# Detecting and Mapping Video Impairments

Todd R. Goodall and Alan C. Bovik, *Fellow*, *IEEE*

***Abstract*—Automatically identifying the locations and severities of video artifacts without the advantage of an original reference video is a difficult task. We present a novel approach to conducting no-reference artifact detection in digital videos, implemented as an efficient and unique dual-path (parallel) excitatory/inhibitory neural network that uses a simple discrimination rule to define a bank of accurate distortion detectors. The learning engine is distortion-sensitized by pre-processing each video using a statistical image model. The overall system is able to produce full-resolution space-time distortion maps for visualization, as well as providing global distortion detection decisions that represent the state of the art in performance.**

**Our model, which we call the Video Impairment Mapper (VIDMAP), produces a first-of-a-kind full resolution map of artifact detection probabilities. The current realization of this system is able to accurately detect and map eight of the most important artifact categories encountered during streaming video source inspection: aliasing, video encoding corruptions, quantization, contours/banding, combing, compression, dropped frames, and upscaling artifacts. We show that it is either competitive with or significantly outperforms the previous state-of-the-art on the whole-image artifact detection task. A software release of VIDMAP that has been trained to detect and map these artifacts is available online: http://live.ece.utexas.edu/research/quality/VIDMAP_release.zip for public use and evaluation.**

***Index Terms*—VIDMAP; Artifact Mapping; Detection**

## I. Introduction

Digital video streaming companies such as Netflix, Hulu, and YouTube are actively growing not only the number of subscribers served but also the diversity of video content offered [1], [2], [3]. The need to curate video content grows as video production expands. Production studios that differ in professional capability will invariably produce videos exhibiting various degrees of quality. Since customers have grown to expect high video quality across all platforms, the source videos accepted into video catalogs should be as free as possible from the kinds of distortions that plague new video content. Existing distortions need to be identified and dealt with in a scalable way.

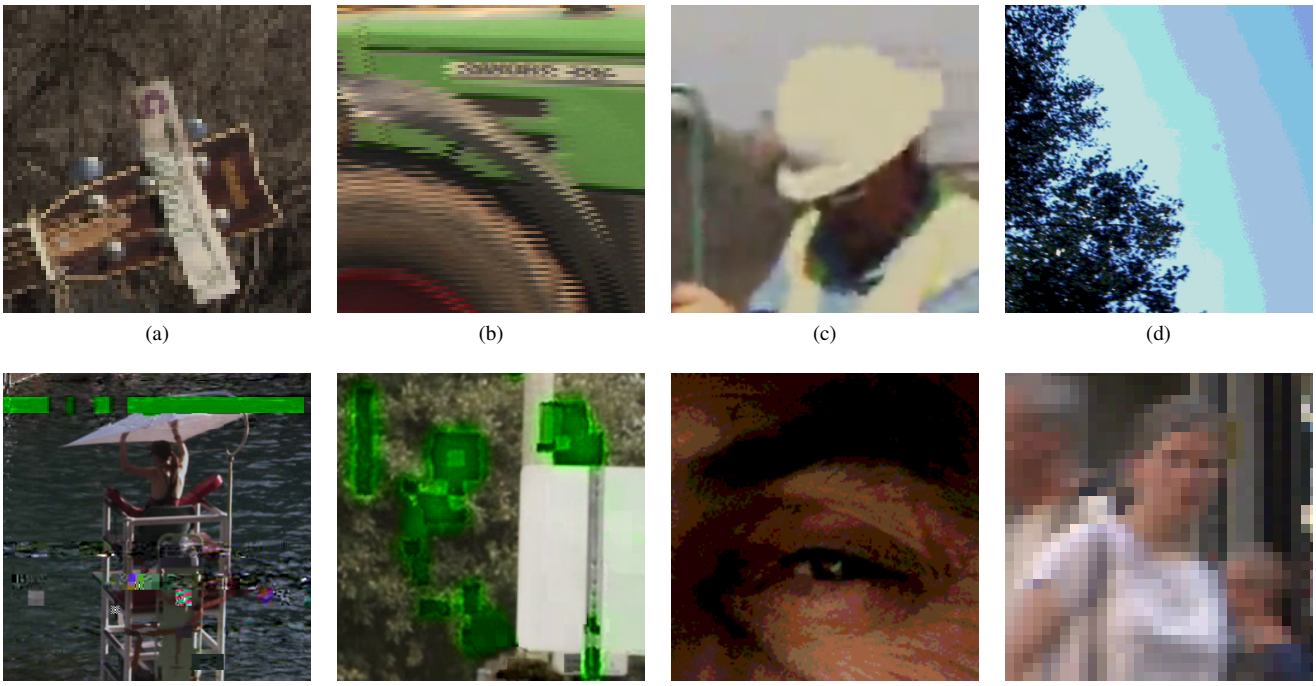
Some videos that production studios consider pristine may contain a variety of artifacts. When the resolution of a digital video is increased to fulfill a resolution requirement during post-production, upscaling artifacts will be introduced. This upscaling process adds no additional information while possibly increasing storage requirements. Unfortunately, upscaling artifacts become quite annoying at more extreme upscaling factors, where distortions such as ringing and blur become visually apparent. Previous methods that have been developed to detect upscaling artifacts include periodicity analysis [4], [5], [6], [7], [8], [9], frequency-based analysis [10], [11] [12], natural-scene statistic analysis [13], and Singular Value Decomposition (SVD) based analysis [14].

Other digital video sources may be improperly downscaled, leading to visible aliasing artifacts. Proper downscaling involves reducing the magnitude of higher frequencies prior to downsampling. Unattenuated energies from higher frequencies wrap around and distort the energies of lower frequency bands after downsampling, causing visible aliasing distortion. The visible manifestations of aliasing include "jaggies," oscillating moiré, and other content-dependent patterns, all of which can be visually annoying. Aliasing detection methods include the Signal-to-Aliasing Ratio [15], which first measures the components of image aliasing, then computes the ratio between the aliasing energy and the estimated aliasing-free energy to determine the degree of present aliasing. Coulange and Moisan [16] developed an *a-contrario* model, which uses knowledge of the original image resolution to measure suspicious co-localizations of Fourier coefficients to build up evidence of aliasing. Lastly, Eunjung *et al.* [17] developed a detection method that combines the Discrete Wavelet Transform (DWT) with the Discrete Fourier Transform (DFT) to filter a potentially aliased image, then differences the filtered result with the original image to provide a measure of aliasing.

When digital video sources are transmitted, transferred, or stored, they may be transmitted among multiple lightly compressed encodes. Unfortunately, compression artifacts can noticeably compound from multiple re-encodings. Some of this loss can be attributed to quantization in a transformed (e.g. DCT) domain. This truncation of bit depth can result in banding, producing the appearance of "false contours," or lines that appear in place of a smooth gradient. Ahn and Kim [18] devised a block-based method for detecting flat regions that appear near banding contours, by making local entropy and contrast measurements on each block. Luo *et al.* [19] explored the effect of quantization in different transform domains, and found that the ratio of densities in the distribution of non-DC components was sensitive to quantization.

Also during transmission, video encoding errors may occur before, during, and after each transmission stage. One such stage may include digital tapes, which are commonly used by studios to physically transport video content. These tapes are known to introduce corruption under certain environmental conditions. Corruptions in encoding packets are commonly called video hits, and may appear as single corrupted blocks or as groups of corrupted blocks that persist for several seconds. Methods for detecting packet corruption and loss, both with and without concealment, usually operate by detecting sharp edges near block boundaries, which are strictly defined by the codecs used in the production pipeline [20], [21]. Winter *et al.* [22] provided a video hit detection mechanism based on edge and row changes, which does not require knowledge of this structure.

Video sources may be transmitted over a channel, such as



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 1. Examples of impairments that occur in source videos ingested by the streaming video industry. (a) Aliasing/jaggies; (b) Combing; (c) Compression; (d) False contours/banding; (e) MPEG2 hits; (f) H264 hits; (g) Quantization; (h) Upscaling.

a wireless network, in which one or more whole frames may be dropped, known as dropped frames. These drops visibly manifest as unnatural staggering when motion is present [23]. Upadhyay and Singh detect dropped frames [24] by first extracting spatial entropy and content variation features followed by utilizing a Support Vector Machine (SVM) for final detection. The earlier method in [25] applies thresholds on frame differences, then detects frame drops when that threshold is exceeded.

Legacy source video content originally intended for viewing on older CRT displays was often encoded using an interlaced mode. During frame rate conversion, this mode was used to interpolate frames by copying even rows from a previous frame, and odd rows from a next frame, then combining these even/odd rows. These new interpolated frames could be added to increase the frame rate, thereby satisfying video broadcast requirements. In addition, interlacing was useful for reducing bandwidth by only transmitting even or odd rows. Unfortunately, source videos designed with interlacing modes produce visible "combing" or "zipper" artifacts on progressive displays. Methods for combing artifact detection commonly involve comparing interpolated row values with previous row values, to find evidence that a subset of previous row values were used [26], [27].

Examples of most of the different types of artifacts that we consider are shown in Fig. 1. Aliasing/jaggies can range in appearance from subtle to dramatic alteration of content, as exemplified by Fig. 1(a). Interlacing leads to "combing" artifacts, as depicted in Fig. 1(b). H264 compression, which increases blockiness and reduces details, is depicted in Fig. 1(c). We regard quantization as a separate distortion from banding, which can manifest differently, as seen by comparing Figs. 1(d) and 1(g). MPEG2 hits corruption produces small blocky artifacts, which can manifest as changes in the transform coefficient magnitudes, or in horizontal striping, as seen in Fig. 1(e). H264 hits corruption rarely leads to horizontal striping, but often causes blocky impairments, as shown in Fig. 1(f). Lastly, upscaling is an often subtle artifact, which presents as a loss of detail as in the "nearest neighbor" upscaling shown in Fig. 1(h).

Even presumably pristine video sources can be inhabited by each of these common distortions. Since no corresponding higher quality source video exists for any given impaired source video, Full-Reference (FR) detection methods cannot be used. Hence, we only pursue No-Reference (NR) methods, which measure statistics intrinsic to each video and provide detection results based on these distortion-induced statistical regularities. We first pre-process each video using a transformation motivated by perceptually relevant natural scene statistics (NSS) models, which have proven useful in developing models of image quality, and algorithms derived therefrom, such as BRISQUE [28], [29], [30], NIQE [31], FRIQUEE [32], Video BLIINDS [33], BIQI [34], IL-NIQE [35], CORNIA [36], and QAC [37]. Unlike these prior models, we do not use NSS to create features to learn on. Instead, we uniquely use an NSS model to pre-process the video, in order to perceptually sensitize the input to a convolutional neural network, to better handle NSS-destroying distortions. By leveraging these excellent natural visual priors, we are better able to efficiently isolate, model, and predict how source video artifacts perturb the local statistics in source videos.

We have devised a generalized artifact detector, called the Video Impairment Detection Mapper (VIDMAP), which can both detect and *localize* each of the aforementioned arti-

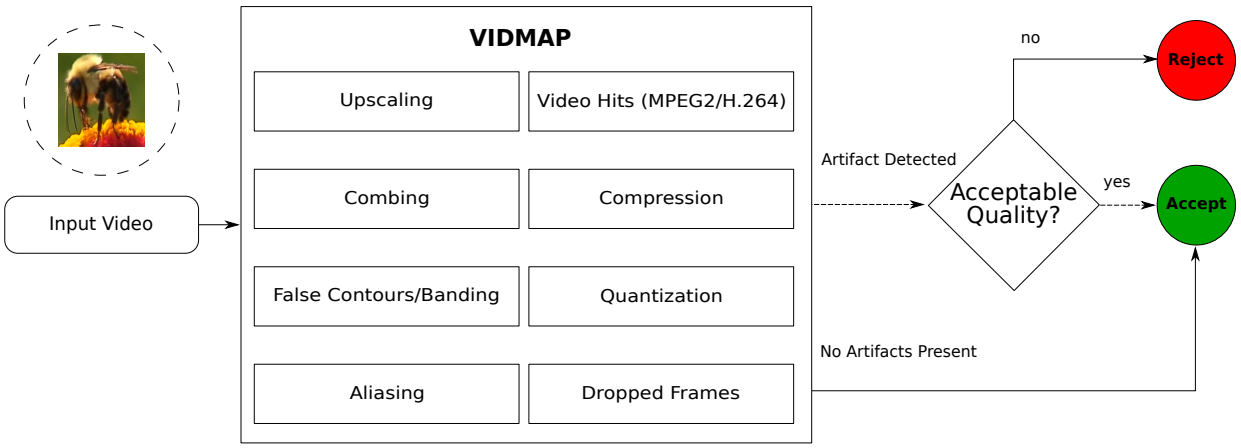



Fig. 2. VIDMAP system design. An input video is submitted to VIDMAP for artifact analysis. If an artifact is detected, the video is flagged for either manual or automatic quality assessment. Videos with an acceptably low number of artifacts can be ingested. Otherwise, the video may be rejected.

facts, without requiring a higher-quality reference video. We evaluate VIDMAP detection performance on several artifact detection tasks and compare to the performance of competing methods. We show that VIDMAP is a state-of-the-art detector of all tested artifact types, using the same simple NSS-driven network architecture across all distortions.

We make the following contributions:

- A new source artifact detection framework (Section II) called VIDMAP, that we designed to automatically analyze ingested video sources. VIDMAP uniquely produces a full-resolution detection probability map for each distortion, along with a global detection probability for each distortion. If a global detection occurs, VIDMAP assigns that video for further processing, such as manual or automatic quality assessment.
- We designed a unique NSS-based pre-processing stage that provides an intrinsically distortion-sensitive input to a shallow convolutional network (Section II-B) architecture, which is designed to locally detect and map artifacts using only frame-global distortion labels during training.
- Our CNN design includes a unique arrangement of two parallel excitatory and inhibitory (positive/negative) networks to greatly improve performance.
- We supply extensive performance comparisons (Section IV) between other leading global distortion detection methods and across varying configurations of VIDMAP.
- We provide a public release of our new model at [38], including the trained weights for each artifact type, ready for use by production studios and large-scale streaming video providers.

We believe that no high-performance, practical video source inspection system similar to VIDMAP exists, which we develop in the following sections. Section II describes the VIDMAP system, which includes the proposed pre-processing model and convolutional framework in detail. Section III describes the dataset development process for each artifact. Section IV describes detection results and visualizations for 9 artificially induced artifacts and additional results for 2 non-synthesized source video datasets. Finally, Section V presents concluding remarks.

## II. VIDMAP System

We present the VIDMAP system in Fig. 2, which provides multiple artifact detectors followed by a quality-sensitive decision module that curates input videos. An input video is provided to a bank of artifact detectors which each produce a detection result. These results are then aggregated and delivered to the curation stage. This last stage produces either a manual or automated decision regarding the final fate of each input video. When at least one artifact is detected, this stage can decide to keep the video in the collection if the quality is high or reject the video if quality is poor. If no artifacts are detected, then the video can bypass this stage, as it is assumed to be free of artifacts.

Each artifact detector component of VIDMAP makes use of a shared pre-processing stage. The output of this stage is fed to multiple identically designed convolutional networks, which use artifact-specific weights to detect each artifact. The next two subsections describe the pre-processing model and learning framework architecture in detail.

### A. Pre-Processing Model

Prior to applying VIDMAP detectors to an input video, the video is first pre-processed by center-surround, isotropic bandpass filtering, followed by a non-linear divisive normalization process [41]. We will refer to these steps collectively as Mean-Subtracted Contrast Normalization (MSCN). This transformation is used in many successful image quality assessment (IQA) models since it tends to strongly Gaussianize and decorrelate the pixels of high-quality images, while different behavior is observed on distorted image pixels [41], [30], [31].

The MSCN pre-processing stage reflects both a well-established NSS model [41], as well as simple center-surround retinal processing [28]. The BRISQUE IQA model [30] deploys parametric fits of empirical probability distributions of MSCN coefficients as the basis for extracting quality-aware picture features. However, regularities in the statistics of the sigma field $\sigma(\mathbf{x})$ have also been shown to possess significant, and complementary picture quality prediction power, e.g., as used in the FRIQUEE [32] and NIQE [31] image quality models. We have found that using both the sigma field and

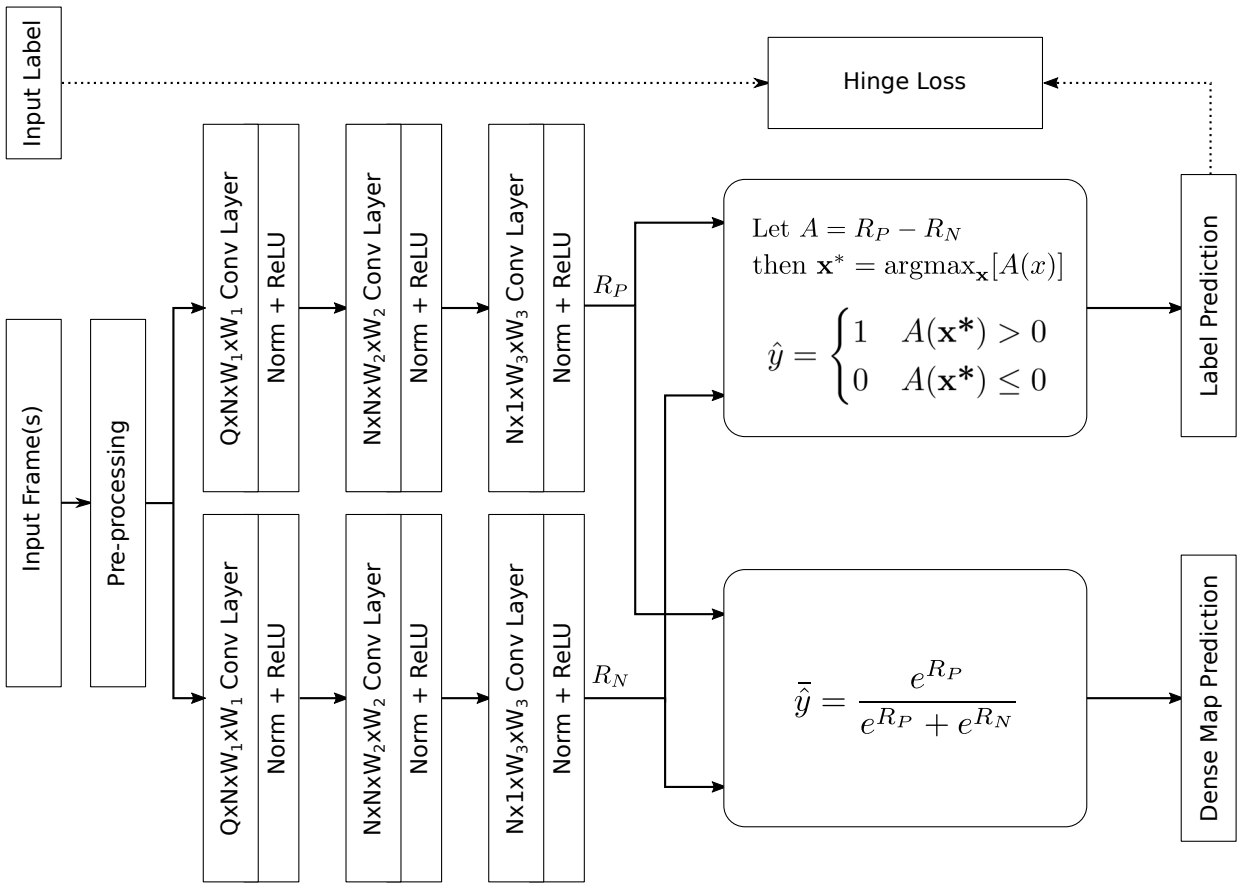



Fig. 3. VIDMAP convolutional network architecture. The pre-processing layer computes and outputs only $\sigma$ and MSCN coefficient maps from input frames or frame differences. Dotted lines indicate the portion of the network that exists only for training. During training, each input frame has a single associated binary label indicating whether the input video frames are distorted or not. Note that no loss is propagated through the dense map prediction. Rectified Linear Units [39] (ReLUs) and Instance Normalization [40] are present at all but the ouput convolutional layers.

the MSCN transformed image improve the prediction power and thus the generalizability of the VIDMAP model. This pre-processing model is unlike any of the NSS-based IQA models, since, instead of using MSCN to produce features, it uses MSCN to sensitize the system to perceptual distortions. Without this pre-processing step, the designed detectors perform significantly worse, and fail to detect visually subtle distortions, such as compression and upscaling artifacts.

### *B. Convolutional Detection Map Network*

A visual summary of the VIDMAP artifact detection network is provided in Fig. 3. The pre-processing step produces $Q = 2$ channels, the MSCN coefficients and $\sigma(\mathbf{x})$ map, from each input frame. For multi-frame input, these pre-processing channels are concatenated. For $N$ frames, the number of channels is $Q = 2N$. Alternatively, the input frames may be differenced prior to the pre-processing stage, which would provide $Q = 2(N-1)$ channels. In any case, all pre-processed input formats are reorganized into a single multichannel input before being submitted to the VIDMAP artifact detectors.

The multichannel pre-processed input is processed by three convolutional layers in two identical branches. Although identical in architecture, each branch is trained independently. Between the convolutional layers, an Instance Normalization [40] is computed to independently normalize output channels, improving training convergence. A Rectified Linear Unit (ReLU) nonlinearity is computed on the output of each of these normalizations. The size of the internal representation (i.e. the number of output channels for the first layer) is fixed at $N = 50$ for optimal classification performance. The final convolutional layers are configured to project input channels into a single response map per branch. As depicted in Fig. 3, the response map associated with the lower branch is labeled $R_N$, which serves as the inhibitory (negative) response, while the response map associated with the upper branch is labeled $R_P$, which represents the excitatory (positive) response. A final probability prediction map $\bar{\hat{y}}$ may be formed as

$$\bar{\hat{y}}(\mathbf{x}) = \frac{e^{R_P(\mathbf{x})}}{e^{R_P(\mathbf{x})} + e^{R_N(\mathbf{x})}}, \tag{1}$$

where $\mathbf{x}$ are spatial coordinates. The dotted lines in Fig. 3 indicate the portion of the network that is removed during testing.

Conceptually, a given input multichannel frame may be either non-distorted or distorted, which can be summarized by a global binary indicator label, which is only known and provided to the network during training. Some distortions affect an entire image or video frame, whereas others may only affect a small portion of a frame. No matter how a distortion might manifest, a global label indicating that at least some

subset of the image locations are distorted can be extremely useful for finding discriminating statistics between populations of distorted and non-distorted frames.

To make use of this global indicator label, we choose to backpropagate error measured from classification loss through the most discriminative point $\mathbf{x}^*$ at each iteration. By selecting this specific point, positive predictions made from frames with distortions are reinforced. Positive predictions made on frames without distortions (i.e. false positives) are minimized. The point $\mathbf{x}^*$ is found by reformulating $p(\mathbf{x})$ as

$$p(\mathbf{x}) = \frac{1}{1 + e^{-A(\mathbf{x})}}, \tag{2}$$

where $A(\mathbf{x}) = R_P(\mathbf{x}) - R_N(\mathbf{x})$ is the discrimination distance. Positive values of $A$ indicate positive detection responses, implying $p(\mathbf{x}) > 0.5$. Thus, $\mathbf{x}^*$ is determined by finding the point $\mathbf{x}$ that maximizes $A(\mathbf{x})$. This approach removes the need to know the locations of artifacts *a priori*, since the network will find them as a natural consequence of making point $\mathbf{x}^*$ more discriminative. This approach also removes the need for estimating the artifact probability at point $\mathbf{x}^*$ and differs from models that learn to compute dense image segmentation maps [42], using class labels at each coordinate of the training image.

For classification loss, we use the hinge loss [43], which maximizes the margin between two classes. By design, this allows the network to better classify data that lies closer to the decision boundary, avoiding optimizing for data that is easily classified. By contrast, cross-entropy loss, which is widely used for classification tasks, optimizes over probabilities, which is a stronger constraint. Although both loss functions provide similar state-of-the-art results, we choose hinge loss since gradients for well-classified points are zero, requiring less computation during training.

After training, artifact detection per pixel can be realized by using the same trained weights. Response maps $R_p$ and $R_N$ are fed into the softmax function in (1) to produce per pixel probability maps, as also indicated in Fig. 3. Despite the fact that propagating error through $\mathbf{x}^*$ produces excellent performance, we found that the resulting probability maps did not label some of the distorted regions. This is a phenomenon similar to that observed by Singh and Yee [44], who proposed randomly hiding the most discriminative data during training. We tried this by sampling different discriminative points, which did not produce smoother maps. Instead, we extended our approach by adding a local smoothness constraint on the output map, by using a small Gaussian kernel on $R_P$ before computing the most discriminative point $\mathbf{x}^*$. This serves two purposes: first, to allow the network to consider a neighborhood of responses while determining the most discriminative point, and second, to backpropagate error through a neighborhood of points in the map. In some cases this improves the overall detection performance of VIDMAP, but in all cases it produces more complete probability maps. Although visually helpful, we did not include this additional processing step in any analysis since the degree of smoothness depends the artifact statistics.

The trainable parameters in each VIDMAP detection network are the convolutional templates and bias vectors. The first layers in both branches contain $2N(QW_1^2 + 1)$ free parameters in total, the second layers contain $2N(NW_2^2 + 1)$ free parameters, and the last layers contain $2(NW_3^2 + 1)$ free parameters. We found that setting $N = 50$ and $W_1 = W_2 = W_3 = 11$ provided excellent generalizable performance. These settings imply that each VIDMAP detector contains about 1 million parameters. For most distortions tested, removing the middle convolutional layer from each branch yielded comparable detection performance with drastically reduced complexity. These reduced VIDMAP detection networks yield approximately 40,000 parameters.

The complexity of this "lightweight" model is vastly lower than recent deep convolutional algorithms [45] which can have greater than 100 million parameters. Light networks are highly desirable in practical source inspection environments, where the video throughput can be quite large, and where the inspection model should be easily retrainable to adapt to new distortions, compression formats, resolutions, and video sources. The efficiency of our network is greatly enhanced by the perceptual pre-processing that feeds the network, which provides a localized contrast normalization. While a much deeper network might learn to replicate or resemble this "perceptual process," this would require additional computational expense.

To train this network, we used Adam [46] with a learning rate of $10^{-4}$ and batch size of 10. All weights were initialized using a zero-mean normal distribution with scale $10^{-4}$. Each training batch was balanced, always maintaining a total of 5 positive and 5 negative samples per batch, which resulted in faster training convergence. To augment training and reduce overfitting, we randomly flipped training samples horizontally or vertically. We applied these same settings across all distortions. Training convergence was usually achieved in under 10 epochs.

## III. Dataset Preparation

We created a separate dataset for each artifact type: aliasing, combing/interlacing, compression, dropped frames, false contours, hits (H264), hits (MPEG2), quantization, and upscaling. The artifacts were generated artificially using a pristine set of videos derived from the Netflix collection. We collected a total of 1150 480p scenes and a total of 431 1080p scenes, clipped from a total of 536 different pristine contents. We identified scene boundaries using [47], which compares luminance distributions between frames. When synthesizing artifacts, we sought to maintain similar appearances as observed in discovered distorted source videos. Artifacts were introduced onto each video, and 256x256 patches extracted from random spatial locations. For each extracted patch, co-located neighboring patches in the next and previous frames were also extracted, to capture artifact behavior over multiple frames. We also required that each patch that contained an artifact had at least a minimum variance, to ensure that enough evidence existed in the patch for a detection to occur. Training and testing sets were created by dividing the input video

contents in half prior to patch extraction, to minimize any content overlap.

Videos with aliasing were created by simply downscaling frames without anti-alias filtering. On each patch, the downscaling range was chosen in the range [2.0, 4.0]. To focus on aliasing that results in visible jaggedness, we compared anti-aliased and non-anti-aliased patches. If contrast energy increased in the non-anti-aliased case, we measured contour length in the contrast difference image, which corresponds to the jaggy lines that result from aliasing. We produced a total of 60,894 samples in this dataset.

Interlaced video was produced by considering sequences of 3 frames. For example, a pristine video contains no artifacts within the 3 frames, but an interlaced video recreates the center frame by interleaving rows from the adjacent frames. For each video content, we extracted a maximum of 10 example 3x256x256 patches on the pristine original and a maximum of 10 additional patches from the interlaced copy. We collected a total of 61,653 samples in this way.

The compression dataset was created by considering the H264 encoder, which at a minimum, performs a transform-domain quantization and a deblocking filter. We randomly selected Constant Rate Factors (CRF) in the range of 24 to 37, and we randomly selected from the commonly used encoding profiles "baseline," "main," and "high" for each sample. Any compressed video was considered to be a positive sample, and any video part of the pristine sources was considered to a be a negative sample. A total of 63,012 samples were generated in this way.

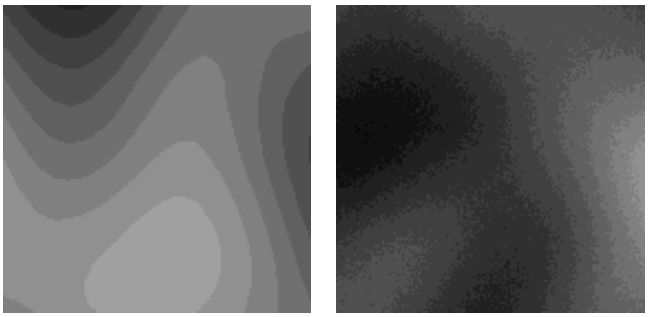
The dataset for videos with dropped frames was created by considering sequences of 4 frames, based on the design of previous algorithms that compute frame-differences before and after each potential drop. The number of frames dropped in a positive sequence were $N \in \{3, 6, 9\}$. To ensure that the drop would be visible (i.e. enough motion exists between frames), we discarded positive samples having small temporal activity TI [23]. A total of 63,030 samples were generated in this way.

Quantized video was produced by first selecting a $q \in \{8, 16, 32\}$, then for a given patch $P$, applying

$$Q = q \left\lfloor \frac{P}{q} \right\rfloor.$$

to yield the quantized patch $Q$. A total of 31,281 samples were produced in this manner.

We synthesized false contours by quantizing smooth gradients. Uniform random noise was smoothed using a Gaussian filter to produce a rich diversity of gradients. We then quantized these gradients by factors $q \in \{8, 16, 32\}$. An example of the contours produced is depicted in Fig. 4(a). After observing how film grain noise can affect the smoothness of these contours in video data, we simulated film-grain noise by adding a small amount of random Gaussian noise to our gradient prior to quantization. Examples of the contours produced on noisy gradients are provided in Fig. 4(b). The negative samples in this contour dataset were supplemented with pristine video data. The final dataset contained 730,600 100x100 samples.



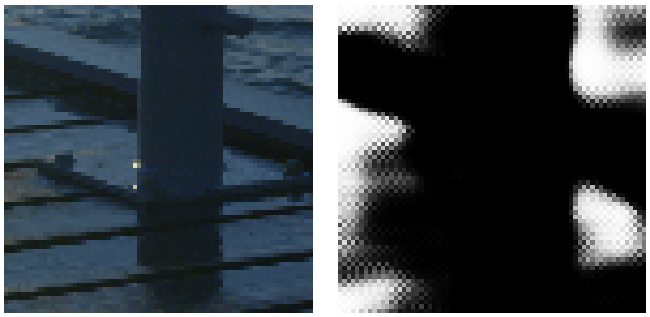
(a) (b)

Fig. 4. Examples of generated false contours. (a) False contours without noise; (b) False contours with noise.

Two video hits datasets were created, based on corrupting H264 or MPEG2 bitstreams. When corrupting the bitstreams, we used FFmpeg's 'bsf' noise flag, which allows setting the corruption ratio, defined as the ratio of correct bits to distorted bits. The lower this ratio, the more corruptions that appear. We set the ratio to a reasonable level to ensure that both large scale and small-scale artifacts would appear in the corrupted videos. To guarantee that an extracted patch contained a video hit, we applied a small threshold to compare the absolute differences between corrupted patches and their corresponding pristine patches. We set the threshold to ensure that the video hits were just noticeable when the video was played. We also avoided using error concealment during decoding of the corrupted videos. A total of 31,510 H264 and 30,043 MPEG2 hit samples were generated.

Upscaled video was produced by using one of "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos-4 Upscaling," or "Nearest Neighbor Upscaling." These commonly applied interpolation techniques vary in the number of samples required from the source video frame to estimate the destination pixel values. For example, nearest neighbor requires a single sample while Lanczos-4 uses a 8x8 sampling neighborhood. We mixed two philosophies of upscaling. First, we spatially downscaled video using Lanczos-4 rescaling, then upscaled them back to the original native frame size using one of the four interpolation methods. Second, we produced upscaled samples by upscaling video and selecting patches directly. We kept positive samples balanced with respect to these two philosophies. Pristine sequences were clipped directly from the pristine sources, and we generated additional samples by downsampling the pristine sources by a random amount, to counteract the detection of any downsampling artifacts present within the positive set of samples. The upscaling and downscaling factors were randomly selected from the range [1.25, 6.0]. We collected a total of 129,428 samples.

Finally, we gathered two datasets of non-synthesized video artifacts, which we collected by manually inspecting a large video corpus. The first dataset contains 135 video scenes that exemplify "jaggies." The second dataset contains 548 video scenes that contain interlacing artifacts. All scenes are clipped from larger 480p videos. During training and validation, we balanced the number of distorted video samples with distortion-free video samples. We divided both combing



(a) (b)

Fig. 5. Aliasing impairment map. (a) Video frame with aliasing distortion; (b) VIDMAP visualization of (a).

(a) (b)

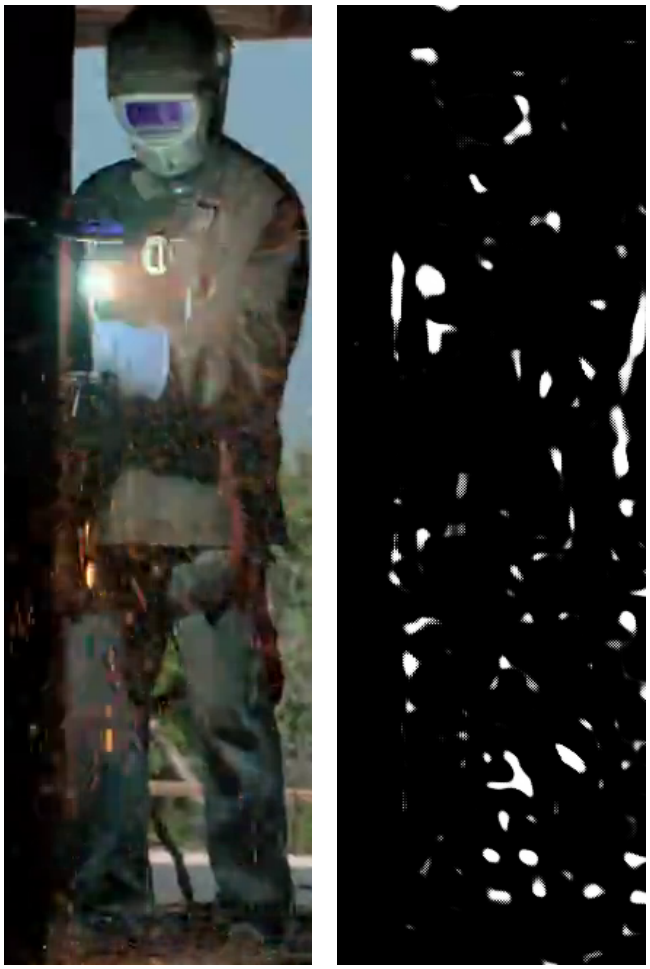
Fig. 6. Combing impairment map. (a) Video frame with combing distortion; (b) VIDMAP visualization of (a).

and jaggies datasets into training and validation subsets, taking care not to overlap content.

## IV. Artifact Detection Results

We evaluated the performance of VIDMAP against state-of-the-art methods on the aforementioned datasets. Our evaluation included measuring the errors between predictions and ground truth binary labels, hence we assessed the binary classification to VIDMAP in terms of F1 score, the harmonic mean between precision and recall, and Matthew's correlation coefficient (MCC) [49], which is a balanced measure related to the chi-square statistic. The F1 score is bounded between 0 and 1, where 1 indicates perfect precision and recall and 0 indicates the worst. A MCC of 1 indicates perfect agreement, 0 indicates no correlation, and -1 indicates perfect disagreement. Table I lists the performance results, where VIDMAP refers to VIDMAP performance using only single frames, VIDMAP (2 layer) refers to VIDMAP with only two convolution layers per branch, and VIDMAP-D refers to VIDMAP performance using frame differences.

On aliasing artifacts, VIDMAP delivered superior detection performance across all configurations. The competing method, the Signal-to-Aliasing ratio measure, performs multiple steps that are not clearly defined within the reference paper. For these steps, we tuned parameters using our aliasing dataset. Although these detection results are excellent, we discovered that that our synthesized aliasing dataset does not fully prepare VIDMAP for detecting the "jaggies" observed in real-world collections. For real "jaggies," we retrained VIDMAP on the non-synthesized data. To increase the training dataset size, we retrained VIDMAP on over 2000 video patches extracted from the non-synthesized video scenes. We evaluated retrained VIDMAP on the 480p video sequences by averaging detections across each input sequence. Videos were classified as containing aliasing when the number of detections exceeded a threshold determined using the validation set. Table II summarizes the non-synthesized aliasing video detection results. VIDMAP with 2 layers yielded superior performance than either the full VIDMAP or VIDMAP-D configurations, which likely suffer from overfitting on the training dataset.



(a) (b)

Fig. 7. Compression impairment map. (a) Compressed frame; (b) VIDMAP visualization of (a).

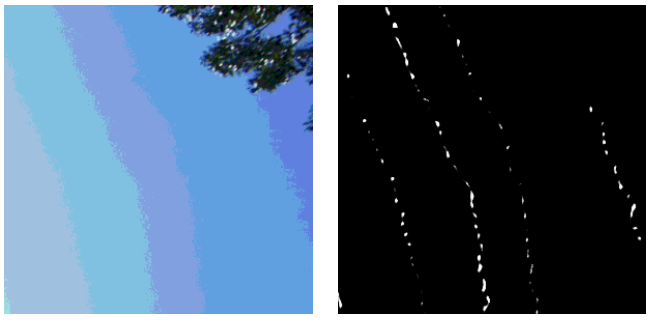
For combing artifacts, VIDMAP, VIDMAP (2 layer), and VIDMAP-D produced top results. FFmpeg's idet detector was a close second in detection performance. BRISQUE also was a very good detector of combing, despite it not being designed for the artifact. To analyze real-world performance, we ran VIDMAP on the non-synthesized combing video sequences. As in aliasing case, per-frame predictions are averaged, then an entire video segment is classified using a threshold learned using the validation set. Table III lists performance corresponding to this non-synthesized dataset. On this dataset, VIDMAP and VIDMAP (2 layer) outperform the other methods.

TABLE I
DETECTION RESULTS ON VALIDATION SETS. TOP PERFORMERS IN BOLDFACE.

| Distortion Category | Method | F1 |
|---|---|---|
| Aliasing | VIDMAP | **0.9892** |
| | VIDMAP (2 layer) | **0.9866** |
| | VIDMAP-D | 0.9760 |
| | BRISQUE [30] | 0.9615 |
| | Signal-to-Aliasing Ratio [15] | 0.6859 |
| Combing | VIDMAP | 0.9785 |
| | VIDMAP (2 layer) | 0.9723 |
| | VIDMAP-D | **0.9993** |
| | BRISQUE [30] | 0.9599 |
| | FFmpeg [26] | 0.9645 |
| | Baylon [27] | 0.9288 |
| Compression | VIDMAP | **0.9941** |
| | VIDMAP (2 layer) | 0.9869 |
| | VIDMAP-D | 0.9811 |
| | BRISQUE [30] | 0.9765 |
| | Luo *et al.* [19] | 0.8422 |
| Dropped Frames | VIDMAP-D | 0.9452 |
| | VIDMAP-D (2 layer) | **0.9550** |
| | BRISQUE [30] | 0.9091 |
| | Upadhyay and Singh [24] | **0.9532** |
| | Wolf [25] | 0.6827 |
| False Contours | VIDMAP | **0.9998** |
| | VIDMAP (2 layer) | **0.9996** |
| | BRISQUE [30] | 0.9276 |
| | Luo *et al.* [19] | 0.9533 |
| | Ahn and Kim [18] | 0.8080 |
| Hits (H264) | VIDMAP | **0.9486** |
| | VIDMAP (2 layer) | 0.9394 |
| | VIDMAP-D | **0.9503** |
| | BRISQUE [30] | 0.8273 |
| | AIDB [21] | 0.7342 |
| | Glavota *et al.* [48] | 0.8794 |
| | Winter *et al.* [22] | 0.5521 |
| Hits (MPEG2) | VIDMAP | **0.9289** |
| | VIDMAP (2 layer) | 0.9145 |
| | VIDMAP-D | 0.9106 |
| | BRISQUE [30] | 0.6342 |
| | AIDB [21] | 0.6413 |
| | Glavota *et al.* [48] | 0.8024 |
| | Winter *et al.* [22] | 0.5159 |
| Quantization | VIDMAP | **0.9930** |
| | VIDMAP (2 layer) | **0.9958** |
| | VIDMAP-D | 0.9831 |
| | BRISQUE [30] | 0.9548 |
| | Luo *et al.* [19] | 0.9871 |
| Upscaling | VIDMAP | **0.9956** |
| | VIDMAP (2 layer) | **0.9925** |
| | VIDMAP-D | 0.9848 |
| | Goodall [13] | 0.9900 |
| | BRISQUE [30] | 0.9814 |
| | Feng et al. [11] | 0.9166 |
| | Vázquez-Padín *et al.* [14] | 0.9788 |

(a) Frame 1 (b) Frame 2 (c) Frame 3 (d) Frame 4 (e) Impairment map

Fig. 8. Dropped frame impairment map. The drop of 9 frames occurred between frames 2 and 3.



(a) (b)

Fig. 9. False contour impairment map. (a) Video frame with false contour distortion; (b) VIDMAP visualization of (a).

Compression artifacts were detected well by all configurations of VIDMAP. BRISQUE, which is known to correlate well with perceived compression, performed almost as well. Luo *et al.*'s method, which is sensitive to quantization-based artifacts, demonstrates comparatively poor performance for compression.

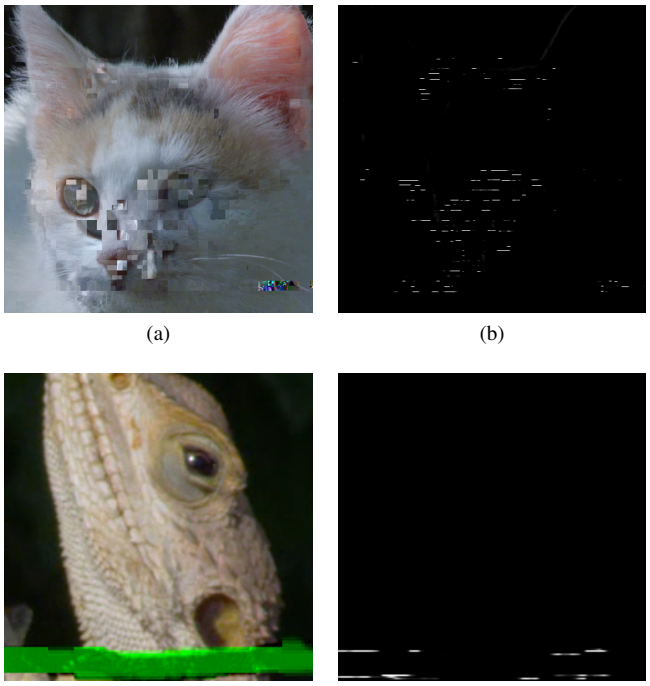
For dropped frames, we trained VIDMAP-D and 2-layer VIDMAP-D using 3 pre-processed frame-differences. The 2-layer VIDMAP-D configuration gave the best results, outperforming the full VIDMAP-D detector, which was found to be overfitting the training dataset. Upadhyay and Singh's detector gave the second best results, using a threshold value of 30

TABLE II
DETECTION RESULTS ON NON-SYNTHESIZED VIDEOS EXHIBITING ALIASING/JAGGIES ARTIFACTS.

| Method | F1 Score | MCC |
|---|---|---|
| VIDMAP | 0.8679 | 0.7992 |
| VIDMAP (2 layer) | **0.9020** | **0.8519** |
| VIDMAP-D | 0.6197 | 0.3872 |
| BRISQUE [30] | 0.8197 | 0.6804 |
| Signal-to-Aliasing Ratio [15] | 0.5376 | 0.2269 |

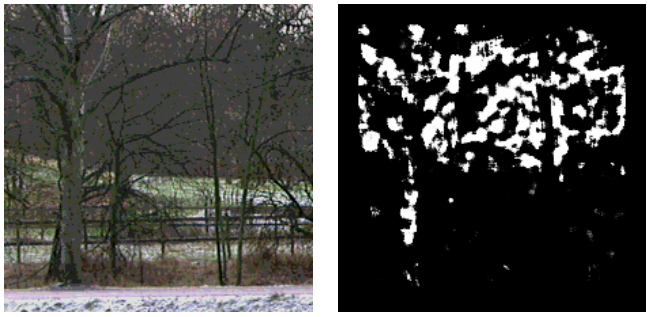
TABLE III
DETECTION RESULTS ON NON-SYNTHESIZED VIDEOS EXHIBITING COMBING ARTIFACTS.

| Method | F1 Score | MCC |
|---|---|---|
| VIDMAP | **0.9565** | **0.9094** |
| VIDMAP (2 layer) | 0.9477 | 0.8836 |
| VIDMAP-D | 0.8893 | 0.7834 |
| BRISQUE [30] | 0.9065 | 0.8141 |
| FFmpeg [26] | 0.9154 | 0.8316 |
| Baylon [27] | 0.8535 | 0.7122 |



(a) (b) (c) (d)

Fig. 10. Video Hits Impairment Maps. (a) Video frame with H264 video hits; (b) VIDMAP visualization of (a); (c) Video frame with MPEG2 video hits; (d) VIDMAP visualization of (c).

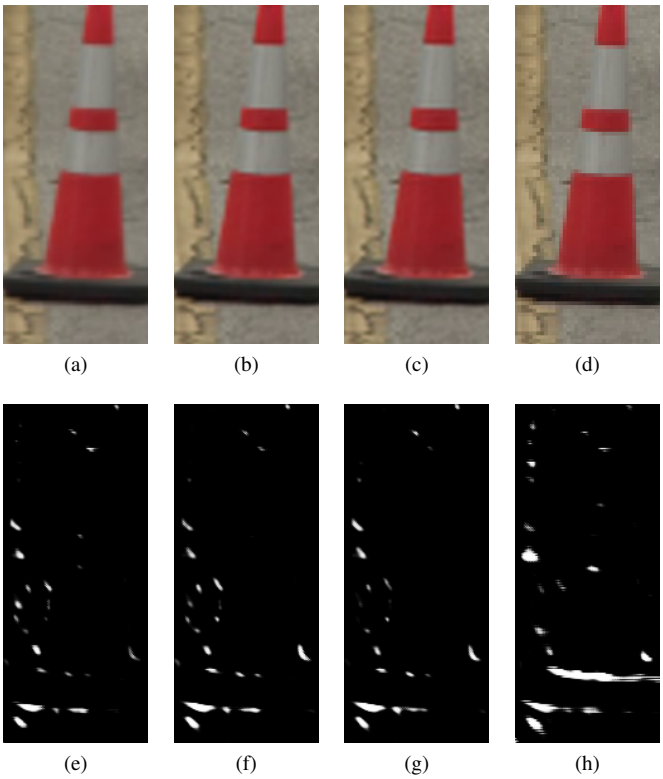


(a) (b)

Fig. 11. Quantization impairment map. (a) Quantized frame; (b) VIDMAP visualization of (a).

in their algorithm in the frame-difference binarization step. The default parameters in the Wolf's model yielded inadequate performance on the Netflix dataset. Tuning these parameters did not improve results significantly. Surprisingly, BRISQUE features extracted on the 3 frame-differences were able to provide good performance.

On the detection of false contours, we observed that VIDMAP and VIDMAP (2 layer) again outperformed other methods. The 2-layer version of VIDMAP yielded performance that was indistinguishable from VIDMAP. Luo *et al.*'s method detected nearly all of the false contours in the dataset containing quantized gradients without noise, but was less able to capture contours that appeared when quantizing noisy gradients. BRISQUE performed next best. We did not notice much difference in Ahn and Kim's method when applied to noisy vs. non-noisy gradients, since this method measures contrast and entropy at the block scale, and is unaffected by



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 12. Upscaling impairment maps. (a) Bilinear upscaled; (b) Bicubic upscaled frame; (c) Lanczos upscaled frame; (d) Neighbor upscaled frame; (e) VIDMAP visualization of (a); (f) VIDMAP visualization of (b); (g) VIDMAP visualization of (c); (h) VIDMAP visualization of (d).

differences in boundary appearance. We configured this last method with 16x16 blocks, a contrast threshold of 14.5, and entropy threshold of 3.0, and a flat region area threshold of 12.5.

For both H264 and MPEG2 video hits, we found that VIDMAP produced the best detection results compared to other methods. VIDMAP (2 layer) yielded lower performance, likely due to the variability in artifact presentation. Glavota *et al.*'s features, which measured statistics related to structured block sizes of 8x8, 16x16, and 32x32 pixels, performed quite well when paired with an SVR for prediction. There is a gap in performance for BRISQUE between detection of H264 versus MPEG2 artifacts, which is likely due to how the artifacts dataset was constructed. H264 artifacts were more numerous and more uniformly distributed across each frame, whereas the MPEG2 artifacts were fewer and much more isolated.

For quantization artifacts, a trivial detector could be devised to exploit periodic gaps in the simple image histogram. However, such an approach could not account for the local visibility or masking of quantization artifacts, nor is it interesting, since quantization can occur in a transform domain as in compression. Despite this conceptual simplicity, we found that both VIDMAP and VIDMAP (2 layer) yielded best performance. We compared performance against Luo *et al.*'s approach which was designed specifically for generalized quantization detection, finding that it performed almost as well.

For the upscaling detection problem, VIDMAP achieved

top performance, followed by VIDMAP (2 layer). The recent principal components based method [13] yielded second best performance. BRISQUE features-based prediction performed surprisingly well on detecting upscaling, since it is such a subtle artifact. The method from Feng *et al.*, which measures 2D frequency magnitude shapes, performs worst at detecting upscaling. By contrast, the spectral energy method from Vázquez-Padín *et al.* performed almost as well as BRISQUE. The difference in performance between the spectral energy and magnitude-shape methods is likely due to measurement rather than methodology, since the characterization of the frequency falloff which they both measure is highly indicative of upscaling.

Example visualizations of the probability maps predicted by VIDMAP for each artifact type are provided in the figures. In each example, the black regions depict a probability of 0, grey regions depict a probability of 0.5, and white regions depict a probability of 1. The aliased regions in Fig. 5 are detected with high certainty along edges. Figure 6 shows detection of the combing artifact, where the map appears to capture all visible portions of the artifact. Figure 7 depicts detection of H264 compression artifacts. VIDMAP does not seem to measure edge strength, but rather characteristic smoothness in low contrast regions. Figure 8 shows the computed spatial detection map for the case where 9 frames were dropped in between the remaining frames 2 and 3. Highlighted regions in the impairment map indicate motion discontinuities. Figure 9 depicts the detection of false contours on a frame with film grain noise that was quantized. The contour lines were largely captured. Figure 10 demonstrates predicted corruptions on exemplar H264 and MPEG2 streams. Notice that nearly all of the visible artifact edges are highlighted. As shown in Fig. 11, the background behind the trees is highly quantized, but the foreground toward the lower half of the image is less quantized because of the increased contrast. Figure 12 depicts the results of several upscaling interpolation methods and corresponding artifact maps computed on a video of a traffic cone. These visualizations add confidence in the reliability of VIDMAP as a general artifact detector.

To understand the tradeoff between detection accuracy and the number of convolution filters $N$, we retrained VIDMAP detectors for $N \in \{1, 5, 10, 25, 50\}$. Figure 13 depicts the results on the testset for fully trained VIDMAP detectors as a function of $N$. We found that the performance of VIDMAP plateaued after $N = 50$, with near optimal performance at $N = 25$. Similarly, Fig. 14, which traces the performance of VIDMAP-D across artifact types, shows that the performance also plateaued when the number of filters exceeded 10 except on MPEG2 video hits. This data also implies that a ranking exists among video artifacts based on the required representational capacity of VIDMAP. Intuitively, combing is a much simpler artifact to model than either of the H264 or MPEG2 video hits.

VIDMAP can be trained on small video frame sizes and then applied for detection on larger frame sizes. To explore how small the patch sizes can be, we tested 32x32, 64x64, 128x128, 196x196, and 256x256 patch sizes by randomly cropping the 256x256 patches in the synthesized datasets.

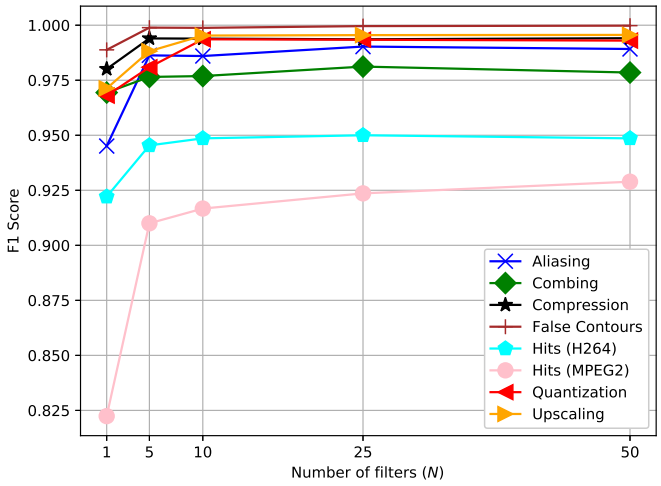



Fig. 13. VIDMAP performance as the number of filters in each layer, $N$, is changed.

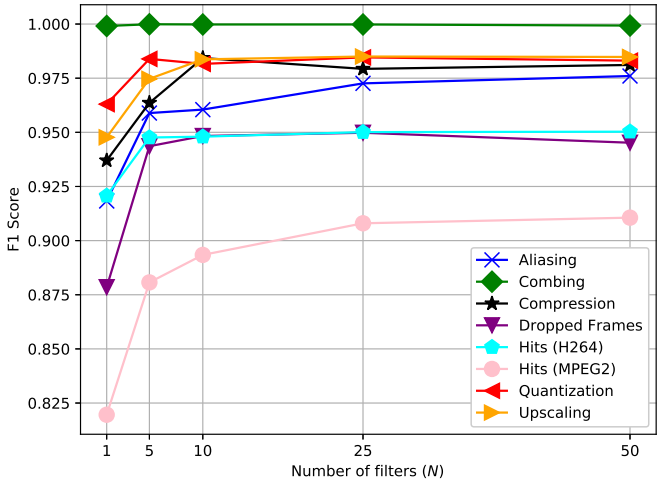



Fig. 14. VIDMAP-D performance as the number of filters in each layer, $N$, is changed.

Figure 15 depicts the accuracy on the test set of 256x256 patches as a function of this training patch size. The first size, 32x32, yields incredibly poor performance across all artifact categories. This is likely due to the 2x2 resolution of the $A(x)$ difference map, which forces VIDMAP to choose poorly discriminating $\mathbf{x}^*$ points. In general, VIDMAP finds good discriminating points for patches of size 128x128 and above. For both video hits artifact categories, the distortion is localized to some portion of the synthesized patches, which may be missed by the random cropping during training. If an artifact is missed, this contributes to noise in the training label, causing a steady increase in performance as a function of patch size for H264 and MPEG2.

## V. Conclusion/Future Work

We proposed a new video source inspection concept called VIDMAP, which is able to effectively learn how to detect and

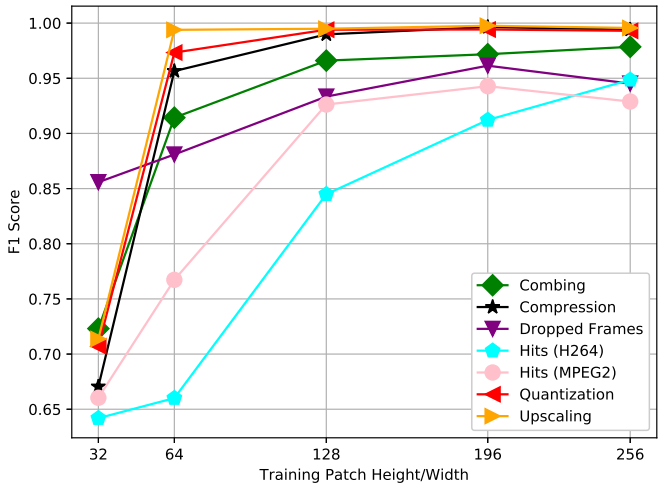



Fig. 15. VIDMAP performance on same test set for different training patch sizes.

spatially localize and map multiple types of video artifacts, using a simple NSS-driven CNN model that does not require *a priori* models of the statistics or structures of the artifacts. We showed that VIDMAP achieves state-of-the-art detection performance over all categories tested. It is a practical tool that can assist video professionals by enabling them to visualize distortion types, locations, and severities. We envision that this model will be useful as a tool for conducting source inspection of large streaming video collections.

## ACKNOWLEDGEMENTS

We thank Netflix for providing their support, access to real world video data, and research guidance. We also thank NVIDIA for providing a Tesla K40 GPGPU and multiple Titan X GPGPUs, which we used to accelerate the convolution operations.

## REFERENCES

[1] N. Sherman, "Netflix tunes into subscriber surge," http://www.bbc.com/news/business-42779953, accessed Feb 2018.

[2] A. Ha, "Hulu reached more than 17M subscribers and $1B in ad revenue last year," https://techcrunch.com/2018/01/09/hulu-17m-subscribers/, accessed Feb 2018.

[3] R. Kyncl, "From the Brandcast stage: New star-studded shows for audiences around the globe," https://youtube.googleblog.com/2017/05/from-brandcast-stage-new-star-studded.html, accessed Feb 2018.

[4] A. C. Gallagher, "Detection of linear and cubic interpolation in jpeg compressed images," *Canadian Conf. Computer Robot Vision*, pp. 65–72, 2005.

[5] S. Prasad and K. Ramakrishnan, "On resampling detection and its application to detect image tampering," *IEEE Int'l Conf Multimedia Expo*, pp. 1325–1328, 2006.

[6] S.-J. Ryu and H.-K. Lee, "Estimation of linear transformation by analyzing the periodicity of interpolation," *Pattern Recog. Lett.*, vol. 36, pp. 89–99, 2014.

[7] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, 2005.

[8] D. Vázquez-Padín and F. Pérez-González, "Prefilter design for forensic resampling estimation," *IEEE Int'l Wkshp Info Forensics Sec.*, pp. 1–6, 2011.

[9] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," *ACM Wkshp. Multimedia Sec.*, pp. 11–20, 2008.

[10] I. Katsavounidis, A. Aaron, and D. Ronca, "Native resolution detection of video sequences," *Soc. Motion Picture Television Engrs.*, 2015.

[11] X. Feng, I. J. Cox, and G. Doerr, "Normalized energy density-based forensic detection of resampled images," *IEEE Trans Multimedia*, vol. 14, no. 3, pp. 536–545, 2012.

[12] S. Pfennig and M. Kirchner, "Spectral methods to determine the exact scaling factor of resampled digital images," *Int'l Symp. Comm. Control Signal Process.*, pp. 1–6, 2012.

[13] T. Goodall, I. Katsavounidis, Z. Li, A. Aaron, and A. C. Bovik, "Blind picture upscaling ratio prediction," *IEEE Signal Process Lett*, vol. 23, no. 12, pp. 1801–1805, 2016.

[14] D. Vázquez-Padín, P. Comesaña, and F. Pérez-González, "An SVD approach to forensic image resampling detection," *EUSIPCO*, pp. 2067–2071, 2015.

[15] A. R. Reibman and S. Suthaharan, "A no-reference spatial aliasing measure for digital image resizing," in *IEEE International Conference on Image Processing*. IEEE, 2008, pp. 1184–1187.

[16] B. Coulange and L. Moisan, "An aliasing detection algorithm based on suspicious colocalizations of fourier coefficients," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 2013–2016.

[17] E. Chae, E. Lee, W. Kang, H. Cheong, and J. Paik, "Spatially adaptive antialiasing for enhancement of mobile imaging system using combined wavelet-fourier transform," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 862–868, 2013.

[18] W. Ahn and J.-S. Kim, "Flat-region detection and false contour removal in the digital tv display," in *IEEE Int'l. Conf. on Multimedia and Expo*, 2005, pp. 1338–1341.

[19] W. Luo, Y. Wang, and J. Huang, "Detection of quantization artifacts and its applications to transform encoder identification," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 810–815, 2010.

[20] N. Teslic, V. Zlokolica, V. Pekovic, T. Teckan, and M. Temerinac, "Packet-loss error detection system for DTV and set-top box functional testing," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, 2010.

[21] D. Shabtay, N. Raviv, and Y. Moshe, "Video packet loss concealment detection based on image content," in *European Signal Processing Conference*, 2008, pp. 1–5.

[22] M. Winter, P. Schallauer, A. Hofmann, and H. Fassold, "Efficient video breakup detection and verification," *Info. Extract. Media Product.*, pp. 63–68, 2010.

[23] A. A. Webster, C. T. Jones, M. H. Pinson, S. D. Voran, and S. Wolf, "Objective video quality assessment system based on human perception," in *Human Vision, Visual Processing, and Digital Display IV*, vol. 1913. International Society for Optics and Photonics, 1993, pp. 15–27.

[24] S. Upadhyay and S. K. Singh, "Learning based video authentication using statistical local information," in *International Conference on Image Information Processing (ICIIP)*, 2011, pp. 1–6.

[25] S. Wolf and M. Pinson, "A no reference (NR) and reduced reference (RR) metric for detecting dropped video frames," *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM*, 2009.

[26] "Interlace Detector (idet)," ffmpeg.org/ffmpeg-filters.html#idet, FFmpeg, accessed Mar 2017.

[27] D. M. Baylon, "On the detection of temporal field order in interlaced video data," *IEEE Int'l. Conf. Image Process.*, vol. 6, pp. VI–129, 2007.

[28] A. C. Bovik, "Automatic prediction of perceptual image and video quality," *Proc. IEEE*, vol. 101, no. 9, pp. 2008–2024, 2013.

[29] A. K. Moorthy and A. C. Bovik, "Visual quality assessment algorithms: what does the future hold?" *Multimedia Tools and Applications*, vol. 51, no. 2, pp. 675–696, 2011.

[30] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012.

[31] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a completely blind image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, 2013.

[32] D. Ghadiyaram and A. C. Bovik, "Feature maps driven no-reference image quality prediction of authentically distorted images," in *SPIE/IS&T Electronic Imaging*, 2015.

[33] M. A. Saad, A. C. Bovik, , and C. Charrier, "Blind prediction of natural video quality," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1352–1365, 2010.

[34] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices," *IEEE Signal processing letters*, vol. 17, no. 5, pp. 513–516, 2010.

[35] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2579–2591, 2015.

[36] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1098–1105.

[37] W. Xue, L. Zhang, and X. Mou, "Learning without human scores for blind image quality assessment," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 995–1002.

[38] "VIDMAP code and demonstrations," http://live.ece.utexas.edu/research/VIDMAP/VIDMAP.html.

[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[40] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[41] D. L. Ruderman, "The statistics of natural images," *Network: Comput Neural Syst*, vol. 5, no. 4, pp. 517–548, 1994.

[42] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[43] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Computation*, vol. 16, no. 5, pp. 1063–1076, 2004.

[44] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," *arXiv preprint arXiv:1704.04232*, 2017.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] K. Otsuji and Y. Tonomura, "Projection-detecting filter for video cut detection," *Multimedia Systems*, vol. 1, no. 5, pp. 205–210, 1994.

[48] I. Glavota, M. Vranješ, M. Herceg, and R. Grbić, "Pixel-based statistical analysis of packet loss artifact features," in *Zooming Innovation in Consumer Electronics International Conference (ZINC), 2016*. IEEE, 2016, pp. 16–19.

[49] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.