Artifact Detection Maps Learned using Shallow Convolutional Networks

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

***Abstract*—Automatically identifying the locations and severities of video artifacts is a difficult problem. We have developed a general method for detecting local artifacts by learning differences between distorted and pristine video frames. Our model, which we call the Video Impairment Mapper (VID-MAP), produces a full resolution map of artifact detection probabilities based on comparisons of exitatory and inhibatory convolutional responses. Validation on a large database shows that our method outperforms the previous state-of-the-art. A software release of VID-MAP that was trained to produce upscaling and combing detection probability maps is available online: http://live.ece.utexas.edu/research/quality/VIDMAP_release.zip for public use and evaluation.**

***Index Terms*—VID-MAP; Artifacts; Natural Scene Statistics; Upscaling Detection; Combing Detection; Source Inspection**

I. Introduction

Detecting the locations and severities of distortion artifacts in videos is a difficult task. Natural scene statistics models have been observed to be highly sensitive to picture impairments in general [1] [2], and have provided a logical path by which specific impairment detectors can be designed. A dense quality descriptor map would be quite valuable, not only to content providers like Netflix and YouTube to assess their own video collections, but also to Forensic investigators for finding image forgeries and to digital camera makers.

Forensic scientists are interested in finding local signs of tampering [3], where tampering might include cropping, rotation, or scaling manipulations. However, simply detecting the presence of an artifact in a video frame does not inform an investigator regarding what caused the positive detection, nor its location in the frame. Attempting to localize the detection of artifacts by processing smaller frame patches may significantly reduce detection accuracy.

A variety of useful upscaling detectors have been proposed that measure local spatial covariance [4], periodicities [5], [6], [7], [8], [9], [10], or frequency magnitude energy [11] [12]. Hybrid upscaling detectors that combine more than one of these methods have also been explored [13] [14]. Recently, a natural-scene based approach was developed that yields better detection performance, by learning sets of sparse features that are highly sensitive to upscaling artifacts [15].

Likewise, previously interlaced videos are sometimes encoded into a progressive video source, resulting in inefficient distorted video encodes. Interlacing often introduces "combing" artifacts, which manifest as annoying jagged patterns that are typically most visible along moving edges. Combing can greatly reduce video quality when played out on progressive displays.

The best existing interlacing detectors attempt to determine the relative strengths of TFF and BFF field ordering of a video. For example, the interlacing detector in FFmpeg [16] compares the field orderings over multiple frames, detecting interlacing when enough frames apparently exhibiting combing are detected. Baylon [17] introduced an interlaced frame detector that analyzes "zipper" points, which are patterns near edges that most strongly exhibit combing artifacts. Although these methods do not indicate exactly where combing is visible, they have proven to be good predictors of combing.

Generalized artifact detection is related to anomaly detection and saliency. If the probability distribution of a video signal is properly described, then anomalous patterns produce deviations from this distribution [18]. A variety of state-of-the-art picture quality prediction models [1] [2] like BRISQUE [19], NIQE [20], FRIQUEE [21], and Video BLIINDS [22] model the statistical regularities of bandpass natural images and videos, then assess distortions that disturb these regularities.

Identifying both the locations and types of artifacts in a video is difficult. Convolutional neural networks (CNNs) have proven to be capable of jointly learning object classification and localization tasks on images [23]. Until recently, these models required ground truth bounding boxes or pixel-wise segmentation masks to mark the locations of training objects [24]. However, Bazzani *et. al.* recently showed that whole-labeled images without any object location markers can be fed to a network to train it to detect and localize objects [25]. These types of techniques have not been applied to distortion artifact detection and localization in images.

Of course, for artifact detection, the problem of local assessment is simplified, since a positive detection occurs if any location in the entire image has distortion. By contrast, negative samples contain no distorted locations. We have developed a general method of detecting and mapping source video artifacts, called the Video Impairment Detection Mapper (VID-MAP), which uses globally applied distortion labels to tune the detection of local artifacts. We train the model on a CNN architecture to learn to detect and localize two of the most common video artifacts: upscaling and interlace combing. Through extensive validation experiments, we show that VID-MAP not only exceeds the performances of prior leading models, but it also yields a dense, full-resolution detection probability map.

The layout of this paper is as follows. Section II describes the proposed model in detail. Section IV presents two experiments, where IV-A describes upscaling detection results and IV-B describes combing detection results. Finally, Section V presents concluding remarks.

II. Models

A. Natural Scene Statistic Pre-Processing Model

A number of successful image quality assessment (IQA) models process images to be quality-analyzed by a local band-pass filtering operation followed by a local non-linear divisive normalization [26], known as the Mean-Subtracted Contrast Normalization (MSCN). This process tends to strongly Gaussianize and decorrelate image pixels [26], [19], [20]. The MSCN coefficients of image I are given by

$$\hat{I}(\mathbf{x}) = \frac{I(\mathbf{x}) - \mu(\mathbf{x})}{\sigma(\mathbf{x}) + C}$$

where

$$\mu(\mathbf{x}) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} I_{k,l}(\mathbf{x})$$

and

$$\sigma(\mathbf{x}) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} (I_{k,l}(\mathbf{x}) - \mu(\mathbf{x}))^2},$$

where $K = L = 3$, $\mathbf{x}$ are spatial coordinates, and $w = \{w_{k,l} | k = -K, \cdots, K, l = -L, \cdots, L\}$ is a 2D circularly-symmetric, unit volume Gaussian weighting function sampled out to 3 standard deviations. The parameter $C = 1$ avoids saturation on low-contrast regions.

III. Convolutional Detection Map Network

A visual summary the VID-MAP artifact detection network is provided in Fig. 1. Each input frame is transformed perceptually into Q channels, selected here as $\mu(\mathbf{x})$, $\sigma(\mathbf{x})$, and MSCN transforms. These channels are passed through the first layer, which includes both convolutional and bias weights. The output of this layer is then passed through an Exponential Linear Unit (ELU) [27] activation function. The layer after this applies convolution and bias weights, followed by another ELU non-linearity activation function, yielding two outputs, R_P and R_N, which are treated as excitatory (positive) and inhibatory (negative) response pairs. A final probability prediction map is formed as

$$p(\mathbf{x}) = \frac{e^{R_P(\mathbf{x})}}{e^{R_P(\mathbf{x})} + e^{R_N(\mathbf{x})}},$$

where $\mathbf{x}$ are spatial coordinates.

The ground-truth labels provided while training the network are binary. A given input image is either non-distorted or distorted, which can be summarized using a global label. Although many distortions do not affect an entire image or video frame, a global label indicating that at least some subset of the image locations are distorted can be extremely useful when finding discriminating statistics between populations of distorted and non-distorted image distributions.

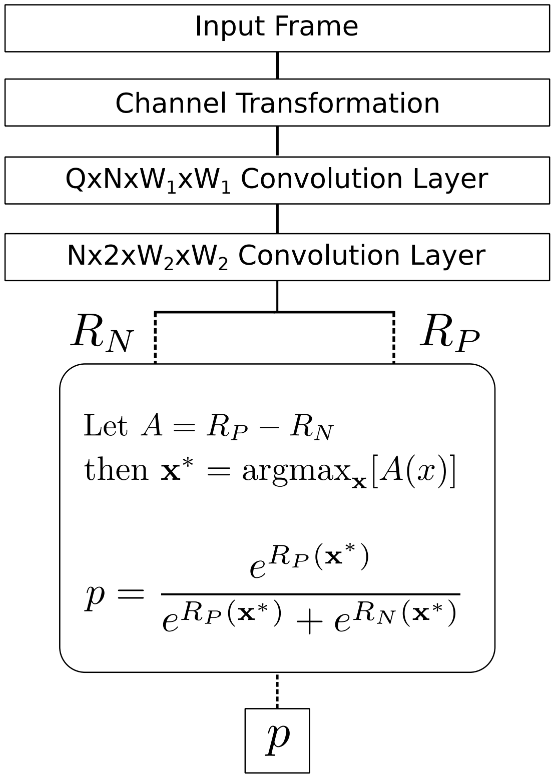
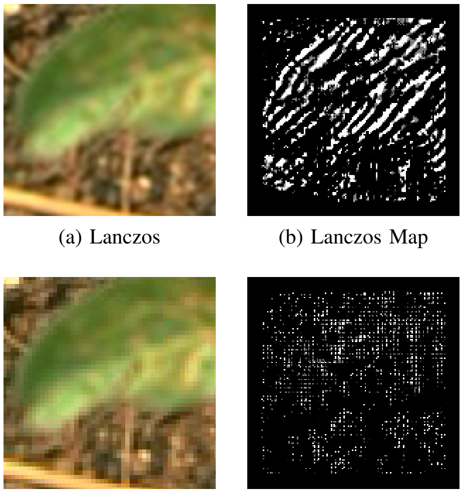

Input Frame
Channel Transformation
$\text{QxNxW}_1\text{xW}_1$ Convolution Layer
$\text{Nx2xW}_2\text{xW}_2$ Convolution Layer
R_N
R_P
Let $A = R_P - R_N$
then $\mathbf{x}^* = \text{argmax}_{\mathbf{x}}[A(x)]$
$p = \frac{e^{R_P(\mathbf{x}^*)}}{e^{R_P(\mathbf{x}^*)} + e^{R_N(\mathbf{x}^*)}}$
p

Fig. 1. VID-MAP convolutional network architecture. Dotted lines indicate the portion of the network that is removed when creating full resolution artifact detection maps. The channel transformation layer computes μ, σ, and MSCN coefficient maps. Each input frame has a single binary label indicating whether the frame is distorted or not. Exponential Linear Units [27] (not shown) are present at the convolution layer outputs.



(a) Lanczos (b) Lanczos Map

(c) Nearest (d) Nearest Map

Fig. 2. Upscaling detection maps.

Instead of backpropagating error at each response location based from each global label, we instead only backpropagate error through the most positively discriminative point $\mathbf{x}^*$. By selecting this specific point, positively labeled input images are reinforced. Negatively labeled input images help to minimize false positive responses. The point $\mathbf{x}^*$ is found by reformulating $p(\mathbf{x})$ as

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

where $A(\mathbf{x}) = R_P(\mathbf{x}) - R_N(\mathbf{x})$ is the discrimination distance.

TABLE I
UPSCALING DETECTION F1 SCORES COMPUTED ON THE TEST SET. UPSCALING TYPE INCLUDES "NOT UPSCALED," "BILINEAR UPSCALING," "BICUBIC UPSCALING," "LANCZOS UPSCALING," AND "NEAREST NEIGHBOR UPSCALING."

Algorithm	Bilinear	Bicubic	Lanczos	Nearest
VID-MAP	**0.9902**	**0.9916**	**0.9915**	0.9932
Vázquez-Padín [29]	0.9736	0.9706	0.9683	0.9929
Goodall *et al.* [15]	0.9872	0.9885	0.9941	**0.9977**
BRISQUE [19]	0.9331	0.8988	0.8847	0.8847
Feng *et al.* [12]	0.8609	0.9162	0.9577	0.9099

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 $A(\mathbf{x})$. By following this approach, the locations of artifacts in the training data are not known *a priori* or needed. This is in contrast to models that learn to compute dense image segmentation maps [28], which use class labels at each coordinate of the training image. The dotted lines in Fig. 1 indicate the portion of the network that is used during training. During testing, the R_p and R_N responses are passed through a softmax function to produce full resolution artifact detection probability maps.

The only learned parameters in this network are the convolutional and bias weights. The first layer contains $N(Q*W_1^2+1)$ free parameters, while the second layer learns $2(N*W_2^2+1)$ free parameters. Thus, the complexity of this "lightweight" model is quite low as compared with recent deep convolutional algorithms. The first layer filters learn local statistics, while the second layer learns larger scale features. The efficiency of the network is greatly enhanced by the perceptual preprocessing that computes the MSCN inputs. While a much deeper network might learn to replicate or resemble this "perceptual process," this would require additional computational expense. We used the nominal values $W_1 = 5$ and $W_2 = 11$ in all experiments.

IV. RESULTS

We evaluated two types of problems using the same network: the video upscaling (interpolation) detection and the combing (interlacing artifact) detection problems.

A. Upscaling Detection

For the upscaling problem, we selected a total of 663,462 pristine patches from the Netflix video collection. We divided these patches into non-overlapping training and testing sets of nearly equal sizes that did not share any frame content. This yielded 332,759 test patches and 330,703 training patches. Positive samples of upscaling were produced using one of two methods. In the first method, we downscaled pristine video frames using a Lanczos-4 filter before upscaling by "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," or "Nearest Neighbor Upscaling." In the second method, we center-cropped from within the video frame, then upscaled using one of the same four interpolation methods. Negative samples were generated as both original pristine patches and Lanczos-4 downscaled patches. All patches were of size 100x100. The VID-MAP network was trained using a batch size of 100, while ensuring that batches were class-balanced at each iteration.

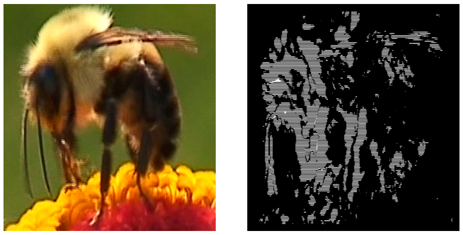

(a) Input Frame (b) Combing Map

Fig. 3. Example of a positive combing detection map.

There are two important parameters of the model: the filter sizes and the number of filters N. We chose to use 5x5 filters in the first layer to be responsive to fine details, and 11x11 filters in the next layer able to summarize these features for detection. We found that not many filters are needed to achieve high accuracy on this detection task.

For visualization, we selected an image from the Berkeley Segmentation Dataset, then upscaled it by 3x using two of the four upscaling methods. A visual comparison of the probability maps computed on the "Lanczos Upscaling" and "Nearest Neighbor Upscaling" versions of that image are provided in Fig. 2.

We numerically evaluated the performances of the models using the F1 score, which is the harmonic mean of precision and recall. Table I compares the performance of VID-MAP to several other models, using $p(\mathbf{x}^*)$ as the final predicted class label. One of the compared models is a general-purpose blind IQA algorithm (BRISQUE). We included this high-performance general model to determine whether, and to what degree, the BRISQUE features contribute to the detection task. As shown in the Table, BRISQUE did not perform nearly as well as artifact-specific detectors, while remaining competitive with Feng *et. al.* [12].

B. Combing Detection

We collected a training/validation dataset of 581 interlaced videos, each containing 3 frames, which were determined to contain visible combing artifacts. Specifically, a combed video of three frames was so labelled if at least the middle frame of the three contains visible combing when visually examined. Another set of 581 non-interlaced videos was gathered as negative samples. A negative sample was defined as one where none of the three frames was deemed to exhibit any visible combing. We also collected a separate content-distinct test dataset containing 75 interlaced three-frame sequences and 75 undistorted three-frame sequences, where videos were selected in the same manner.

Since combing artifacts generally manifest locally, we created a complementary training dataset to increase the number of samples. We manually selected regions of interest (ROIs) that visually exhibited the combing artifact from among the videos in the training set. Having selected the ROIs, we extracted 100x100 patches centered at these points. Overall, we collected 3,102 combed patches. To balance these combed

TABLE II
F1 SCORES ACHIEVED BY THE COMPARED COMBING DETECTION MODELS ON THE SET OF 150 VIDEO SEQUENCES.

Algorithm	F1
VID-MAP	**0.9868**
BRISQUE [19]	0.8718
FFmpeg	0.9167
Baylon [17]	0.8811

patches, we also extracted a total of 25 from each negative sample in the training set, yielding a total of (581 x 25) 14,525 negative patches.

For probability map visualization, we obtained a video "Bee on Flower" from the internet archive [30], which contains visible interlacing. A portion of this frame, along with the corresponding aligned portion of the detection map, is shown in Fig. 3. Detected combing artifacts are in white. Combing artifacts are clearly detected on the bee, while the background, which exhibits no visible combing artifacts, produces no false detections. The complete video and detection map can be viewed at [31].

Table II lists the obtained combing detection performance results for multiple models. Our single-frame combing detection model clearly yields stand-out, state-of-the-art combing detection performance.

V. Conclusion/Future Work

We suspect that the generality of our network model will enable us to easily configure it for the detection of other video artifacts, and indeed, we plan to apply it to an array of other important video artifact detection problems. Since a given distorted image often contains more than a single location exhibiting an artifact, we plan to find ways to train on these additional locations. For example, Singh and Yee [32] proposed randomly hiding the most discriminative points during training, to increase generality.

We also plan to introduce temporal information, e.g., by extracting temporal perceptual features relevant to temporal statistics, thereby enriching the dimensionality and diversity of the VID-MAP model.

VI. Acknowledgements

We thank Netflix for providing necessary access to data and research guidance. We also thank NVIDIA for providing a Tesla K40 GPGPU, which we used to develop the convolutional network.

References

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

[2] 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.

[3] D.-K. Hyun, S.-J. Ryu, H.-Y. Lee, and H.-K. Lee, "Detection of upscale-crop and partial manipulation in surveillance video based on sensor pattern noise," *Sensors*, vol. 13, no. 9, pp. 12 605–12 631, 2013.

[4] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Trans. Info. Forensics Sec.*, vol. 3, no. 3, pp. 529–538, 2008.

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

[6] 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.

[7] 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.

[8] 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.

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

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

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

[12] 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.

[13] N. Zhu, X. Gao, and C. Deng, "Image scaling factor estimation based on normalized energy density and learning to rank," *IEEE Int'l. Conf. Sec., Pattern Anal., and Cybern.*, pp. 197–202, 2014.

[14] 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.

[15] 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.

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

[17] 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.

[18] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 17–31, 2007.

[19] 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.

[20] 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.

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

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

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[25] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *2016 Winter Conference on Applications of Computer Vision*. IEEE, 2016.

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

[27] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

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

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

[30] R. C. Schloss, "Bee on Flower," https://archive.org/details/BeeOnFlowerHd1080i, accessed Apr 2017.

[31] "Detected Combing Video Demonstration," http://live.ece.utexas.edu/research/demo/combing_demo.mp4.

[32] 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.