment and detection stages. We use three datasets: ANAKIN, Kinetics-600, and VCDB (Sec 4.1). ANAKIN is our end-to-end dataset used to evaluate all three stages. Kinetics is additionally combined with ANAKIN to evaluate retrieval scalability. VCDB is a benchmark additionally used to baseline the accuracy of our retrieval method.

4.1. Datasets

ANAKIN[1] is a new dataset of mANipulated videos and mAsK annotatIoNs, which we introduce. To our best knowledge, ANAKIN is the first real-world dataset of professionally edited video clips paired with source videos, edit descriptions, and binary mask annotations of the edited regions. ANAKIN consists of 1042 videos in total, including 352 edited videos from the VideoSham [42] dataset, plus 690 new videos collected from the Vimeo platform (see supmat for details and dataset size analysis). For each video, professional video editors were given a task (*e.g.*, "change the color of the shirt of the person running in the background") and a time interval on which to perform the task. Editorial tasks include: frame-level manipulations (duplication, reversing, dropping, speeding up, slowing down), audio changes (addition, replacement), splicing (adding objects, text, or entities), inpainting (removing objects, text or entities), and swap (background or color). The mean length of the original videos and edited clips are 120s and 5s respectively. The video editors were also asked to provide binary segmentation masks as the ground truth for their edited regions. Examples of original and manipulated videos along with their segmentation masks, can be found in Figure 4.

ANAKIN is used to train and evaluate VADER's alignment and manipulation detection models in Sec. 4.3 and 4.4, with an 80:20 train-test split in both tasks. Since we focus on

[1]This dataset will be released as CC-BY 4.0 upon publication.

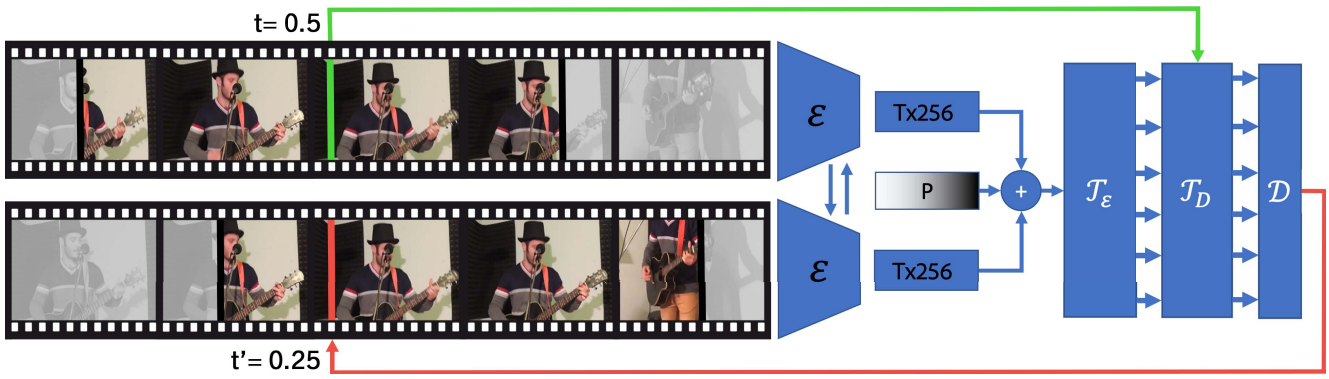


Figure 2. Architecture diagram of the proposed alignment transformer $\mathcal{T}$. Sequences of robust fingerprints are encoded using $\varepsilon(.)$[6] from the query V and candidate videos V'. Together with a positional encoding, they are fed into a transformer encoder $\mathcal{T}_\mathcal{E}$. Given query times t, the decoder transformer $\mathcal{T}_\mathcal{D}$ provides an output, which is interpreted by an MLP D to match query times with corresponding times t'.

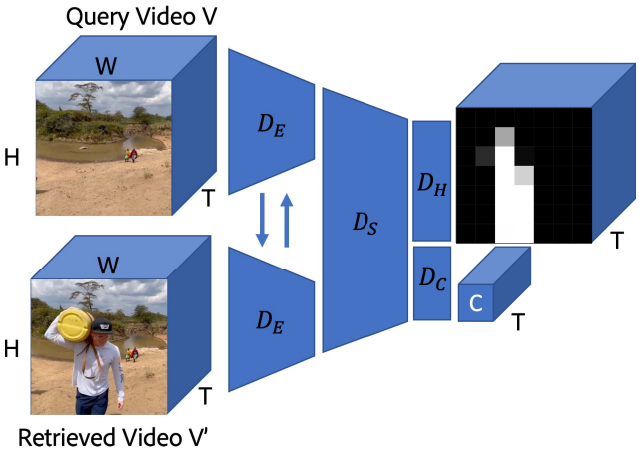


Figure 3. Architecture of the VADER video differencing module. Video tensors are encoded separately with $\mathcal{D}_E$, then a joint embedding is learned in $\mathcal{D}_S$. The embedding is used to produce a heatmap, localizing frame manipulations. A binary classification is also made per frame, marking it as manipulated or not. For non-manipulated frames, the heatmap is zeroed out.

Chunking	Comp. Rate	Aug.-K600 R@1	Aug.-K600 R@5	ANAKIN R@1	ANAKIN R@5
All	32×	67.7	71.2	81.9	84.6
None	-	**76.6**	81.3	96.9	98.6
Constant	2×	75.0	80.0	96.5	97.9
Constant	4×	72.6	77.8	94.7	95.9
Adaptive (k-means)	2×	76.0	**81.9**	96.9	98.2
Adaptive (sim-thresh)	2×	**76.6**	81.5	**97.2**	**98.8**
Adaptive (sim-thresh)	4×	74.0	79.8	96.1	95.9

Table 1. **Adaptive Chunking.** We compare the retrieval performance of our proposed adaptive chunking to baselines retaining all chunks (None), aggregating all the chunks per video (All), and constant chunking (averaging $k \in \{2, 4\}$ consecutive chunks).

visual manipulation detection, videos in which manipulation

Scoring	Aug.-K600 R@1	Aug.-K600 R@5	Aug.-K600 F_1	ANAKIN R@1	ANAKIN R@5	ANAKIN F_1
Count	62.9	77.2	52.0	88.6	98.3	51.6
Max. Sim.	74.8	79.0	50.5	**97.0**	98.5	93.5
Uniform	75.2	79.6	64.3	96.9	98.5	94.6
Ours (w/o σ)	**76.6**	81.2	84.2	96.9	**98.6**	97.6
Ours	**76.6**	**81.7**	**88.0**	96.9	**98.6**	**97.7**

Table 2. **Retrieval Aggregation and Scoring.** We compare our scoring function (Eq. 3) to several alternative formulations.

Method	SP	SR	F_1-Score
CNN [34]	-	-	0.6503
TH+CC+ORB [24]	0.5052	0.9294	0.6546
LAMV [3]	-	-	0.6870
Q-Learning [25]	0.8829	0.7355	0.8025
BTA [62]	0.7600	0.7500	0.7549
VSAL [26]	**0.8971**	0.8462	0.8709
FPVCD [49]	-	-	0.8764
Ours	0.8652	**0.9753**	**0.9167**

Table 3. **Video Copy Detection on VCDB.** We compare to prior results on VCDB reporting maximum segment-level F_1-scores and associated precision (SP) and recall (SR) as defined in [33].

is difficult to visualize, *e.g.*, audio replacement, are excluded from our manipulation detection experiments.

(Aug.-)K600 consists of 60,000 videos in the Kinetics-600 [37] test set. We use the *full-length* source videos (not just the 10-second labeled clips) to build the search index for our scalable retrieval experiment. This results in a search index containing approximately 4M descriptors (without adaptive chunking). Retrieval experiments on ANAKIN also include these descriptors as distractors. We also generate a query set **Aug.-K600** of benign augmented query clips that are

temporally truncated to test partial retrieval. In addition to ANAKIN, the Kinetics-600 training partition is used to train the robust descriptors for the retrieval stage.

VCDB [33] is a well-established video copy detection benchmark. It consists of over 9K pairs of labeled in-the-wild copied video fragments. We use the 'core' dataset to make an additional evaluation of our retrieval method versus partial video copy detection methods.

Benign (Non-Editorial) Augmentations. During the training of all stages of VADER, we apply random video augmentations to clips using AugLy [44] to simulate non-editorial (benign) transformation of content during online distribution. The same set of random augmentations is applied to query fragments, where indicated in the evaluation of each of the 3 VADER stages. These augmentations encompass in-place manipulations (color jittering, blurring, noise, etc.), geometric transforms (cropping, rotations, flipping), and temporal cropping (see supmat).

4.2. Scalable Video Fragment Retrieval

Evaluation Metrics. To evaluate the retrieval performance, we report recall-at-k (R@k) and the maximum fragment-level F1 score as defined in [33]. While recall indicates how well our method can rank the retrieved videos, the F1 score captures the method's ability for coarse temporal localization (detections must overlap with ground truth) and its ability to discriminate between true and false matches.

Adaptive Chunking. We compared our adaptive chunking to several baselines regarding retrieval performance and achieved compression rate (*i.e.*, the average reduction in descriptor length) in Table 1. We consider the following baselines: **All** - In this case, each video is represented via a single aggregated descriptor, *i.e.*, $\hat{F}(\mathbf{v}) = [\bar{\nu}_{0:n}]$. This naive single-descriptor approach corresponds to the maximal achievable compression via chunking. **None** - No chunking is applied, and as a result, no additional compression is achieved. **Constant** - In these variants, the temporal chunking is not adaptive but instead aggregates a fixed number of $k \in \{2, 4\}$ consecutive descriptors, *i.e.*, $\hat{F}(\mathbf{v}) = [\bar{\nu}_{0:k}, \ldots, \bar{\nu}_{n-k:n}]$.

The results in Table 1 indicate the importance of representing long videos with a variable number of temporal descriptors (as in Eq. 1) to achieve good retrieval performance (comparing **All** to the other approaches). Furthermore, we observe consistently and significantly better performance from adaptive chunking compared to the constant chunking baselines at the same compression level. Most noteworthy adaptive chunking can achieve the same or better retrieval performance at $2\times$ compression compared to the non-chunked baseline. This results in storage savings and faster response time in practice due to reduced index size. Given the comparable performance of consecutive chunk-similarity thresholding and clustering for adaptive chunking, we favor similarity thresholding in practice due to its simplicity and better control over the compression rate (the connection between the number of clusters and the compression rate is non-trivial).

Aggregating Chunk-Level Search Results. We evaluate our proposed ranking and match verification score in Eq. 3 and compare in Table 2 to the following alternative formulations and baselines: **Count** - We replace the score in Eq. 3 with $\text{score}(s) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}[s \in \{S_{i1}, \ldots, S_{ir}\}]$, *i.e.*, the proportion of query chunks for which a chunk of video s was in the top r retrievals. **Max. Sim.** - We set $\text{score}(s)$ to be the maximum similarity achieved for any retrieved chunk belonging to video s. **Uniform:** We replace the chunk weight in Eq. 2 with $w_i(s) = 1$ if $s \in \{S_{i1}, \ldots, S_{ir}\}$. The score is an equal weighted sum of the best per-chunk similarities. **Ours (w/o σ):** We set $\sigma_i = 1$ in Eq. 2 and remove dependence on per-query variation in similarities.

Table 2 shows our scoring approach to improve the ranking of retrieved videos and significantly improves the F1 score vs. baselines. Our proposed weighting and aggregation of the chunk-level retrieval results thus can better identify correct matches, by incorporating the chunk-level ranking and variation of similarities in our method.

Video Copy Detection on VCDB. We compare against prior approaches for video copy detection on VCDB [33] in Table 3. Our approach outperforms the prior works by a considerable margin and does so in a more scalable manner. The closest existing solution [49], which also relies on an approximate nearest-neighbor lookup, still leverages additional networks to score and localize candidate matches. In contrast, our approach solely parses the chunk-level retrieval results for coarse localization and scoring without incurring notable additional computing cost.

4.3. Evaluating Alignment

We evaluate VADER's alignment performance by reporting the percentage of queries for which the alignment error is below a threshold $[0.1s, 1s, 10s]$.

We compare against three baselines – for fair comparison we perform alignment using RMAC [50] features in all runs. We report the alignment performance results in Table 4 for four cases: **clean** - query clips are not edited, simply trimmed versions of the originals, **clean+benign** - simple in-place augmentations are applied to query clips, **manip** - clips are manipulated by professional video editors, **manip+benign** both of the above. In all cases, the query video V_q is aligned with the original V_o. Table 4 shows that VADER outperforms the other baselines in all four cases. Since the model is trained to be robust, the introduction of benign augmentations does not significantly affect performance. While all methods perform worse on more challenging manipulations, VADER experiences a lower performance dip, compared to the closest baseline LAMV [3].

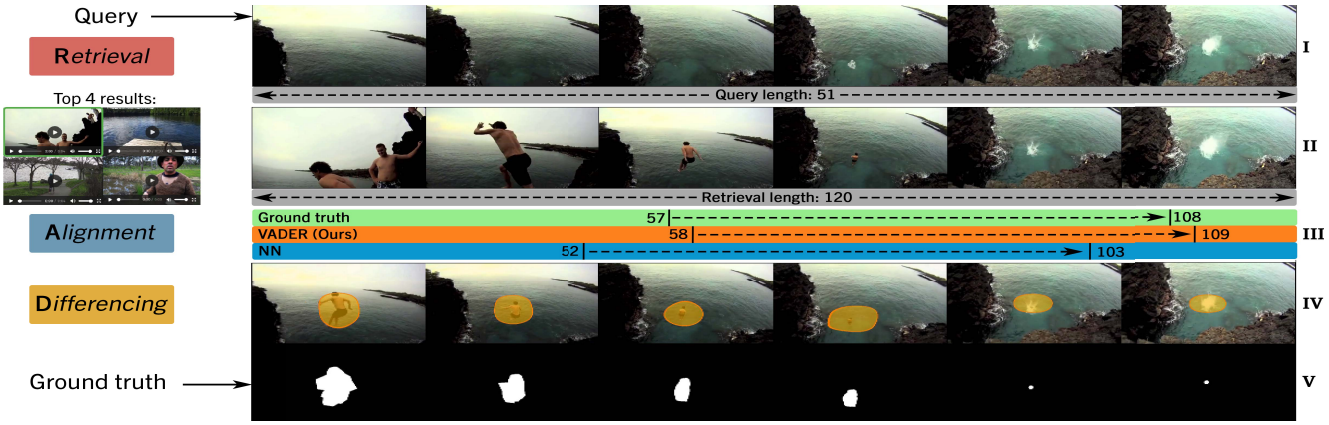


Figure 4. An example of video manipulation detection using VADER. I: a video consisting of 51 frames is used as a query. II : a correct original video with 120 frames is retrieved (top-4 results inset - left). III: alignment module is used to temporally localize the query withing the retrieved video, fifth 1 frame error. IV: differencing module highlights the manipulated region (superimposed here on original video for clarity). V: ground truth mask of the manipulated region from ANAKIN.

Method	clean			clean + benign			manip			manip + benign		
	@0.1s	@1s	@10s	@0.1s	@1s	@10s	@0.1s	@1s	@10s	@0.1s	@1s	@10s
VADER	**69.5**	**91.6**	**97.4**	**64.2**	**78.4**	**93.2**	**54.2**	**81.1**	**94.7**	**53.7**	**74.2**	**91.6**
LAMV [3]	54.7	75.3	87.9	44.7	68.9	85.3	38.4	66.3	86.3	30.0	59.5	83.2
CTE [18]	4.8	25.4	67.7	4.2	20.1	67.2	4.8	21.2	68.3	3.7	19.6	68.3
TMK [45]	2.6	16.8	69.5	1.6	18.4	63.2	1.6	15.8	66.8	3.2	17.9	59.5

Table 4. Alignment module evaluation on ANAKIN dataset. We report percentage of videos aligned better than a threshold value (0.1s, 1s, 10s). VADER outperforms baseline methods in all four query sets and shows more resilience to benign transformations.

Method	IOU↑					Grad↓				
Shift	0	1	2	3	4	0	1	2	3	4
VADER (ours)	**0.804**	**0.801**	**0.786**	**0.760**	**0.729**	**0.0179**	**0.0180**	**0.0182**	**0.0189**	**0.0197**
ICN [6]	0.448	0.408	0.372	0.347	0.331	0.0245	0.0373	0.0408	0.0369	0.0343
ResNetConv [28]	0.354	0.361	0.350	0.358	0.317	0.0241	0.0264	0.0226	0.0230	0.0242
SSD	0.260	0.218	0.201	0.193	0.188	0.0230	0.0271	0.0226	0.0253	0.0276

Table 5. Differencing module evaluation with temporal jittering. Higher IOU scores between the predicted and ground truth heatmaps signify better accuracy of localization, lower values of gradients of the IOUs indicate more temporal stability. Both are reported for 5 different values of temporal shift, between 0 and 4.

4.4. Evaluating manipulation detection

To generate the edited region outline, we upscale the $16 \times 7 \times 7$ network output to the frame resolution $H \times W$, convert to it binary with a threshold and compute Intersection over Union (IoU) with the ground truth $\text{IoU} = \frac{1}{16} \sum_{i=1}^{16} \frac{U_i \cap T_i}{U_i \cup T_i}$, where T_i is the $H \times W$ binary ground truth heatmap, U_i is the predicted heatmap after interpolation and thresholding.

We compare VADER against five baselines. These baselines are image-based, and thus, we apply them to each frame pair separately. **Sum of Squared Distances (SSD)** - we compute SSD between two images at the pixel level, resize it to 7×7, then resize it back before thresholding to create continuity in the detected heatmap. **ResNetConv** - we extract $7 \times 7 \times 2048$ features from pre-trained ImageNet ResNet50 model for both query and original images and calculate the distance between them. **ErrAnalysis** - a shallow blind detection technique [56] leveraging JPEG compression. **MantraNet** - a supervised blind detection method [59] that detects anomalous regions. **Image Comparator Network (ICN)**[6] - neural network for pairwise image comparison.

We introduce misalignment between two videos by offsetting all frames by a small amount (0-4 frames) and report IOU in Table 5. When the original and edited video are perfectly aligned, the image comparison methods are on par with scores expected when evaluating pairwise with frames. Our proposed method is able to leverage the temporal dimension of the data to achieve much stronger results in the presence of misalignment. To demonstrate robustness to be-

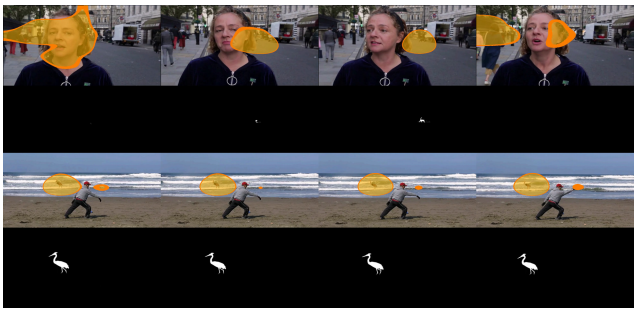


Figure 5. VADER limitations. Top: the model attempts to highlight the object inserted into the video even when it is occluded, some frames are falsely classified as manipulated. Bottom: cyclic actions of both the person and the waves leads to incorrect alignment, person's hand incorrectly marked as manipulated.

Method	IOU	
	manip	manip + benign
VADER	**0.804**	**0.723**
VADER (end-to-end)	0.798	0.716
ICN [6]	0.448	0.311
ResNetConv [28]	0.354	0.174
SSD	0.260	0.205
ErrAnalysis [56]	0.231	0.058
MantraNet [59]	0.140	0.065

Table 6. Differencing module evaluation with in-place transformations. Higher IOU scores between the predicted and ground truth heatmaps signify better accuracy of localization.

nign transformations, in Table 6 we show how IOU changes with such transformation (see supmat for further evaluation of misalignment resilience). In addition to reporting VADER results on the same data as the baselines, we also show the performance achieved in an end-to-end manner. Our proposed space-time method exhibits strong temporal stability, lacking in pairwise frame based approaches. We quantify temporal stability as the mean gradient of IOU within all test videos. Table 5 shows that even in the case of no misalignment, our method results in less flickering. Figure 4 shows examples of the end-to-end VADER pipeline and we discuss limitations of alignment and differencing in Figure 5.

5. Conclusion

We presented VADER - a spatio-temporal method for robust video retrieval, alignment, and differencing. VADER matches video fragments circulating online to a trusted database of original clips and associated provenance metadata. Providing users with information about video provenance and any manipulations present enables them to make informed trust decisions about content they encounter, providing a solution to the 'soft binding' matching called for in emerging open standards for media provenance *e.g.* C2PA [11]. Given a query video fragment, VADER retrieves the original and provides coarse temporal localization of the fragment within the original clip. VADER's alignment module produces a fine-grained frame-to-frame correspondence between the videos, refining the temporal localization. Aligned videos are compared via the space-time differencing module, which highlights the manipulations on each frame with a heatmap. The module is able to compensate for minor misalignments remaining through the end-to-end processing. We train and evaluate on ANAKIN - a novel dataset of manipulated videos and corresponding ground truth annotations. The dataset is released as a further contribution of this paper. Further work could address alternative ways to summarize manipulation in matched videos, for example captioning changes identified via the heatmap using a vision-language model. This might enable larger numbers of results to be reviewed without high cognitive load.

Acknowledgement

We thank the video editors who worked on ANAKIN: Oleksandr Tsiundyk[52], Anne Tsjornaja[53], Daniel Matheus[40], Nageeb Hassan[27], Ilya Ievsiukov[30], Sergey Skrypnik[48]. Work supported in part by DECaDE: EPSRC Grant EP/T022485/1.

References

[1] J. Aythora, R. Burke-Agüero, A. Chamayou, S. Clebsch, M. Costa, J. Deutscher, N. Earnshaw, L. Ellis, P. England, C. Fournet, et al. Multi-stakeholder media provenance management to counter synthetic media risks in news publishing. In *Proc. Intl. Broadcasting Convention (IBC)*, 2020. 1, 2

[2] S. Baba, L. Krekor, T. Arif, and Z. Shaaban. Watermarking scheme for copyright protection of digital images. *IJCSNS*, 9(4), 2019. 2

[3] Lorenzo Baraldi, Matthijs Douze, Rita Cucchiara, and Hervé Jégou. Lamv: Learning to align and match videos with kernelized temporal layers. In *Proc. CVPR*, pages 7804–7813, 2018. 2, 4, 6, 7, 8

[4] A. Bharati, D. Moreira, P.J. Flynn, A. de Rezende Rocha, K.W. Bowyer, and W.J. Scheirer. Transformation-aware embeddings for image provenance. *IEEE Trans. Info. Forensics and Sec.*, 16:2493–2507, 2021. 2

[5] Alexander Black, Tu Bui, Simon Jenni, Viswanathan Swaminathan, and John Collomosse. Vpn: Video provenance network for robust content attribution. In *Proc. CVMP*, pages 1–10, 2021. 2

[6] Alexander Black, Tu Bui, Hailin Jin, Vishy Swaminathan, and John Collomosse. Deep image comparator: Learning to visualize editorial change. *Proc. CVPR WS*, pages 972–980, 6 2021. 1, 2, 5, 6, 8, 9

[7] T. Bui, D. Cooper, J. Collomosse, M. Bell, A. Green, J. Sheridan, J. Higgins, A. Das, J. Keller, and O. Thereaux. Tamper-proofing video with hierarchical attention autoencoder hashing on blockchain. *IEEE Trans. Multimedia (TMM)*, 22(11):2858–2872, 2020. 2

[8] Tu Bui, Ning Yu, and John Collomosse. Repmix: Representation mixing for robust attribution of synthesized images. In *Proc. ECCV*, pages 146–163. Springer, 2022. 2

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 4

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2017. 5

[11] Coalition for Content Provenance and Authenticity. Draft technical specification 0.7. Technical report, C2PA, 2021. 1, 2, 9

[12] J. Collomosse, T. Bui, A. Brown, J. Sheridan, A. Green, M. Bell, J. Fawcett, J. Higgins, and O. Thereaux. ARCHANGEL: Trusted archives of digital public documents. In *Proc. ACM Doc.Eng*, 2018. 2

[13] P. Devi, M. Venkatesan, and K. Duraiswamy. A fragile watermarking scheme for image authentication with tamper localization using integer wavelet transform. *J. Computer Science*, 5(11):831–837, 2019. 2

[14] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (DFDC) dataset. *CoRR*, abs/2006.07397, 2020. 2

[15] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *TPAMI*, 45(3):3539–3553, 2022. 2

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020. 4

[17] Matthijs Douze, Hervé Jégou, and Cordelia Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Trans. Multimedia*, 12(4):257–266, 2010. 2

[18] Matthijs Douze, Jérôme Revaud, Jakob J. Verbeek, Hervé Jégou, and Cordelia Schmid. Circulant temporal encoding for video retrieval and temporal alignment. *IJCV*, 119:291–306, 2015. 2, 4, 8

[19] Mikita Dvornik, Isma Hadji, Konstantinos G Derpanis, Animesh Garg, and Allan Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. *NeurIPS*, 34:13782–13793, 2021. 2

[20] Yang Feng, Lin Ma, Wei Liu, Tong Zhang, and Jiebo Luo. Video re-localization. In *Proc. ECCV*, pages 51–66, 2018. 2

[21] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *Proc. ICML*, pages 3247–3258. PMLR, 2020. 2

[22] S. Gregory. Ticks or it didn't happen. Technical report, Witness.org, 2019. 1

[23] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Deepfake detection by analyzing convolutional traces. In *Proc. CVPR WS*, pages 666–667, 2020. 2

[24] Zobeida Jezabel Guzman-Zavaleta and Claudia Feregrino-Uribe. towards a video passive content fingerprinting method for partial-copy detection robust against non-simulated attacks. *PLOS one*, 11(11):e0166047, 2016. 6

[25] Z Jezabel Guzman-Zavaleta and Claudia Feregrino-Uribe. Partial-copy detection of non-simulated videos using learning at decision level. *Multimedia Tools and Applications*, 78:2427–2446, 2019. 6

[26] Zhen Han, Xiangteng He, Mingqian Tang, and Yiliang Lv. Video similarity and alignment learning on partial video copy detection. In *Proc. ACM Int. Conf. Multimedia*, pages 4165–4173, 2021. 2, 6

[27] Nageeb Hassan. Upwork freelancer's profile. https://www.upwork.com/freelancers/nageebhassan, 2023. [Online; accessed 21-Mar-2023]. 9

[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 5, 8, 9

[29] Sifeng He, Yue He, Minlong Lu, Chen Jiang, Xudong Yang, Feng Qian, Xiaobo Zhang, Lei Yang, and Jiandong Zhang. Transvcl: Attention-enhanced video copy localization network with flexible supervision. In *AAAI*, volume 37, pages 799–807, 2023. 2

[30] Ilya Ievsiukov. Upwork freelancer's profile. https://www.upwork.com/freelancers/~01c7769c28d08e1ac7, 2023. [Online; accessed 21-Mar-2023]. 9

[31] Simon Jenni, Alexander Black, and John Collomosse. Audio-visual contrastive learning with temporal self-supervision. *arXiv preprint arXiv:2302.07702*, 2023. 3

[32] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. *Proc. ICCV*, pages 6187–6197, 3 2021. 4

[33] Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang. Vcdb: a large-scale database for partial copy detection in videos. In *Proc. ECCV*, pages 357–371. Springer, 2014. 2, 6, 7

[34] Yu-Gang Jiang and Jiajun Wang. Partial copy detection in videos: A benchmark and an evaluation of popular methods. *IEEE Trans. Big Data*, 2(1):32–42, 2016. 2, 6

[35] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 3

[36] Alexis Joly, Julien Law-To, and Nozha Boujemaa. Inria-imedia trecvid 2008: Video copy detection. In *TRECVID*, 2008. 2

[37] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 2, 3, 6

[38] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Visil: Fine-grained spatio-temporal video similarity learning. In *Proc. ICCV*, pages 6351–6360, 2019. 2

[39] Y. Li, M-C. Ching, and S. Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *Proc. IEEE WIFS*, 2018. 2

[40] Daniel Matheus. Upwork freelancer's profile. https://www.upwork.com/freelancers/danielm153, 2023. [Online; accessed 21-Mar-2023]. 9

[41] Ghazal Mazaheri and Amit K Roy-Chowdhury. Detection and localization of facial expression manipulations. In *Proc. WCACV*, pages 1035–1045, 2022. 2

[42] Trisha Mittal, Ritwik Sinha, Viswanathan Swaminathan, John Collomosse, and Dinesh Manocha. Video manipulations beyond faces: A dataset with human-machine analysis, 2022. 5

[43] E. Nguyen, T. Bui, V. Swaminathan, and J. Collomosse. Oscar-net: Object-centric scene graph attention for image attribution. In *Proc. ICCV*, 2021. 1, 2

[44] Zoe Papakipos and Joanna Bitton. Augly: Data augmentations for robustness, 2022. 7

[45] Sébastien Poullot, Shunsuke Tsukatani, Anh Phuong Nguyen, Hervé Jégou, and Shin'ichi Satoh. Temporal Matching Kernel with Explicit Feature Maps. In *ACM Multimedia 2018*, pages 1–10, Brisbane, Australia, Oct. 2015. ACM Press. 4, 8

[46] L. Rosenthol, A. Parsons, E. Scouten, J. Aythora, B. MacCormack, P. England, M. Levallee, J. Dotan, et al. Content authenticity initiative (CAI): Setting the standard for content attribution. Technical report, Adobe Inc., 2020. 2

[47] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, pages 1–11, 2019. 2

[48] Sergey Skrypnik. Upwork freelancer's profile. https://www.upwork.com/freelancers/~018c3c040b98565402, 2023. [Online; accessed 21-Mar-2023]. 9

[49] Weijun Tan, Hongwei Guo, and Rushuai Liu. A fast partial video copy detection using knn and global feature database. In *Proc. WCACV*, pages 2191–2199, 2022. 2, 6, 7

[50] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *CoRR*, abs/1511.05879, 2016. 4, 7

[51] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition, 2017. 3

[52] Oleksandr Tsiundyk. Upwork freelancer's profile. https://www.upwork.com/freelancers/~01cb8c5ea5ec44acbf, 2023. [Online; accessed 21-Mar-2023]. 9

[53] Anne Tsjornaja. Upwork freelancer's profile. https://www.upwork.com/freelancers/~013cdbc72e6dff53d0, 2023. [Online; accessed 21-Mar-2023]. 9

[54] S-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. Efros. Detecting photoshopped faces by scripting photoshop. In *Proc. ICCV*, 2019. 2

[55] S-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *Proc. CVPR*, 2020. 2

[56] W. Wang, J. Dong, and T. Tan. Tampered region localization of digital color images based on jpeg compression noise. In *Intl. WS Digital Watermarking*, pages 120–133. Springer, 2010. 2, 8, 9

[57] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *Proc. ICMR*, pages 87–95, 2019. 2

[58] Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In *Proc. ACM Multimedia*, pages 218–227, 2007. 2

[59] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proc. CVPR*, pages 9543–9552, 2019. 2, 8, 9

[60] Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry S Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. In *Proc. ICLR*, 2021. 2

[61] X. Zhang, Z. H. Sun, S. Karaman, and S.F. Chang. Discovering image manipulation history by pairwise relation and forensics tools. *IEEE J. Selected Topics in Signal Processing.*, 14(5):1012–1023, 2020. 2

[62] Yue Zhang and Xinxiang Zhang. Effective real-scenario video copy detection. In *Proc. ICPR*, pages 3951–3956. IEEE, 2016. 6

[63] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proc. CVPR*, pages 2185–2194, 2021. 2

[64] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proc. ACM Multimedia*, pages 2382–2390, 2020. 2