VPN: Video Provenance Network for Robust Content Attribution

Alexander Black
CVSSP, University of Surrey
alex.black@surrey.ac.uk

Tu Bui
CVSSP, University of Surrey
t.v.bui@surrey.ac.uk

Simon Jenni
Adobe Research
jenni@adobe.com

Viswanathan Swaminathan
Adobe Research
vishy@adobe.com

John Collomosse
University of Surrey | Adobe Research
collomos@adobe.com

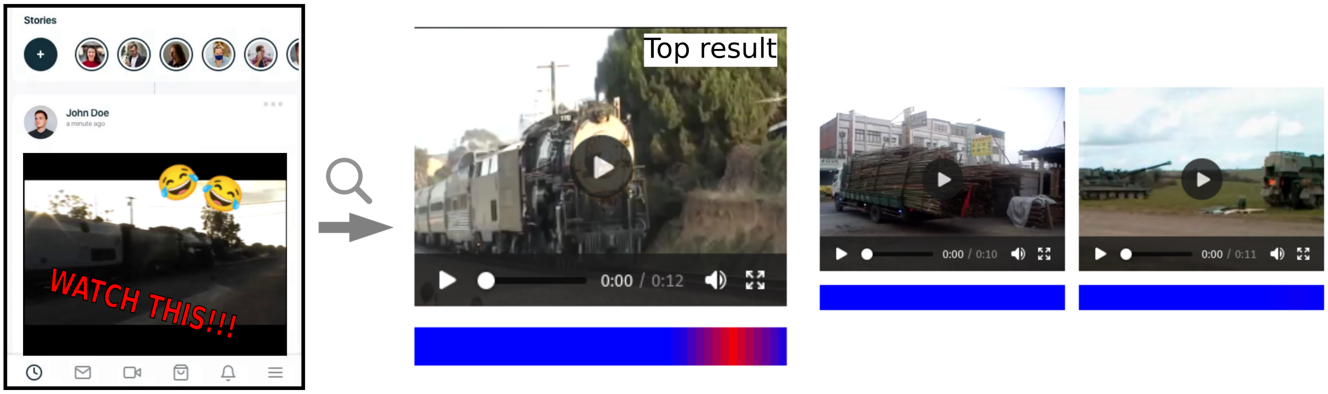

Figure 1: Videos circulating on social media are matched to a trusted database containing media provenance information, enabling the consumer to make informed trust decisions on the content. Matching is robust to audio-visual transformations commonly introduced during redistribution e.g. quality, resolution, or format change as well as partial clips (truncation).

ABSTRACT

We present VPN - a content attribution method for recovering provenance information from videos shared online. Platforms, and users, often transform video into different quality, codecs, sizes, shapes, etc. or slightly edit its content such as adding text or emoji, as they are redistributed online. We learn a robust search embedding for matching such video, invariant to these transformations, using full-length or truncated video queries. Once matched against a trusted database of video clips, associated information on the provenance of the clip is presented to the user. We use an inverted index to match temporal chunks of video using late-fusion to combine both visual and audio features. In both cases, features are extracted via a deep neural network trained using contrastive learning on a dataset of original and augmented video clips. We demonstrate high accuracy recall over a corpus of 100,000 videos.

CCS CONCEPTS

• **Computer methodologies** → **Visual content-based indexing and retrieval**; • **Theory of computation** → *Data provenance.*

KEYWORDS

Video retrieval, media provenance, attribution, content authenticity.

ACM Reference Format:
Alexander Black, Tu Bui, Simon Jenni, Viswanathan Swaminathan, and John Collomosse. 2021. VPN: Video Provenance Network for Robust Content Attribution. In *European Conference on Visual Media Production (CVMP '21), December 6–7, 2021, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485441.3485650

1 INTRODUCTION

Video is a powerful medium for storytelling. Yet, the ease with which digital video may be synthesized, manipulated and shared (*e.g.* via social media) presents a growing societal threat via the amplification of misinformation and fake news [Gregory 2019].

Emerging solutions counter this threat in two main ways: 1) Detecting synthetic or manipulated video [Dolhansky et al. 2020]. Detection methods are caught in a perpetual race against generative AI tools that seek to evade detection. Moreover, digital manipulation is often for editorial rather than deceptive purposes; 2) Determining the '*provenance*' of the video asset [Black et al. 2021; Nguyen et al. 2021]. Provenance techniques do not seek to adjudicate on content validity but instead trace the origins of content, and what was done to it. Rather than automatically deciding on the trustworthiness of

CVMP '21, December 6–7, 2021, London, United Kingdom
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9094-1/21/12…$15.00
https://doi.org/10.1145/3485441.3485650

the content, the goal is to place the consumer in a more informed position to make a trust decision.

This paper falls into the second camp, and complements emerging technical standards that embed cryptographically secured provenance information with the metadata of the asset [Aythora et al. 2020; Coalition for Content Provenance and Authenticity 2021; Rosenthol et al. 2020]. For example, the emerging specification from cross-industry body the 'Coalition for Content Provenance and Authenticity' (C2PA) writes provenance information into a 'manifest' transported within the asset metadata [Coalition for Content Provenance and Authenticity 2021]. Such approaches are vulnerable to removal of metadata, which is common on social media platforms through which misinformation is often spread. For example, video uploaded to any major social media platform today would be stripped of such manifests. Furthermore, alternative manifests may be substituted describing a fake provenance trail or 'back story', so attributing a video out of context to tell a different story. Content misattribution may also deprive creators of credit for their work, enabling intellectual property theft.

This paper contributes a method for robustly matching video assets circulating *without provenance metadata*, to an authoritative copy of that asset with such metadata (such as a C2PA manifest), held within a trusted database. Videos often undergo various transformations during online distribution; changes in format, resolution, size, padding, effect enhancement *etc.* that render cryptographic hashes operating on the binary stream, such as SHA-256, unsuitable as means for matching the video content. Rather, matching must be made robust to these transformations by considering features extracted from the content of the video clip. It is also desirable to match fragments of video (*i.e.* partial or truncated videos) to determine not only the complete source video but also the time offset at which that fragment exist.

We present a system for matching partial video queries robust to such transformations. We build an inverted index of robust audio-visual features trained using contrastive learning and a rich set of augmentations representative of transformations typically applied to video 'in the wild' during online content distribution. We leverage recent work exploring the visual comparison of image matches under similar transformations, and show this technique may be applied to visually compare a query and its matched results to identify areas of content manipulation. We demonstrate our matching technique within the context of a system for tracing the provenance of video assets, using a corpus of 100,000 videos from the VGGSound dataset [Chen et al. 2020b].

2 RELATED WORK

Video content authenticity has been explored primarily from the digital forensics perspective of detecting generative (or 'deep fake') content, or identifying if and where video has been digitally tampered. The Deep Fake Detection Challenge (DFDC) [Dolhansky et al. 2020] explored detection of facial manipulation in video. Networks may be trained to characterize departures from natural image statistics [Wang et al. 2020], or artifacts introduced can be recognized to localize manipulated regions [Wang et al. 2019], in some cases even identifying which GAN synthesized an image [Yu et al. 2021]. Detection of video manipulation similarly exploits temporal anomalies [Wu et al. 2019] or GAN limitations such as lack of blinking [Li et al. 2018]. Forensic models are in a perpetual 'arms-race' with the rapid innovation of generative methods seeking to evade detection.

Attribution methods trace the provenance of media – where it came from, and what has been done to it. A robust identifier, such as a *watermark* [Baba et al. 2019; Devi et al. 2019; Hameed et al. 2006; Profrock et al. 2006], may be embedded in media or a perceptual hash or *fingerprint* is computed from the media content itself [Bharati et al. 2021; Cao et al. 2017; Khelifi and Bouridane 2017; Liu et al. 2016; Moreira et al. 2018; Pan 2019; Zhang et al. 2020]. The identifier serves as a key to lookup provenance information in a provenance database operated by a trusted source [Aythora et al. 2020], or in a decentralized trusted database such as a blockchain [Bui et al. 2020; Collomosse et al. 2018].

In the audio domain, music information retrieval (MIR) has been effective in matching audio or audio fragments at scale [Choi et al. 2018; Downie 2003]. Classical approaches analyze frequency distributions, e.g. short-time Fourier Transform (STFT) [Grill and Schluter 2015], Mel-spectrogram variants such as MFCC [Zheng et al. 2001], or Constant-Q transform [Humphrey and Bello 2012], to identify music fragments. More recently, convolutional neural networks (CNNs) have been applied to classify spectrogram representations of audio for instrument recognition [Han et al. 2017], on-set detection [Schluter and Bock 2013], music tagging at scale [Bertin-Mahieux et al. 2011; Dieleman and Schrauwen 2014; Lee et al. 2017] and genre recognition [Lee et al. 2009; Tzanetakis and Cook 2002]. Music similarity has been explored using feature embeddings regressed via CNN [Lu et al. 2017]. Recurrent neural networks (RNN) have also been applied to integrate such features for tagging [Choi et al. 2017], transcription [Rigaud and Radenen 2016; Sigtia et al. 2015] and hit song prediction [Yang et al. 2017].

In the image domain, visual similarity algorithms focus upon compact, robust perceptual hashing [Wang et al. 2004]. Early approaches compute content fingerprints from coefficients in the spectral domain [Jacobs et al. 1995; Pan 2019; Zauner 2010]. CNNs are now the dominant method for image similarity. Deep Hashing Networks (DHNs) [Zhu et al. 2016] extended an ImageNet-trained AlexNet [Krizhevsky et al. 2012] feature encoder [Hadsell et al. 2006] with a quantization loss, to obtain hashes that retained semantic discrimination. Deep Supervised Hashing (DSH) [Liu et al. 2016] and HashNet [Cao et al. 2017] also train CNNs to learn visual hashes, using a siamese network and ranking loss; such losses are better suited to retrieval than classification tasks, and are used extensively in visual search [Gordo et al. 2016]. DSDH [Li et al. 2017] learns metric ranking and classification directly from the hash code using a discrete cyclic coordinate descend technique. DPSH [Li et al. 2016] proposes a binary pairwise loss to estimate image pair similarity. DFH [Li et al. 2019] and GreedyHash [Su et al. 2018] use different customized layers to address the ill-posed gradient of the sign function. ADSH [Jiang and Li 2018] and CSQ [Yuan et al. 2020] treat hashing as a retrieval/attribution optimization problem, but require semantic annotation unavailable in the content attribution problem domain.

Visual similarity for attribution and content provenance was explicitly addressed by Nguyen *et al.* [Nguyen et al. 2021] who learn a compact binary hash robust to non-editorial transformation, but sensitive to digital manipulation in the image. The goal was to match robustly but intentionally fail to match manipulated content. Conversely, Black *et al.* [Black et al. 2021] learn a fingerprint robust to both non-editorial and editorial change, opting instead to highlight regions of likely manipulation once matched. In both cases provenance information for the asset is surfaced after the match. Our work explores a similar matching strategy for video, and explores change visualization via [Black et al. 2021] over frames. Our work combines robust matching of both audio and visual features, exploring appropriate fusion strategies and indexing to support partial asset (video fragment) matching. We also train our features under data augmentations recently proposed for the task of image similarity for content attribution [Douze et al. 2021].

3 METHODOLOGY

To learn a robust video fingerprinting model, we propose a self-supervised network capable of encoding both visual and audio streams in a video. We leverage contrastive learning and a rich set of data augmentations for videos to train our model. To enable partial video matching, we follow a 'divide and conquer' approach where video is split into chunks and each chunk is indexed and search-able within an inverted index. Our retrieval pipeline is shown in Figure 2. We describe our encoding model in sec. 3.1, our chunking, indexing method and ranking strategy in sec. 3.2, and our audio-visual fusion method in sec. 3.3.

3.1 Learning robust visual-audio features

A video circulated on the Internet may undergo certain transformations that affect either the visual or audio stream or both. For example, the visual stream may be subjected to quality reduction (noise, blur, pixelization,...) during reformatting, change in aspect ratio/geometry (padding, resize, rotation, flip,...), visual enhancement (brightness, color adjustment,...) or editorial change (text/emoji overlay, photoshop,...). Similarly, the audio stream could also be altered (compression, background noise, trimming, effect enhancement, ...). We treat such transformations as *perturbations* to the original video and propose to learn a frame-level visual model and an audio model to encode the respective video streams robust to the defined perturbations.

3.1.1 Learning frame-level visual features. We train a CNN model $f^v(.)$ to project a video frame to a compact embedding space. We employ the ResNet50 architecture [He et al. 2016], replacing the final classifier layer with a 256-D fully connected (fc) layer that serves as the embedding. Our model is trained with loss:

$$\mathcal{L}(z) = -\log \frac{e^{d(z,\bar{z}_+)/\tau}}{e^{d(z,\bar{z}_+)/\tau} + \sum_{z_-} e^{d(z,z_-)/\tau}} \tag{1}$$

$$\text{where } d(u,v) = \frac{g(u) \cdot g(v)}{|g(u)|\,|g(v)|} \tag{2}$$

where z is the embedding of a video frame v: $z = f^v(v) \in \mathbb{R}^{256}$; $\bar{z}_+$ is the average embedding of all transformations of v in the mini-batch; z_- denotes other frame instances; $g(.)$ is a set of two MLP layers separated by ReLU that acts as a buffer between the embedding and the loss function; $d(u,v)$ measures the cosine similarity between the intermediate embeddings $g(u)$ and $g(v)$; τ is the contrastive temperature ($\tau = 0.1$ in our experiments). $\mathcal{L}(.)$ aims to bring the embeddings of all transformations of an image (frame) together, while pushing away other image instances. $\mathcal{L}(.)$ resembles NTXent loss [Chen et al. 2020a], however eq. 1 accepts multiple positives in a batch instead of just a single pair of image augmentations.

We initialize our model with weights from DeepComp [Black et al. 2021]. During training, we randomly sample frames from the training videos to construct a batch. For each frame image we create p augmentations to serve as positive samples (to compute $\bar{z}_+$), while the rest in the batch acts as negatives (the denominator term in eq. 1). We empirically set $p = 3$ for optimal performance (improves by 1% as compared with standard NTXentLoss). We observe that larger p causes a drop in performance, probably because the number of unique images in the batch must be reduced accordingly in order to fit a GPU.

We use an exhaustive list of frame-level augmentations for our training, including random Noise (variance 0.01), Blur (radius $[0, 10]$), Horizontal Flip, Pixelization (ratio $[0.1, 1.0]$), Rotation ($[-30, +30]$ degrees), random Emoji Overlay (opacity $[80, 100]\%$, size $[10, 30]\%$, random position), Text Overlay (text length $[5, 10]$, size $[10, 20]\%$, random characters, typeface and position), Color Jitter (brightness, contrast and saturation $[0.6, 1.4]$), Padding ($[0, 25]\%$ dimension, random color). These are on top of 16 Imagenet-C [Hendrycks and Dietterich 2019] transformations seen by the pretrained model [Black et al. 2021] which our model initialises from. Since our model operates on individual video frames, all transformations are applied at frame level, *i.e.*, the temporal coherence between frames are ignored during data augmentation. However, at test time our query videos are transformed at video level to reflect the video editing and distribution practice. Fig. 4 illustrates several benign transformations applied to an example video frame.

3.1.2 Learning audio features. We split the audio signal of a video into overlapping 1-second chunks and encode it via log mel-spectrogram. We then visualize the log mel-spectrogram as a 2D RGB image and treat it as input to our audio model $f^a(.)$. The same model architecture and loss as in sec. 3.1.1 are employed for our training, but using a different set of data augmentation methods to learn robust audio features. In general, the benign audio transformations can be categorized in two groups - those that lengthen or shorten the audio signal and those that add, remove or alter audio components. The former includes audio Clipping ($[0, 20]\%$ audio length) and Time Stretching (slow down 0.5x - speed up 1.5x). The latter includes Adding Clicks (random click rate 0.5sec - full length), Adding Background Noise (SNR 5db), Filtering Harmonics (margin $[1.0, 1.5]$), Filtering Percussive (margin $[1.0, 1.5]$) and Pitch Shifting (semitone $[-5, 5]$). These transformations are commonly encountered during audio redistribution and editing practice. Fig. 3 shows the effects of these transformations on an example audio segment.

3.2 Chunking and Indexing

To index a variable length video X, we propose to split X into fixed-length chunks $X = \{x_i | i = 1, 2, ..., N\}$ $s.t.\ len(x_i) = len(x_j) = l \quad \forall i, j \in [1, N]$ where $len(.)$ is the length function (in seconds), constant l is the chunk length (l=10sec) and N is number of chunks (the last chunk is padded if necessary). We split videos into chunks in a sliding window fashion with chunk stride $s_c \leq l$, thus $N =$

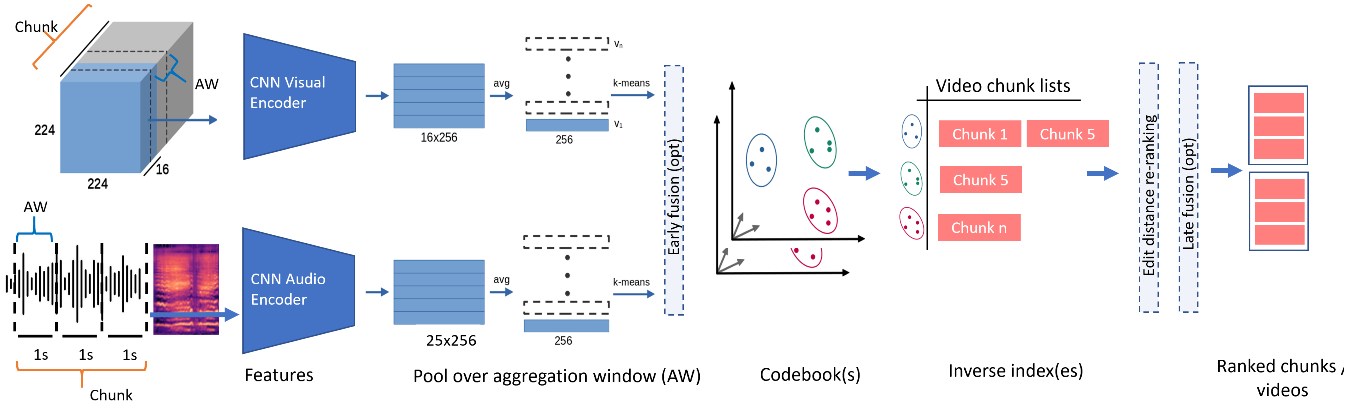

Figure 2: Architecture of the proposed system. Video is regularly sub-divided into equal-length chunks, in which a set of visual and audio descriptors is computed for each chunk. These independently computed features are regularly sampled from short temporal aggregation windows (AW) within the chunk. The visual and audio features are quantised to create a dictionary upon which an inverted index is built to index videos at chunk-level. We explore 3 fusion strategies for video and audio retrieval, that differ in which stage the audio and video features are combined. A re-ranking on edit distance yields the ranked list of video chunks returned to the user.

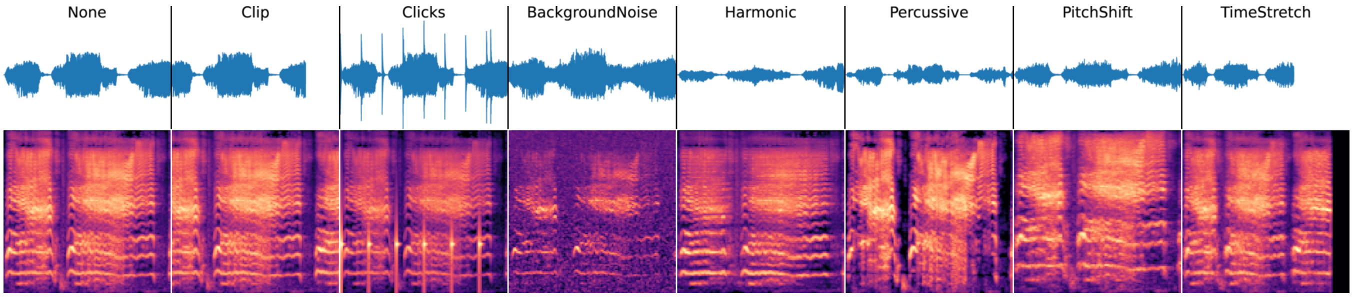

Figure 3: Audio augmentations: Magnitude (top) and mel-spectrogram (bottom) of an audio segment and random benign transformations used during training the audio feature encoder.

$\lceil len(X)/s_c \rceil$. We use chunks as an atomic unit in our approach where a 'bag of features' is computed for each chunk, also indexing and search are performed at chunk level. We use $s_c = l/2$ in our experiments.

3.2.1 Computing chunk features. Given a video chunk $x_i = \{x_i^v, x_i^a\}$ containing a visual stream x_i^v and an audio stream x_i^a of the same length l, we feed the two streams into the respective visual (sec.3.1.1) and audio (sec.3.1.2) models, obtaining a set of descriptors for both streams. For the visual stream, we sample x_i^v at 16fps with stride s_f ($s_f = 0.5$ second or 8 frames), extract and average CNN features on every 16-frame aggregation window (AW) to get one visual descriptor per second.

$$z_i^v = \mathcal{F}^v(x_i^v) \in \mathbb{R}^{n \times 256}, \quad n = \lceil \frac{l}{s_f} \rceil \tag{3}$$

where $\mathcal{F}^v$ is our aggregation function, at sampling point j $\mathcal{F}^v(x_{i,j}^v) = \frac{1}{16}\sum_{t=0}^{15} f^v(x_{i,j+t}^v)$; n is number of visual descriptors per chunk.

For the audio stream, since our audio model has input size of 1 second audio length, we sample x_i^a at 1 second interval with the same stride s_f as in the visual model.

$$z_i^a = f^a(x_i^a) \in \mathbb{R}^{n \times 256} \tag{4}$$

This makes our audio extraction in sync with the visual extraction process (both have an aggregation window of 1 second), resulting in the same number of audio and visual descriptors per video chunk x_i. This setting also offers a level of flexibility to our audio-fusion strategy (sec.3.3).

3.2.2 Inverted index. Similar to text search systems, we construct an inverted index that supports video retrieval at chunk-level. We sample 1M random descriptors (audio, visual or fusion, more details in sec. 3.3) and build a dictionary with codebook size K using KMeans [Lloyd 1957]. Given a database video, we break it into chunks where each chunk is represented as a bag of codewords. The K codewords are used as entries to our inverted index, listing all chunks in the database (a mapping between a chunk and video ID is also stored).

The relevance of a query chunk $q = \{q_1, q_2, ..., q_n\}$ to a database chunk x_i is determined by a *relevance score* $\mathcal{R}$, defined as:

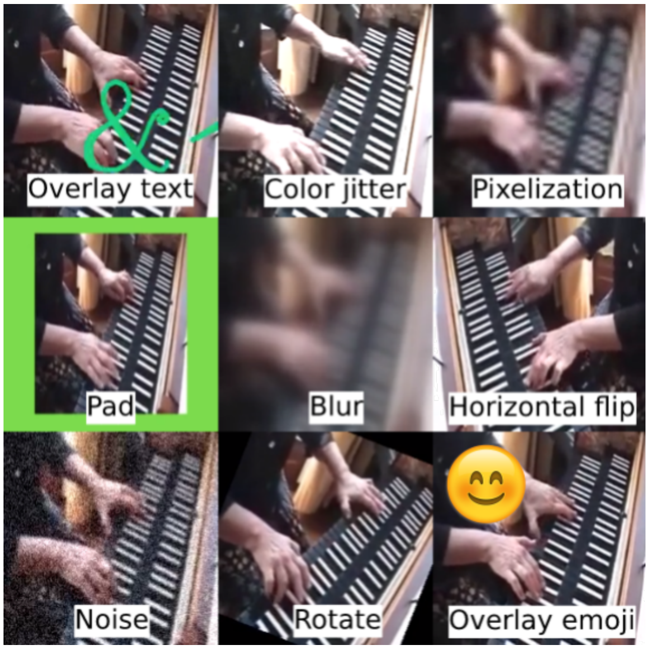

Figure 4: Visual augmentation: examples of random benign augmentations applied to a video frame during contrastive training of the visual feature encoder.

Table 1: Visual performance

Method	R@1	R@10	R@100
Ours	0.973	0.980	0.989
DeepComp [Black et al. 2021]	0.922	0.945	0.963
ResNet50 [He et al. 2016]	0.238	0.314	0.411

$$\mathcal{R}(q, x_i) = \sum_{t=1}^{n} tf(q_t, x_i) \times idf(q_t) \tag{5}$$

where $tf(q_t, x_i)$ is the *term frequency* that denotes the number of times codeword q_t appears in the video chunk x_i and $idf(q_t)$ is the *inverse document frequency*, which measures how common is q_t across all of the chunks in the dataset.

Finally, the relevance of a query video Q to a database video X is defined as:

$$\mathcal{R}(Q, X) = \sum_{q}^{Q} \sum_{x_i}^{X} \mathcal{R}(q, x_i) \tag{6}$$

3.2.3 Re-ranking. Inverted index does not take into account the order of descriptors within a chunk, and chunk order within a video. We therefore employ TF-IDF [Jones 1972] to retrieve top-k candidate videos (k=200) before performing an additional re-ranking stage based on Levenshtein distance [Levenshtein et al. 1966] which quantifies the similarity between two sequences by counting the number of *edits* (insertions, deletions or substitutions) required to turn one sequence into the other. Sorting by edit distance promotes to the top videos in which visual words appear in the order that matches the query the closest.

Table 2: Audio performance

Method	R@1	R@10	R@100
Ours	0.826	0.899	0.969
AVENet (finetuned)	0.572	0.688	0.728
AVENet [Chen et al. 2020b]	0.530	0.626	0.671
VGGish [Shawn et al. 2017]	0.492	0.579	0.610

3.2.4 Partial video matching and localization. An advantage of our chunking and inverted index method is that it enables retrieval even if the query is only a segment of a database video. As a by product, our method also supports localization of a video segment by searching for the closest chunk in the database, or even the closest codeword within a chunk for more fine-grained localization.

3.3 Audio-visual fusion

We explore 3 fusion approaches for audio and visual streams to create an unified video retrieval system.

Early fusion - a single codebook is constructed for $z = [z_i^v, a_i^v] \in \mathbb{R}^{512}$, which is a concatenation of the visual and audio descriptor of a single aggregation window.

Late fusion - we build separate codebooks and inverted indices for audio and visual domains, but compute the relevance score in eq. 6 and re-ranking jointly.

Learned fusion - we learn an unified audio-visual embedding for a video AW. Since the visual model $f^v(.)$ operates at frame-level, we average the embeddings of the frames within an AW to represent the visual feature of that AW, before concatenating with the audio feature and projecting to the unified embedding:

$$z = E_p([\frac{1}{|AW|} \sum_{x^v}^{AW} f^v(x^v), f^a(x^a)]) \in \mathbb{R}^{256} \tag{7}$$

where E_p is a fc layer for dimensional reduction; [,] is a concatenation; $|AW|$ is number of frames in an AW (to make the model small, we sample video at 4fps, thus $|AW| = 4$). To train our model, we first train the audio and visual models separately, then use their weights to initialize our joint model training.

4 RESULTS AND DISCUSSION

4.1 Datasets and evaluation metrics

We use the following 2 datasets for training and evaluation:

(1) **VGGSound** [Chen et al. 2020b] is a large audio dataset with 160k train and 14k test videos. All videos are of 10 seconds length of 310 audio classes, originally used for audio classification. We train all of our models using the public VGGSound train set without using class annotations. To create a test set for retrieval, we first construct a 100k video database from the 14k test videos plus training videos to serve as distractors. Next, we sample 1k random videos from the test set and apply random transformations on both audio and video streams in order to make the query set. For experiments with query length, we randomly clip the queries keeping a

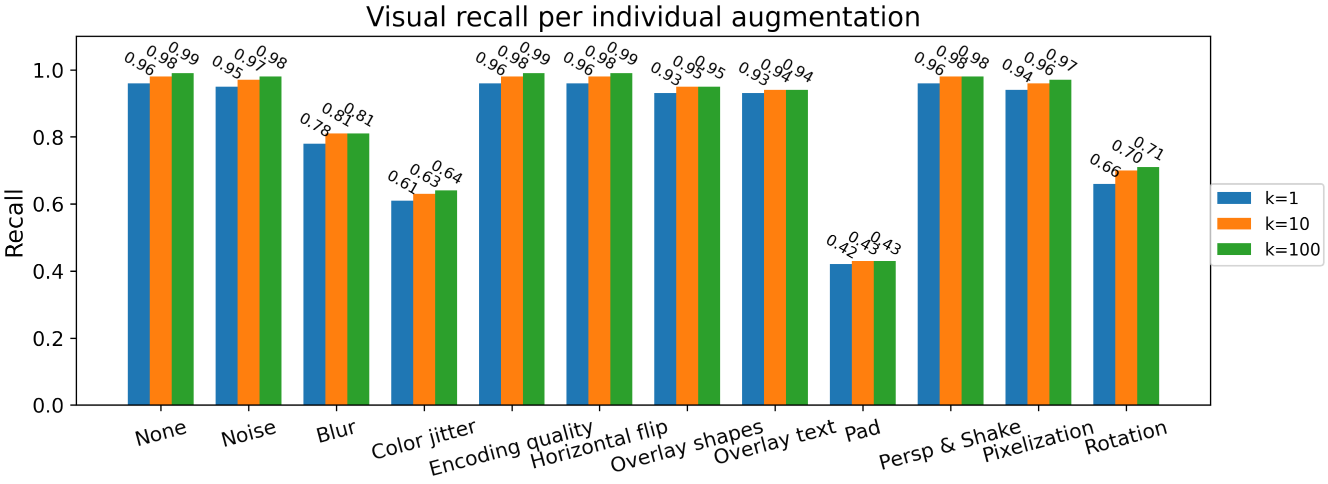

Figure 5: Visual retrieval performance breakdown per individual augmentations applied to the query, showing recall at k = 1, 10, 100.

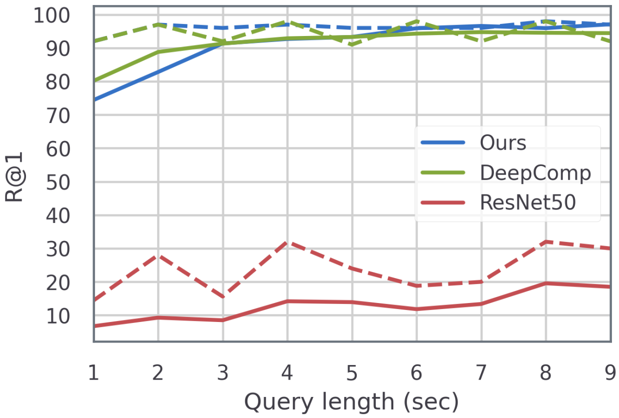

Figure 6: Visual retrieval performance showing R@1 and R@100 (dashed) in % vs query segment length. Evaluated on VGGSound dataset.

minimum 10% length (1 second). We use this dataset for the majority of our experiments.

(2) **DFDC-Preview** [Brian et al. 2019] consists of 5k original and manipulated videos. The dataset has a public train/test split and all videos have 15 seconds length. We employ DFDC-Preview to evaluate our model generalization to new domains (people and human voice) and unseen transformations (face swap manipulations). We do not retrain our model on DFDC-Preview, instead we evaluate our VGGSound-trained model directly on it. We use all 1131 real videos in DFDC-Preview together with distracting videos from VGGSound to make a 100k database. We create two query sets - a *real* set containing 276 real test videos that are pre-applied transformations with reduction in fps, resolution and encoding quality; and a *fake* set containing 501 face-swap manipulated videos. We wish to attribute the queries to the original videos in the database regardless of it being real or fake.

Table 3: Fusion performance

Method	R@1	R@10	R@100
Late fusion	0.982	0.991	0.996
Learned fusion	0.941	0.949	0.956
Early fusion	0.789	0.842	0.913

Since there is only 1 relevant video in the database for each query, we use recall at k ($R@k$) to evaluate the retrieval performance of our approach. R@k represents the ratio of queries for which the correct video was retrieved within top k returned results.

4.2 Experiment settings

We train our models using the Pytorch library, with Adam optimizer, initial learning rate 0.0005 with step decay and early stopping schedule. We use a batch size of 32 unique visual or audio samples where each sample is randomly augmented 3 times (totally 96, sec. 3.1.1-3.1.2). For chunking and indexing, since both VGGSound and DFDC-Preview videos are short and have equal length, we set the chunk size the same as the video length (note that our settings in sec. 3.2 allow indexing and querying of arbitrary-length videos). For inverted index, we set the number of entries equal 100k, which is also the codebook size for both visual and audio modalities.

4.3 Evaluating visual features

We compare our method with two following baselines:

- **ResNet50 (ImageNet)**[He et al. 2016] - a generic model pretrained on ImageNet for image classification. We use the 2048-D output of the average pooling layer for our embedding.
- **DeepComp**[Black et al. 2021] - a Resnet50-based model that is trained to be robust against ImagenetC[Hendrycks and

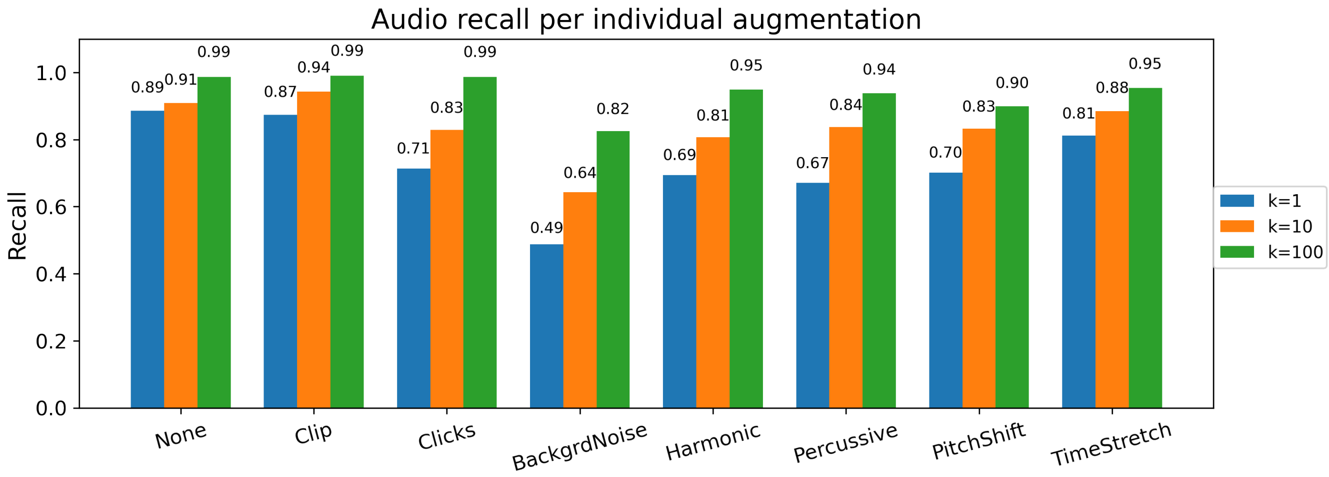

Figure 7: Audio performance on individual augmentations.

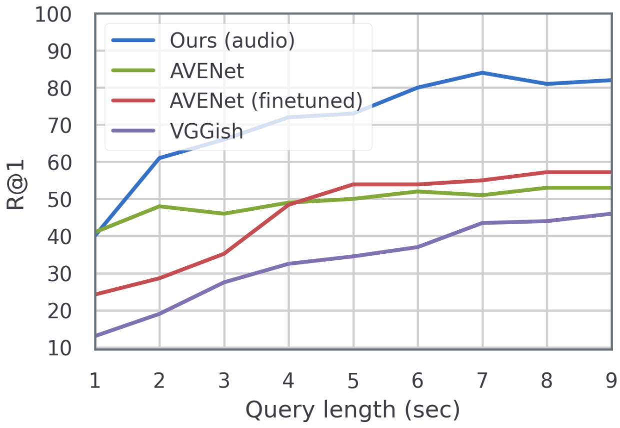

Figure 8: Audio performance on query segment length.

Dietterich 2019] perturbation as well as photoshop manipulation.

While changing the backbone network, all other parameters, such as codebook size and search algorithm are fixed. Table 1 shows the recall performance of our approach and the baselines on the VGGSound dataset. Although all 3 models have the same CNN architecture, ResNet50 (ImageNet) performs the worst since it is not exposed to the video transformations (sec. 3.1.1) during training. DeepComp [Black et al. 2021] has significantly better performance because it is trained to be robust against a number of perturbations on images. Finally, the proposed approach outperforms the rest at all recall levels. This demonstrates that robustness against particular perturbations could be achieved by exposing the model to the same augmentations during training.

The partial matching performance on visual stream is depicted in Figure 6. Overall, the retrieval is more challenging with shorter queries. The proposed method slightly underperforms DeepComp [Black et al. 2021] at R@1 on very short queries, but outperforms on queries longer than 3 seconds (30% length of the original videos) and always performs favorably at R@10 on all query lengths. In contrast, ResNet50 [He et al. 2016] performs poorly again, with R@1 (R@10) ranges between 10-20% (15-30%) as the query length increases.

Table 4: DFDC search performance

Query set	R@1	R@10	R@100
Real	0.918	0.972	0.999
Fake	0.912	0.966	0.998

We report the robustness of our visual model against individual transformations in Figure 5. Overall, there is not much difference between top-1 and top-100 recall, thanks to the re-ranking step (as demonstrated later in sec. 4.7). Also, our model is most robust against common degradation (noise, encoding quality, pixelization) and minor content editing (text/shape overlay) with above 90% recall at all levels, while being more sensitive to padding, rotation and color jitter.

4.4 Evaluating audio features

We compare our method with 3 following baselines:

- **VGGish** [Shawn et al. 2017] - a VGG-based model trained on 70M audio clips to classify 3k audio categories. This model produces a 128-D descriptor per 0.96ms AW.
- **AVENet** [Chen et al. 2020b] - a ResNet18-based model released with the VGGSound dataset. This model produces a 512-D descriptor per 10-second audio.
- **AVENet (finetuned)** is the AVENet model finetuned using our data augmentations and loss. We keep the same model architecture and data processing but adopt our training procedure for fair comparison.

The recall performance is reported in Table 2. Our method outperforms the rest with a significant margin (21-25% consistent

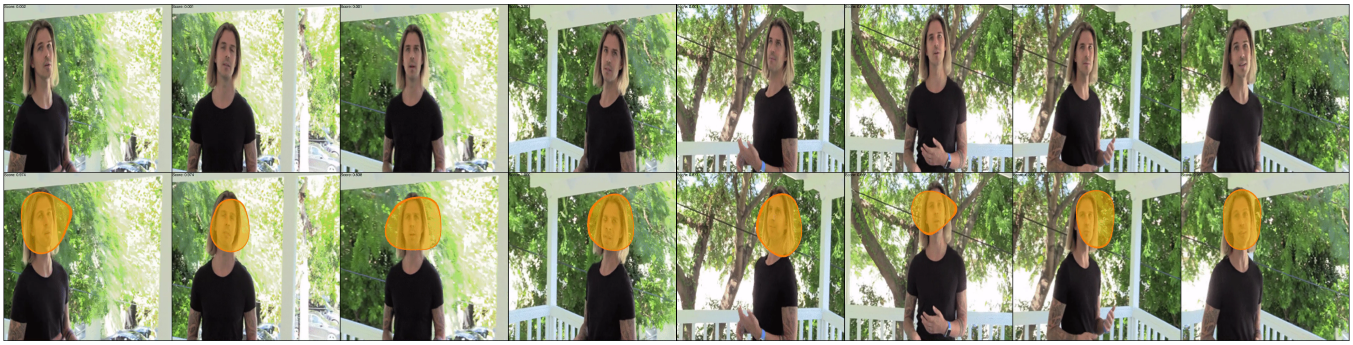

Figure 9: Using DFDC fake video (bottom) as query, we are able to retrieve the original (top) and use ICN[Black et al. 2021] to visualize the manipulated areas.

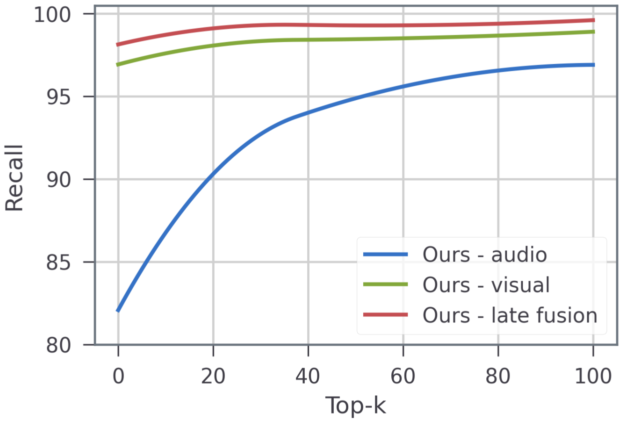

Figure 10: Recall curve in % versus top-k of our audio, visual and late fusion models.

improvement from the nearest competitor at all recall levels). Our finetuned version of AVENet also has better performance than other pretrained models, while VGGish [Shawn et al. 2017] performs the worst.

The similar trend is observed in our experiment with query length (Figure 8). Our method leads a 20% gap ahead of AVENet and AVENet (finetuned) at all query lengths, except at 1-second queries where its R@1 score levels with AVENet.

Figure 7 shows the effects of individual audio perturbations on the recall performance of the proposed method. Our model is most sensitive to background noise while being less affected by other sources (c.f. Figure 3).

4.5 Audio-visual fusion

We evaluate 3 fusion methods described in sec. 3.3. Table 3 indicates the superiority of the late fusion method, with 98% recall at top-1 and near perfect performance at top-100. Learned fusion method has better score than early fusion but lower than late fusion, probably because the unified embedding and inverted index are effectively twice as compact as the combined audio and visual. Another advantage of late fusion is that it enables querying of individual modality, for example in case an user only has single-modal data or prefer to retrieval an individual stream. Figure 10 demonstrates that audio and visual retrieval has complementary effects, as the late fusion method improves performance versus any single-stream retrieval method.

4.6 Video attribution on DFDC-Preview

We employ our VGGSound-trained model together with the best fusion method (late fusion) and test directly on the DFDC-Preview dataset. Despite the domain shift, our method performs extremely well on the public augmented-ready *real* set and the manipulated *fake* set whose transformations are unseen during training, as shown in Table 4. On both sets we achieve above 91% R@1 and the performance is on par with the VGGSound results at R@100 (c.f. Table 3 late fusion), demonstrating the generalization of our approach.

For the *fake* set, we further visualize the manipulated area using the ICN method proposed in [Black et al. 2021]. Figure 9 shows the visualization between an example query and the top-1 retrieval video. This is useful in practice for a user to be able to attribute a query video back to its original source, localize its position in the source and highlight any possibly manipulated regions.

4.7 Ablation study

We test the efficacy of our retrieval pipeline when stripping off one or several components (Table 5). We first turn off the re-ranking stage (sec. 3.2.3) and rank the results using only the TF-IDF relevance score. Without re-ranking the recall score significantly drops by 18% at R@1. This indicates that re-ranking can promote the relevant video to the top of the ranking by leveraging the temporal sequence of codewords within a chunk (and sequence of chunks within a long video). Next, we further turn off TF-IDF and use only the histogram count of codewords in the inverted index to rank the videos. The performance further reduces by 3% at R@1.

4.8 Localization of video segments

We demonstrate an example of partial matching and localization of a video segment in Fig. 11. Since each video can be represented as a variable-length sequence of chunks and each chunk is a sequence of codewords, we can compute the similarity between the query

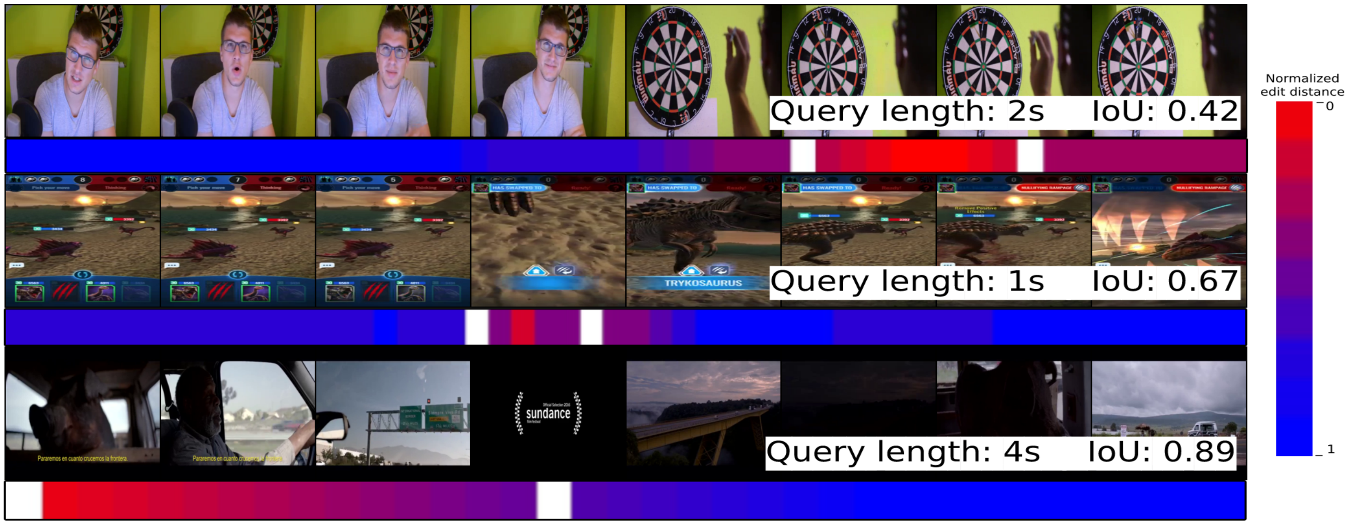

Figure 11: Localization of query clips within the top-1 retrieved video. For each row, the query clip is the video segment within the white bars of the original video (also subjected to random augmentations). In all 3 cases the original video is returned at top-1. The heatmap bar shows the edit distance between the query sequence of codewords and a same-length segment of the candidate video in sliding window fashion, which could be used to represent the confidence in localization of the query within the candidate video. IoU score can be computed over a thresholded heatmap and the ground truth location.

Table 5: Ablation study when re-ranking with edit distance and TF-IDF is turned off.

Method	R@1	R@10	R@100
proposed	0.982	0.991	0.996
w/o re-ranking	0.798	0.984	0.995
w/o TF-IDF + re-ranking	0.764	0.982	0.992

sequence with a sequence of a candidate video using edit distance, thus be able to localize position of the query in the candidate video (at chunk-level for long videos and codeword-level for short videos).

5 CONCLUSION

We present a robust video provenance system capable of matching full or partial videos circulating online without provenance metadata, to an authoritative copy of that video held within a trusted database. Our matching process is robust to transformations of the video such as changes in format, resolution, size, padding that video often undergoes during online redistribution. We employ self-supervised contrastive learning to learn audio and visual features robust to such transformations, and match these using a vector quantization and inverse index. We explored the relative benefits of early versus late fusion for accommodating both modalities, and proposed an edit-distance based re-ranking process for shortlisted results. For feature learning we showed that recently proposed augmentation sets for image similarity extend well to video for the authenticity use, and that recent techniques for visualizing manipulated regions in images may be effectively applied to video on a per-frame basis.

Future work could explore integration of our feature embeddings with emerging standards [Aythora et al. 2020; Coalition for Content Provenance and Authenticity 2021; Rosenthol et al. 2020] and decentralized frameworks such as blockchain to enable decentralized content registries for tracing videos distributed online. For example, in the draft technical specification of the cross-industry C2PA body [Coalition for Content Provenance and Authenticity 2021] it is recognised that metadata will frequently be stripped from assets and that a 'soft binding' (such as the proposed video matching solution) may be used re-unite assets with their originals in a trusted 'provenance datastore'. Since this implies a centralization of trust, it may be interesting to store this database in a decentralized fashion. Such approaches are already being explored for example in the domain of archives and memory institutions [Bui et al. 2020; Collomosse et al. 2018].

ACKNOWLEDGMENTS

We thank Andy Parsons, Leonard Rosenthol and the Adobe Content Authenticity Initiative (CAI) for discussions. This work was supported by Adobe Research via a PhD studentship and via the DECaDE Centre by EPSRC Grant Ref EP/T022485/1.

REFERENCES

J. Aythora, R. Burke-Agüero, A. Chamayou, S. Clebsch, M. Costa, J. Deutscher, N. Earnshaw, L. Ellis, P. England, C. Fournet, et al. 2020. Multi-stakeholder Media Provenance Management to Counter Synthetic Media Risks in News Publishing. In *Proc. Intl. Broadcasting Convention (IBC)*.

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

T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. 2011. The million song dataset. In *Proc. Intl. Conf of Soceity for Music Infomation Retrieval.* 591–596.

A. Bharati, D. Moreira, P.J. Flynn, A. de Rezende Rocha, K.W. Bowyer, and W.J. Scheirer. 2021. Transformation-Aware Embeddings for Image Provenance. *IEEE Trans. Info. Forensics and Sec.* 16 (2021), 2493–2507.

A. Black, T. Bui, H. Jin, V. Swaminathan, and J. Collomosse. 2021. Deep Image Comparator: Learning To Visualize Editorial Change. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 972–980.

D. Brian, H. Russ, P. Ben, B. Nicole, and F. C. Canton. 2019. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854* (2019).

T. Bui, D. Cooper, J. Collomosse, M. Bell, A. Green, J. Sheridan, J. Higgins, A. Das, J. Keller, and O. Thereaux. 2020. Tamper-proofing Video with Hierarchical Attention Autoencoder Hashing on Blockchain. *IEEE Trans. Multimedia (TMM)* 22, 11 (2020), 2858–2872. https://doi.org/10.1109/TMM.2020.2967640

Z. Cao, M. Long, J. Wang, and P. S. Yu. 2017. Hashnet: Deep learning to hash by continuation. In *Proc. CVPR*. 5608–5617.

H. Chen, W. Xie, A. Vedaldi, and A. Zisserman. 2020b. VGG-Sound: A large scale audio-visual dataset. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proc. ICML*. 1597–1607.

K. Choi, G. Fazekas, K. Cho, and M. Sandler. 2018. A Tutorial on Deep Learning for Music Information Retrieval. *arXiv:1709.04396v2* (2018).

K. Choi, G. Fazekas, M. Sandler, and K. Cho. 2017. Convolutional recurrent neural networks for music classification. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Coalition for Content Provenance and Authenticity. 2021. *Draft Technical Specification 0.7.* Technical Report. C2PA. https://c2pa.org/public-draft/

J. Collomosse, T. Bui, A. Brown, J. Sheridan, A. Green, M. Bell, J. Fawcett, J. Higgins, and O. Thereaux. 2018. ARCHANGEL: Trusted Archives of Digital Public Documents. In *Proc. ACM Doc.Eng.*

P. Devi, M. Venkatesan, and K. Duraiswamy. 2019. A Fragile Watermarking scheme for Image Authentication with Tamper Localization Using Integer Wavelet transform. *J. Computer Science* 5, 11 (2019), 831–837.

S. Dieleman and B. Schrauwen. 2014. End-to-end learning for music audio. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. 2020. The DeepFake Detection Challenge (DFDC) Dataset. *CoRR* abs/2006.07397 (2020). arXiv:2006.07397 http://arxiv.org/abs/2006.07397

M. Douze, G. Tolias, E. Pizzi, Z. Papakipos, L. Chanussot, F. Radenovic, T. Jenicek, M. Maximov, L. Leal-Taixé, I. Elezi, O. Chum, and C. C. Ferrer. 2021. The 2021 Image Similarity Dataset and Challenge. *CoRR* abs/2106.09672 (2021). arXiv:2106.09672 http://arxiv.org/abs/2106.09672

J. S. Downie. 2003. Music Information Retrieval. *Annual review of information science and technology* 37, 1 (2003), 295–340.

A. Gordo, J. Almazán, J. Revaud, and D. Larlus. 2016. Deep image retrieval: Learning global representations for image search. In *Proc. ECCV*. 241–257.

S. Gregory. 2019. *Ticks or it didn't happen.* Technical Report. Witness.org.

T. Grill and J. Schluter. 2015. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *Proc. EUSPICO*.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*. 1735–1742.

K. Hameed, A. Mumtax, and S. Gilani. 2006. Digital image Watermarking in the wavelet transform domain. *WASET* 13 (2006), 86–89.

Y. Han, J. Kim, and K. Lee. 2017. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE Trans. Audio, Speech and Language Processing* 25, 1 (2017), 208–221.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR*. 770–778.

D. Hendrycks and T. Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proc. ICLR*.

E. J. Humphrey and J. P. Bello. 2012. Rethinking automatic chord recognition with convolutional neural networks. In *Proc. Intl. Conf. on Machine Learning and Applications.*

C. Jacobs, A. Finkelstein, and D. Salesin. 1995. Fast multiresolution image querying. In *Proc. ACM SIGGRAPH*. ACM, 277–286.

Q-Y. Jiang and W-J. Li. 2018. Asymmetric Deep Supervised Hashing. In *AAAI*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* (1972).

F. Khelifi and A. Bouridane. 2017. Perceptual Video Hashing for Content Identification and Authentication. *IEEE TCSVT* 1, 29 (2017).

A. Krizhevsky, I. Sutskever, and G. Hinton. 2012. ImageNet Classification with Deep Convnets. In *Proc. NIPS*.

H. Lee, P. Pham, Y. Largman, and A. Y. Ng. 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*.

J. Lee, J. Park, K. L. Kim, and J. Nam. 2017. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789* (2017).

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

Q. Li, Z. Sun, R. He, and T. Tan. 2017. Deep supervised discrete hashing. In *Proc. NeurIPS*. 2482–2491.

W. Li, S. Wang, and W-C. Kang. 2016. Feature learning based deep supervised hashing with pairwise labels. In *Proc. IJCAI*. 1711–1717.

Y. Li, M-C. Ching, and S. Lyu. 2018. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In *Proc. IEEE WIFS*.

Y. Li, W. Pei, and J. van Gemert. 2019. Push for Quantization: Deep Fisher Hashing. *BMVC* (2019).

H. Liu, R. Wang, S. Shan, and X. Chen. 2016. Deep supervised hashing for fast image retrieval. In *Proc. CVPR*. 2064–2072.

SP Lloyd. 1957. Least square quantization in PCM. Bell Telephone Laboratories Paper. Published in journal much later: Lloyd, SP: Least squares quantization in PCM. *IEEE Trans. Inform. Theor.(1957/1982)* 18 (1957).

R. Lu, K. Wu, Z. Duan, and C. Zhang. 2017. Deep ranking: triplet matchnet for music metric learning. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

D. Moreira, A. Bharati, J. Brogan, A. Pinto, M. Parowski, K.W. Bowyer, P.J. Flynn, A. Rocha, and W.J. Scheirer. 2018. Image provenance analysis at scale. *IEEE Trans. Image Proc.* 27, 12 (2018), 6109–6122.

E. Nguyen, T. Bui, V. Swaminathan, and J. Collomosse. 2021. OSCAR-Net: Object-centric Scene Graph Attention for Image Attribution. In *Proc. ICCV*.

T. Pan. 2019. Digital-Content-Based Identification: Similarity hashing for content identification in decentralized environments. In *Proc. Blockchain for Science*.

D. Profrock, M. Schlauweg, and E. Muller. 2006. Content-Based Watermarking by Geometric Wrapping and Feature- Based Image Segmentation. In *Proc. SITIS*. 572–581.

F. Rigaud and M. Radenen. 2016. Singing voice melody transcription using deep neural networks. In *Proc. Intl. Conf. on Music Information Retreival (ISMIR)*.

L. Rosenthol, A. Parsons, E. Scouten, J. Aythora, B. MacCormack, P. England, M. Levallee, J. Dotan, et al. 2020. *Content Authenticity Initiative (CAI): Setting the Standard for Content Attribution.* Technical Report. Adobe Inc.

J. Schluter and S. Bock. 2013. Musical onset detection with convolutional neural networks. In *Proc. Intl. Workshop on Machine Learning and Music (MML)*.

H. Shawn, C. Sourish, E. Daniel PW, G. Jort F, J. Aren, M. R. Channing, P. Manoj, P. Devin, S. Rif A, S. Bryan, et al. 2017. CNN architectures for large-scale audio classification. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 131–135.

S. Sigtia, E. Benetos, and S. Dixon. 2015. An end-to-end neural network for polyphonic music transcription. *arXiv preprint arXiv:1508.01774* (2015).

S. Su, C. Zhang, K. Han, and Y. Tian. 2018. Greedy hash: Towards fast optimization for accurate hash coding in CNN. In *Proc. NeurIPS*. 798–807.

G. Tzanetakis and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Trans. on Audio and Speech Proc.* (2002).

J. Wang, H. T. Shen, J. Song, and J. Ji. 2004. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927* (2004).

S-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. Efros. 2019. Detecting Photoshopped Faces by Scripting Photoshop. In *Proc. ICCV*.

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

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

L.-C. Yang, S.-Y. Chou, J.-Y. Liu, Y.-H. Yang, and Y.-A. Chen. 2017. Revisiting the problem of audio-based hit song prediction using convolutional neural networks. *arXiv preprint arXiv:1704.01280* (2017).

N. Yu, V. Skripniuk, S.r Abdelnabi, and M. Fritz. 2021. Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data. In *Proc. Intl. Conf. Computer Vision (ICCV)*.

L. Yuan, T. Wang, X. Zhang, F. Tay, Z. Jie, W. Liu, and J. Feng. 2020. Central Similarity Quantization for Efficient Image and Video Retrieval. In *Proc. CVPR*. 3083–3092.

C. Zauner. 2010. *Implementation and Benchmarking of Perceptual Image Hash Functions.* Master's thesis. Upper Austria University of Applied Sciences, Hagenberg.

X. Zhang, Z. H. Sun, S. Karaman, and S.F. Chang. 2020. Discovering Image Manipulation History by Pairwise Relation and Forensics Tools. *IEEE J. Selected Topics in Signal Processing.* 14, 5 (2020), 1012–1023.

F. Zheng, G. Zhang, and Z. Song. 2001. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *J. Computer Science and Technology* 16, 6 (2001), 582–589.

H. Zhu, M. Long, J. Wang, and Y. Cao. 2016. Deep hashing network for efficient similarity retrieval. In *Proc. AAAI*.