The Dissertation Committee for Todd Richard Goodall
certifies that this is the approved version of the following dissertation:

Inspection and Evaluation of Artifacts in Digital Video Sources

Committee:

Alan C. Bovik, Supervisor

Wilson S. Geisler

Haris Vikalo

Joydeep Ghosh

Anush Moorthy

Inspection and Evaluation of Artifacts in Digital Video Sources

by

Todd Richard Goodall

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2018

Dedicated to Jeana.

Acknowledgments

I must express my appreciation for the multitude of remarkable people who have helped me reach this moment. I'm not sure I would have survived the PhD program without their help.

First of all, I must thank Prof. Bovik, who has shared his patience and wisdom in an effort to usher me through the graduate program at UT Austin. He accepted my request to be challenged, which was my motivation for joining the program in the first place. He has since sent many interesting and exciting opportunities my way over the years. No matter what, he has always been there to support me and all of the LIVE members. I have greatly appreciated his mentoring style, where instead of providing solutions that can be copied, he offers interesting ideas that can be explored. He allowed me to spend much of my time working independently, examining solutions up to the point where he sends an email not to demand progress, but to ask how I'm doing. For the papers that we co-authored, he has provided swift and invaluable feedback, usually filling the spaces of my papers with blood-red ink that looks a little frightening to the uninitiated. I truly believe that I could not have asked for a better advisor.

Next, I thank the LIVE members past and present – especially Christos, Leo, Lark, Greg, Janice, Zeina, Deepti, Praful, Anish, and Anush for our lively discussions in the lab and in industry. Our outings will always be fondly remembered. Christos was an awesome roommate, is a knowledgeable colleague, and a dear friend. He has never turned down a lunch invitation. I also appreciate the time spent talking to Leo, who is someone who has inspired me to be more ambitious.

My friends and cofounders of CargoSpectre – Mikey, Shai, Jeremy, and Jason – have remained supportive of my PhD goals. Mikey in particular has been an amazing friend, allowing me to borrow his car many times, giving me personal tours of the Seattle-Redmond area, and volunteering to take on some of my startup workload when the PhD devoured too much time.

My girlfriend Jeana has put up with my many sleepless nights when I had to meet a deadline or try that new idea. She has always kept her mind on my health, reminding me to watch my nutrition and to find time to exercise. She has also humored my vague, confusing, and absurd attempts at explaining my ideas through analogy. I am grateful for her support, and for her reminders that there is more to life than research.

Our two cats, Dexter and Rita, were always supportive, and they have enough personality to be considered people. By aggressively displacing my keyboard with her furry little body, Rita has kept me feeling calm. Dexter has selflessly suggested that I take breaks throughout my day, usually when he is hungry.

Finally, I'd like to thank the researchers who have come before me.

Without their discoveries, my research would have been less interesting in the sense that it would not have been possible. A beautiful summary of that fact has been state by the late Bernard of Chartres. He said that we are like dwarves perched on the shoulders of giants, and thus we are able to see more and further than the latter. My research will be at least a small part of the mountain that the future dwarves can stand upon.

Inspection and Evaluation of Artifacts in Digital Video Sources

Publication No. ____________________

Todd Richard Goodall, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Alan C. Bovik

Streaming digital video content providers such as YouTube, Amazon, Hulu, and Netflix collaborate with production teams to obtain new and old video content. These collaborations lead to an accumulation of video sources, some of which might contain unacceptable visual artifacts. Artifacts may inadvertently enter the video master at any point in the production pipeline, due to any of a number of equipment and user failures. Unfortunately, these artifacts are difficult to detect since no pristine reference exists for comparison. As of now, few automated tools exist that can effectively capture the most common forms of these artifacts. This work studies no-reference video source inspection for generalized artifact detection and subjective quality prediction, which will ultimate inform decisions related to acquisition of new content.

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 in the statistics between distorted and pristine video frames. Our model, which we call the Video Impairment Mapper (VIDMAP), produces a full resolution map of artifact detection probabilities based on comparisons of excitatory and inhibatory convolutional responses. Validation on a large database shows that our method outperforms the previous state-of-the-art of even distortion-specific detectors.

A variety of powerful picture quality predictors are available that rely on neuro-statistical models of distortion perception. We extend these principles to video source inspection, by coupling spatial divisive normalization with a series of filterbanks tuned for artifact detection, implemented using a common convolutional framework. We developed the Video Impairment Detection by SParse Error CapTure (VIDSPECT) model, which leverages discriminative sparse dictionaries that are tuned to detect specific artifacts. VIDSPECT is simple, highly generalizable, and yields better accuracy than competing methods.

To evaluate the perceived quality of video sources containing artifacts, we built a new digital video database, called the LIVE Video Masters Database, which contains 384 videos affected by the types of artifacts encountered in otherwise pristine digital video sources. We find that VIDSPECT delivers top performance on this database for most artifacts tested, and competitive performance otherwise, using the same basic architecture in all cases.

Table of Contents

Acknowledgments **v**

Abstract **viii**

List of Tables **xiii**

List of Figures **xv**

Chapter 1. Introduction **1**

1.1 Problem 1

1.2 Common Artifacts Observed in Source Videos 2

1.3 Contributions 7

Chapter 2. Background **10**

2.1 Image and Video Quality Assessment (I/VQA) 10

2.2 Real World No-Reference I/VQA Performance 12

2.3 Specialized Source Artifact Detectors 13

2.3.1 Upscaling 13

2.3.2 Combing 15

2.3.3 Aliasing 16

2.3.4 "False Contours" and Banding 17

2.3.5 Dropped Frames 17

2.3.6 Video Hits 18

2.3.7 Incorrect Aspect Ratio 19

2.4 Sparsity and Natural Scene Statistics 19

Chapter 3. VIDMAP: Video Impairment Detection MAP 20

3.1 VIDMAP System 22

3.2 Models 22

3.2.1 Pre-Processing Model 22

3.2.2 Convolutional Detection Map Network 25

3.2.2.1 Version 1 25

3.2.2.2 Version 2 28

3.3 Dataset Preparation 30

3.3.1 Synthesized Datasets 30

3.3.2 Non-synthesized Datasets 34

3.4 Artifact Detection Results 34

3.5 Artifact Detection Maps 40

3.6 Discussion and Conclusion 41

Chapter 4. LIVE Video Masters Database 52

4.1 Video Synthesis 52

4.2 Subjective Study Design 55

4.3 Subjective Data Analysis 57

4.4 Discussion and Conclusion 60

Chapter 5. VIDSPECT: Video Impairment Detection by SParse Error CapTure 63

5.1 Upscaling-Sensitive model 64

5.1.1 Proposed Natural Scene-Based Model 64

5.1.2 General Prediction Performance 70

5.1.3 Movie and TV Show Upscaling Prediction Performance 72

5.1.4 General Classification Performance 74

5.1.5 Movie and TV Show Upscaling Classification Performance 75

5.1.6 Discussion 75

5.2 VIDSPECT System Design 76

5.2.1 Pre-Processing Model 78

5.2.2 Patch-based Sparsity Model 80

5.2.2.1 Upscaling Problem Analysis 85

5.2.2.1.1 Detection 87
5.2.2.1.2 Method Discrimination 91
5.2.2.1.3 Kernel Estimation 92
5.2.2.1.4 Native Resolution Prediction 94
5.2.2.2 Interlace Detection 95
5.2.3 Convolutional Sparsity Model 97
5.3 Distorted Video Datasets 101
5.4 Model Analysis 104
5.5 Detection Analysis 107
5.6 Video Quality Prediction Model 109
5.7 VIDSPECT System Analysis 111
5.8 Discussion and Conclusion 113

Chapter 6. Conclusion and Future Work **118**

Appendices **121**

Appendix A. Sparse Functional in ADMM Form **122**

Appendix B. Combing Detection Algorithms **126**

Appendix C. Instructions for Subjects **129**
C.1 General Instructions 129
C.2 How to Score the Videos 130
C.3 Training 131

Bibliography **132**

List of Tables

1.1 List of Artifacts found in Source Videos. 4

3.1 Upscaling detection F1 scores computed on the test set for VIDMAP version 1. Upscaling type includes “Not Upscaled,” “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling.” 34

3.2 F1 scores achieved by the compared combing detection models on the set of 150 video sequences for VIDMAP version 1. . . 35

3.3 Detection results on validation sets. Top performers in boldface. 36

3.4 Detection results on validation sets. Top performers in boldface. 44

3.5 Detection results on videos exhibiting combing artifacts. . . . 44

3.6 Detection results on videos exhibiting aliasing/jaggies artifacts. 45

5.1 Median prediction performance across upscaling methods over 1000 train/test trials on “Berkeley” dataset. 71

5.2 Median prediction performance across upscaling methods over 1000 train/test trials on “Berkeley” dataset. The presence of ’*’ indicates that all upscaling methods are present in the testing and training sets. 72

5.3 Median prediction performance across upscaling methods over 1000 train/test trials on “Movie and TV Show” image dataset. 72

5.4 Median prediction performance across upscaling methods over 1000 train/test trials on “Movie and TV Show” image dataset. The presence of ’*’ indicates that all upscaling methods are present in the testing and training sets. 73

5.5 Median classification accuracy across upscaling methods over 1000 train/test trials on “Berkeley” dataset. The presence of ’*’ indicates that all upscaling methods are present in the testing and training sets. 73

5.6 Median classification accuracy across upscaling methods over 1000 train/test trials on “Movie and TV Show” dataset. 74

5.7 Upscaling detection performance measured on the test set of 20,000 patches, where upscaling type includes “Not Upscaled,” “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling.” The reported measure is the F1 score. 88

5.8 Upscaling detection performance measured on the test set of 20,000 patches, where upscaling type includes “Not Upscaled,” “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling.” The reported measure is Matthew’s Correlation Coefficient (MCC). 89

5.9 Upscaling type discrimination performance on the test set of 20,000 video frame patches when classifying upscaling type among “Not Upscaled,” “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling.” Reported values are F1-macro scores, since the classes are well-balanced. 93

5.10 MSE between predicted kernel and true kernel for different α and λ evaluated on the test set of patches. The number of basis functions used is 100. 93

5.11 Native resolution prediction on patches that were not upscaled, and upscaled using “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling” using upscaling ratios chosen from the range 1.25x to 3x. The metric being measured is SRCC. 94

5.12 Native resolution prediction on patches that were not upscaled, and upscaled using “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling” using upscaling ratios chosen from the range 1.25x to 3x. The metric being measured in MSE. 95

5.13 Combing detection results computed on the test set of 150 video sequences. 97

5.14 Detection results evaluated on test datasets. Boldface indicates best performing method. 115

5.15 Median quality prediction results evaluated on 1000 randomized train/test splits. 116

5.16 Normalized confusion matrix computed using VIDSPECT to classify videos in the Video Masters database. Class prediction probabilities are averaged over 1000 trials. Boldface indicates highest predicted value. Aspect Ratio is abbreviated as AR. . 117

C.1 Qualitative levels of the continuous video rating scale. 130

C.2 Distortion types 130

List of Figures

1.1 Hours of original content added each year to Netflix collection [1, 11, 14]. 3

1.2 Hours of video uploaded to YouTube per minute measured each year [19, 20]. 3

1.3 Examples of impairments that occur in source videos ingested by the streaming video industry. (a) Aliasing/jaggies; (b) MPEG2 hits; (c) H.264 hits; (d) Quantization; (e) False contours/banding; (f) Combing; (g) Upscaling; (h) Compression. 6

3.1 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 is rejected. 23

3.2 VIDMAP convolutional network architecture in the first configuration. 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 [39] (not shown) are present at the convolution layer outputs. 25

3.3 Second version of the VIDMAP convolutional network architecture. Dotted lines indicate the portion of the network that exists only for training. No loss is propagated through the dense map prediction. The pre-processing layer computes σ and MSCN coefficient maps. Each input frame has a single associated binary label indicating whether the frame is distorted or not. Exponential Linear Units [39] are present at all convolution layer outputs. 42

3.4 Examples of generated false contours. (a) False contours without noise; (b) False contours with noise. 43

3.5 Video Hits Impairment Maps. (a) Video frame with H.264 video hits; (b) VIDMAP visualization of (a); (c) Video frame with MPEG2 video hits; (d) VIDMAP visualization of (c). 46

3.6 Combing impairment map. (a) Video frame with combing distortion; (b) VIDMAP visualization of (a). 47

3.7 Aliasing impairment map. (a) Video frame with aliasing distortion; (b) VIDMAP visualization of (a). 47

3.8 False contour impairment map. (a) Video frame with false contour distortion; (b) VIDMAP visualization of (a). 48

3.9 Quantization impairment map. (a) Quantized frame; (b) VIDMAP visualization of (a). 48

3.10 Compression impairment map. (a) Compressed frame; (b) VIDMAP visualization of (a). 49

3.11 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). 50

3.12 Dropped frame impairment map. The drop of 9 frames occurred between frames 2 and 3. 51

4.1 Thumbnails of free contents in Video Masters database. 53

4.2 Examples of impairments that occur in source videos ingested by the streaming video industry. (a) H.264 hits; (b) MPEG2 hits; (c) Incorrect Aspect Ratio; (d) Quantization; (e) Upscaling. 54

4.3 Distribution of mean opinion scores per distortion. 55

4.4 Distribution of overall raw scores. 59

4.5 Distribution of per-video mean opinion scores. 59

4.6 Distribution of z-scored per-video mean opinion scores. 60

4.7 Distribution of z-scored mean opinion scores per distortion type. 61

4.8 Subject to subject median agreement from 1000 random splits. 62

5.1 Exemplar pristine image selected from the Berkeley image segmentation database [79]. 65

5.2 Basis functions computed using PCA on 5x5 patches. All patches were obtained on pristine images from the Berkeley image segmentation database [79]. 66

5.3 Histograms of MSCN and vertical paired product for basis filter 6 for different degrees of upscaling. These coefficients were computed using bicubic upscaling of the image in Fig. 5.1. . . 67

5.4 Absolute value SROCC between each basis function and the upscaling ratio. Images are upscaled using one of bilinear, bicubic, or Lanczos interpolation. 69

5.5 VIDSPECT system for detecting and assessing artifact severity. 76

5.6 Processing stages of VIDSPECT used to compute an artifact prediction given any input image and trained sparse filterbank. 84

5.7 Filter developed for positive and negative evidence categories. 88

5.8 Upscaling kernel prediction for individual samples randomly selected from test set. The basis learned using $\lambda = 0.1$ and $\alpha = 1000.0$ was used. 89

5.9 Dictionaries learned for each evidence category, when assigning 10 filters to each. Filter size is held constant at 25x25. 92

5.10 Sparse filters learned for interlacing. 95

5.11 Convolutional sparsity feature extraction flowchart. 100

5.12 Basis functions in ϕ, when training on (a) Upscaling; (b) Banding; (c) Hits (H.264); (d) Hits (MPEG2); and (e) Incorrect Aspect Ratio. 105

5.13 Hallucinated distortion patterns. (a) Upscaling; (b) Banding; (c) Hits (H.264); (d) Hits (MPEG2); and (e) Incorrect Aspect Ratio. 106

5.14 Measurement of total loss of VIDSPECT during training. . . . 107

5.15 Correlations of pooled θ output to human opinion scores. Responses are sorted by correlation magnitude. 112

Chapter 1

Introduction

1.1 Problem

Over the past decade, video streaming companies such as such as Netflix, Hulu, and YouTube have been at the forefront of new content. Netflix has been increasing its production of original content, even outpacing the production of existing content giant HBO [9]. With the advent of on-demand streaming, consumers are capable of choosing from a variety of services. Consumers can subscribe to YouTube Red, an increasingly popular platform for independent content producers [21], which has an increasingly large number of channels. As of 2017, over one billion hours of YouTube video are watched worldwide each day [18]. These same consumers can also sign up for Amazon Prime Video, Netflix, Hulu, HBO Now, which all promise premium streaming video content services. Keeping pace with increased consumer demand, the aforementioned streaming companies have an ever-increasing demand for new content.

As streaming companies expand, they acquire a diverse and growing consumer base [119, 51, 63] based on their selection of content. Such content is comprised of productions both new and old. Netflix began releasing original content in 2013 [1], and released a total of 1000 hours of original content in 2017 [11], as summarized by the trend depicted in Fig. 1.1. This trend continues into 2018 with the addition of 80 original films [126]. Other streaming service companies, such as Hulu and Amazon Prime Video, have increased their video production to maintain a competitive edge [13]. Disney has plans to entering the streaming business [95] in 2019. A haven for independent content producers, YouTube reports that videos are uploaded at the rate of 400 hours per minute [19]. This upload rate has been steadily increasing since 2007, when just 6 hours of video were uploaded per minute [20]. This trend is shown in Fig. 1.2. With increased on-demand video streaming services, more content is being made to fulfill consumer demand and maintain a competitive edge.

Demand necessitates an increase in content production. With such an increase, maintaining excellent video quality becomes more laborious. Videos cannot be manually inspected with fine granularity, and even a coarse inspection becomes a major burden for any inspection team. A set of automated quality assessment tools that can ensure that source video quality standards are maintained would be an invaluable asset as content collections grow.

1.2 Common Artifacts Observed in Source Videos

Ideally, video content is sourced directly from professional production studios, who try to guarantee that videos are generally free of distortions. This sourced content takes the form of both new and old productions, for

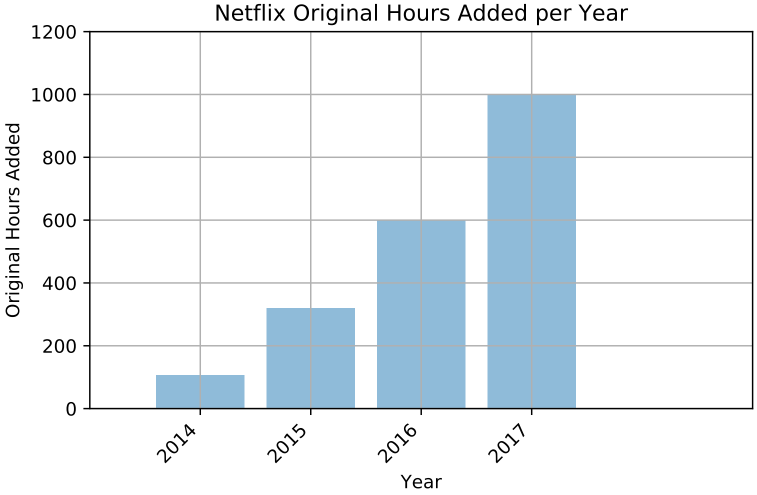


Figure 1.1: Hours of original content added each year to Netflix collection [1, 11, 14].

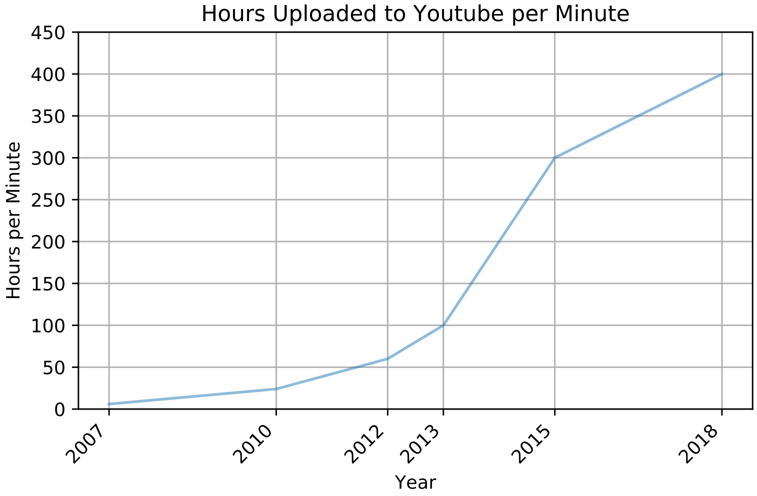


Figure 1.2: Hours of video uploaded to YouTube per minute measured each year [19, 20].

Table 1.1: List of Artifacts found in Source Videos.

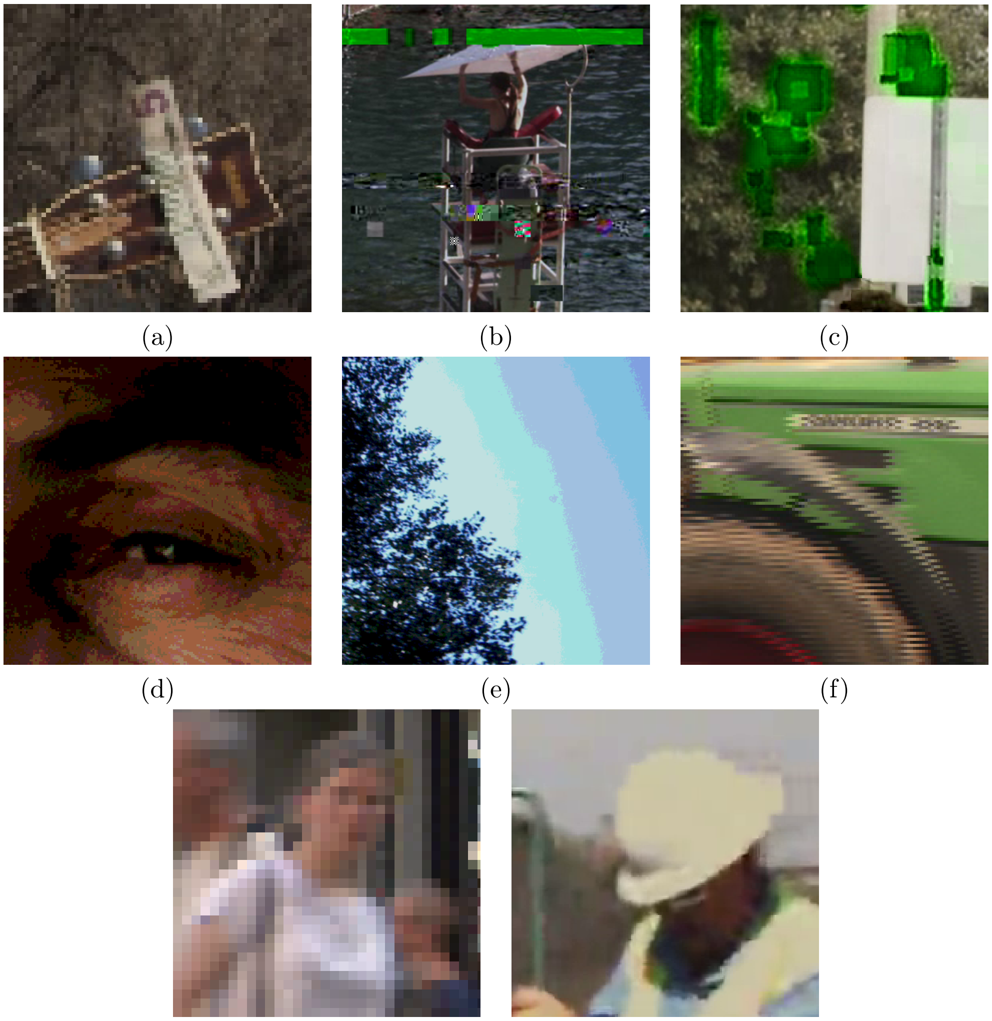
Combing	Upscaling	Video hits
Aliasing/"Jaggies"	Dropped frames	Banding/Quantization
Compression	Non-native aspect ratio	

which there exists a “golden” source video. Unfortunately, visual impairments can still be introduced due to human and systematic errors. The original video sources of older contents are sometimes lost, requiring a compromise between availability and quality of content. Ultimately, the highest possible source quality is ingested by streaming companies and delivered in a variety of formats, depending on client requirements.

In some cases, these source videos contain artifacts that, if accepted into the encoding pipeline, would yield video encodes with poor quality being distributed to consumers. These artifacts may appear as a result of how a content has been produced, stored, and/or manipulated. Simply knowing that a distortion is present opens a path to remediation. Common artifacts that get introduced into the video source during production and storage include upscaling, video hits, frame drops, banding (false contours from quantization), incorrect aspect ratio, among many others [8]. By detecting these artifacts and measuring their perceptibility, sources can be considered on a case-by-case basis, based on the degree to which they might be distorted.

Typical artifacts that may occur in source videos are provided in Table 1.1. Combing / Blending occurs most often in sources derived from DVDs and from videos prepared for broadcast television. When the framerate is increased as in these scenarios, additional frames are introduced by interleaving or blending adjacent video frames, causing visible distortion on modern progressive displays. Upscaling may occur when spatially resizing a video source to match a particular larger frame size, encoding these interpolated pixels into the source. Video hits corrupt random blocks within one or more consecutive video frames, commonly caused by packet loss. Aliasing artifacts ("jaggies") appear after spatially downscaling videos without using a low-pass anti-aliasing filter, causing high-frequencies to interfere with low frequencies. Jaggies can also appear by upscaling using nearest neighbor interpolation. Banding, also known as "False Contouring," appears when the pixel values in a video frame are quantized, usually through compression, creating visible contours along smooth gradients. Non-native aspect ratio refers to the case when a frame is rescaled too far either vertically or horizontally, causing objects in a scene to distort in shape. Dropped frames artifacts are simply frames that are missing, perhaps from recording a network stream where frames were dropped to maintain a constant frame rate. Lastly, moderate to severe compression may be present in source videos, resulting from a lengthy re-transcoding process or incorrect selection of encoding parameters when the source is being produced. This list is not all-inclusive, but it does represent the types of distortions that are important and currently difficult to detect. A more comprehensive list is provided in Netflix's backlot pages [8].

A subset of these artifacts are shown in Fig. 4.2. Aliasing/jaggies can range in appearance from subtle to dramatic alteration of content, as



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

Figure 1.3: Examples of impairments that occur in source videos ingested by the streaming video industry. (a) Aliasing/jaggies; (b) MPEG2 hits; (c) H.264 hits; (d) Quantization; (e) False contours/banding; (f) Combing; (g) Upscaling; (h) Compression.

exemplified by Fig. (a). MPEG2 corruption produces small blocky artifacts, which can manifest as changes in the transform coefficient magnitudes, or in horizontal striping, as seen in Fig. (b). H.264 corruption rarely leads to horizontal striping, but often causes blocky impairments, as shown in Fig. (c). We regard quantization as a separate distortion than banding, which can arise in a variety of ways, and can manifest differently, as can be seen by comparing Figs. (d) and (e). Interlacing leads to "combing" artifacts, as depicted in Fig. (f). Upscaling is an often subtle artifact, which presents as a loss of detail as in the "nearest neighbor" upscaling shown in Fig. (g). Lastly, H.264 compression, which increases blockiness and reduces details, is depicted in Fig. (h).

It becomes clear that models developed need to be "No-Reference," meaning predictions are made with no knowledge of the original pristine video. Finding top-performing no-reference detectors for each of these artifacts is the primary objective of the source inspection problem. Once these detectors are identified, they can be leveraged to predict artifact severity as perceived by the viewer. Within the types of artifacts to be detected, some artifacts are less humanly perceptible than others, making study of subjective artifact severity worthwhile when curating large video collections. Automated no-reference inspection tools can assist and ideally replace manual video source inspection.

1.3 Contributions

An open question is: how do these artifacts impact the quality of experience of the video? Digital video collection curators perhaps care about different aspects, but one important trait in any domain is the overall apparent quality of each source video. Streaming companies such as Netflix, YouTube, and Hulu maintain such large collections that manual assessment of each video's quality is not practical. A first step in assessing that quality is to determine if there are any dominating distortions that can seriously impact a user's experience. Once the dominating distortions are determined, the degree of impact of that distortion on user experience can then be predicted. Detecting these distortions and subsequently assessing quality are tasks that should be automated, given the vast and increasing volume of purveyed video content.

In this age of deep neural networks, one may wonder whether large-scale, data-driven machine learning methods might be used. However, there are several problems with this. First, large amounts of perceptually labeled video data are not available for any kinds of distortions [148]. Second, even for still pictures, deep learning methods for quality assessment are developing slowly, impeded by a lack of subjective data at scale [60]. Thirdly, deep learning on video for any tasks is itself a nascent field, with serious solutions still several years away [57]. Fourthly, while deep networks can produce excellent results on databases, they require considerable tuning, and can produce unexpected results, which is not acceptable when streaming to tens of millions of viewers [33, 91]. Lastly, content providers may inspect hundreds of video masters every day, hence highly efficient, lightweight solutions are needed.

Toward this goal of automated artifact detection, we present a generalized system for detecting artifacts called the Video Impairment Detection MAP (VIDMAP). For a given source video, this system can detect and *localize* artifacts that may be present in a video. Only the data itself and a global label is used for training VIDMAP, and the model learns to find local evidence of an artifact based on the global label. The output of VIDMAP is a probability map of the distortions detected. This system and model will be further described in Chapter 3.

We release the LIVE Video Masters database, which can be used to assess performance of source inspection frameworks. It contains 384 total videos, each based on a set of source videos collected from the Netflix collection and various public domain archives. A total of 30 subjective opinions were collected per video. The subjective assessment and data analysis will be further described in Chapter 4.

Lastly, we also present Video Impairment Detection by SParse Error CapTure (VIDSPECT) to address both the video artifact detection and quality prediction tasks. VIDSPECT is a general two-stage artifact assessment framework that exploits sparse coding principles to learn a discriminative dictionary that can be used for detecting artifacts. These filters that are tuned for detection are shown to be effective when used for the subsequent video quality assessment task. This system and model will be further described in Chapter 5.

Chapter 2

Background

2.1 Image and Video Quality Assessment (I/VQA)

The word “quality” is touted as a measure of excellence, but there are several distinct meanings important to image and video processing. For instance, “perceptual quality” refers to the subjective quality of the stimuli as perceived by the observer. In contrast, “aesthetic quality” refers to the appreciation of beauty and possibly the artistic quality of the stimuli. Yet still, “objective quality” is the quality of the stimuli without observer opinion. Given the wide widespread adoption of images and videos, the human observer is the most important receiver of that content to consider. Perceptual quality captures this most important aspect, and the most successful perceptual quality algorithms model the same statistics within the human visual system.

The top Image Quality Assessment (IQA) and Video Quality Assessment (VQA) models can be assembled into three major categories. Full-Reference (FR) algorithms compare the distorted signal directly with the pristine reference signal to obtain the quality measurement. MOtion-based Video Integrity Evaluation (MOVIE) [112] is the leading FR VQA method, which uses a Gabor filterbank to compare signal responses between reference and distorted videos. The Multi-scale Structural SIMilarity (MS-SSIM) index [140] is one of several top performing FR IQA methods, which computes the similarity between source and reference images over multiple scales. Reduced-Reference (RR) algorithms compare a distorted video to some information regarding the original pristine video such as a imposed watermark or measured signal. The leading RR VQA and IQA methods are the Spatio-Temporal Reduced Reference Entropic Differences (ST-RRED) [125] and the Reduced Reference Entropic Differences (RRED) [124], which both measure entropy differences between reference and distorted signals after wavelet decomposition. No-Reference (NR) algorithms work using only the distorted video which may not even have a pristine original version, such as the case with source videos. The top NR VQA algorithm is Video BLind Image Integrity Notator using DCT-Statistics (Video BLIINDS) [108], which uses motion regularities and natural scene statistics as its foundation. Some top IQA no-reference algorithms include the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [82] and the Natural Image Quality Evaulator (NIQE) [81], which make different measurements on divisively normalized image coefficients.

The top general NR quality models utilize perceptually relevant Natural Scene Statistics (NSS) models, which describe statistical regularities arising in images and videos of real-world scenes, to predict perceptual quality. To predict the quality score of an image or video, NSS-based algorithms use ‘quality-aware’ features that capture statistical departures from pristine images and videos. These departures are defined as distortions. Quality-aware features are designed to be sensitive to a large or even unknown set of distortions, such as blur, noise, and blocking. Finally, these quality-aware features correlate well with human opinions of quality, allowing them to accurately predict quality when trained and evaluated using various image databases.

2.2 Real World No-Reference I/VQA Performance

The performance of IQA NR methods is good for databases with artificially generated artifacts, especially when images are singly distorted. Popular benchmark databases such as the LIVE Image Quality Database [117], the TID2008 Database [99], the TID2013 Database [98], and the CSIQ Database [64] each offer a set of pristine images and corruptions of those pristine images, where the images are affected by small number of distortions. However, performance degrades significantly when multiple distortions are present in an image [47]. This limitation is overcome with deeper feature representations, that can learn how these distortions interact to produce an expected quality score.

The CID2013 Database [135] and Live Challenge Database [48] were constructed to evaluate real-world image quality, which involve images that contain many types of distortions to varying degrees. The latest predictors that perform well on these databases capture deeper statistics, by incorporating local luminance, local contrast, and local structure information [67] [47].

Automated solutions are being developed and explored to meet the needs of video curators, who are concerned with visual artifacts and overall quality. The open source quality control tool for video preservation, VCQ, enables automated objective analysis of digitized video through multiple indicators, the results of which require interpretation by the user [131]. These indicators indicate possible abnormalities found in the input video signal, but lack the deeper statistics that artifact-based analysis requires.

2.3 Specialized Source Artifact Detectors

Previous work in source artifact detection has largely involved developing specialized distortion-specific algorithms. We will discuss these existing specialized detection methods for upscaling, combing, aliasing, false contours, dropped frames, video hits, and incorrect aspect ratio.

2.3.1 Upscaling

Upscaling artifacts often appear in videos, hence detecting them is of importance. Many video contents may be upscaled during post-production, transcoding, or to fit larger formats. Upscaling artifacts are produced by imputing missing information from surrounding pixel data. This data imputation happens, for example, during color interpolation (demosaicking) and when adapting images for higher resolution displays. Since data imputation does not add information, and usually involves interpolation, upscaled images tend to be smoother than their originals, with reduced high-frequency energy. Upscaling a patch effectively results in a lower dimensional data in a higher dimensional space.

Forensic scientists are interested in identifying doctored images and video [54]. To be able to place more confidence in image and video evidence, all traces of tampering should be detected. Often images are tampered with using upscale-crop-move manipulations. Among the many types of image/video artifacts that occur in doctored videos are those that arise from upscaling when either replacing or moving objects, or when placing one image within another. This nearly always leads to re-scaling the image or object.

Upscaling prediction algorithms exist for (1) finding image-based evidence of upscaling, (2) predicting the native resolution of an original image/video, (3) classifying the upscaling method by type, and (4) quantifying perceptible loss of quality. Most existing methods do not fully cover this problem space, instead being designed to solve (1) or (2).

For problem type (1), typical approaches include analyzing spatial covariance or using radon transform analysis [76] to design upscaling detectors. Periodicities introduced by common upscaling techniques have been deeply studied [46, 101, 107, 100, 134, 61]. For problem type (3), many frequency-based approaches have been developed that derive a closed form prediction, but more general energy falloff-based models have benefited from machine learning to better characterize differences amongst upscaling techniques [56] [44]. However, both the falloff observed in the frequency spectrum and the periodicities introduced by upscaling can be reduced intentionally, to fool existing models [62], or by standard compression techniques. Methods that rely upon the Discrete Fourier Transform (DFT) typically lose prediction power when handling upscaling ratios outside the range of 1x-2x [44] [96].

2.3.2 Combing

Legacy video content that was originally intended for viewing on older televisions often was often encoded in an interlaced mode, which provided both a way to modify the frame rate for the end user, and a means of achieving further compression of the video signal by exploiting the persistence of CRT displays. During frame rate conversion, interlacing can be used to interpolate frames by copying even rows from a previous frame, and odd rows from a next frame, then combining the even/odd fields. To achieving compression, only half of the video information is required at a given frame rate, since only the even or the odd rows of the current frame need be delivered. Methods for interlace detection involve comparing interpolated row values with previous row values, to find evidence that a subset of previous row values were used [5, 28].

Another type of artifact that afflicts videos is combing, which occurs when videos are represented in an interlaced form, where whole video frames are sequenced as “top-bottom” or “even-odd” frame pairs, each having half the rows (and requiring half the bandwidths). Since the even-odd frame pairs are slightly temporally displaced in time, then when they are reconstituted into whole (progressive) frames, combing artifacts may occur, particularly in regions where the video contains motion.

For combing (interlacing artifact) detection, existing detectors have utilized top-field-first (TFF) and bottom-field-first (BFF) information across several video frames to determine whether combing artifacts are present. For example, the interlace detector within FFmpeg [5] computes the ratio of TFF to BFF, and when this ratio exceeds a specified range, the three frames are flagged as interlaced. The AVISynth detector [4] uses the same ratio, but only analyzes frames where motion is detected. Baylon [28] introduced a zipper filter, which was used to detect the difference between TFF and BFF by looking at "zipper points," which are moving edges between frames that strongly exhibit the combing artifact. Each of these models requires more than one frame to affect detection, despite the fact that the combing artifact is present in a single frame. Slight modifications to these detectors are provided in [53, 97, 58].

2.3.3 Aliasing

A digital video may also be downscaled improperly, leading to visible aliasing artifacts. The frequency content of higher frequency bands must be appropriately reduced, otherwise it will wrap around onto lower frequency bands after downsampling, causing visible distortions. The visible manifestations of aliasing can be "jaggies," oscillating moire, or other content-dependent patterns, which can be visually annoying. Aliasing detection methods include Reibman and Suthaharan [103], who developed a Signal-to-Aliasing Ratio, which measures the components of image aliasing at points of high contrast, by computing the ratio between the estimated aliasing energy, and the image energy with the estimated aliasing energy removed. Coulange and Moisan

[40] developed an *a-contrario* model, by measuring suspicious co-localization of Fourier coefficients to build up evidence of aliasing. This model requires knowledge of the original resolution of the image in order to determine the co-localization. Lastly, Eunjung *et. al.* [34] 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.

2.3.4 "False Contours" and Banding

Video content can be compressed at any point in the production pipeline, with loss occurring during 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 [25] 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.* [75] 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.

2.3.5 Dropped Frames

When video content is delivered over a network, entire frames might be lost, resulting in dropped frames, i.e. the loss of one or more frames. Frame drops are most obvious when motion is present in a video, and produce the appearance of unnatural staggering of moving objects [141]. Upadhyay and Singh's method for detecting frame drops [132] extracts spatial entropy and content variation features from binarized frame differences, then uses them to predict frame drops using a Support Vector Machine (SVM). The earlier method in [144] applies thresholds on frame differences, then detects frame drops when the threshold is exceeded.

2.3.6 Video Hits

When a digital video is transmitted, transferred, or stored, it might be re-encoded multiple times, often at relatively low levels of compression. Unfortunately compression artifacts can noticeably compound, and encoding and video packet errors can occur before, during, and after transmission. Also, digital tapes, which are commonly used to transport video content, might introduce corruption, depending on the environmental conditions. These corruptions, commonly called video hits, may appear as single corrupted blocks or as groups of corrupted blocks that persist for several seconds. Methods for detecting packet loss, both with and without concealment, usually operate by detecting sharp edges near block boundaries, whose locations are defined by the coding standard that was used [128, 115]. These methods only work if the structural information loss can be modeled. Winter *et. al.* [142] acknowledged that this structure is often unknown, especially on analog recordings. They provided an alternative row change measure and an edge ratio measure, that when used in conjunction, define a video hit detection mechanism.

2.3.7 Incorrect Aspect Ratio

Some video sources are packaged with aspect ratio metadata, that if ignored or unsupported by the decoder, will result in improper encodes. This will cause the final aspect ratio to be incorrect. Common aspect ratios observed at ingest are 16:9 high-definition sources, and older 4:3 standard-definition sources. Detecting incorrect aspect ratios has been attempted using convolutional neural networks [109] for the purpose of correcting it.

2.4 Sparsity and Natural Scene Statistics

Toward the goal of automation, we leverage natural scene statistics (NSS) models going forward. NSS models seek to capture the regularities that exist in the statistics of images of the physical world. NSS tend to be disrupted by visual artifacts, making them powerful tools for video inspection. Generalized artifact detection is related to anomaly detection and saliency. If the data distribution of a video signal is properly described, then anomalous patterns produce deviations from the NSS model [30] that can be identified. A variety of state-of-the-art picture quality prediction models [31] [86] such as BRISQUE [82], NIQE [81], and FRIQUEE [47] model the statistical regularities of natural images and videos, then assess distortions that disturb these regularities. More recently, general no-reference video quality prediction models have been developed, including Video BLIINDS [108], VIIDEO [80], Li *et. al.* [69], and Shabeer *et. al.* [114].

Chapter 3

VIDMAP: Video Impairment Detection MAP

For generalized artifact detection, we present the Video Impairment Detection MAPper (VIDMAP), which can both detect and *localize* most of the artifacts described in Chapter 1, without need for a reference video. We evaluate detection performance of VIDMAP on several artifact detection tasks and compare that performance against competing methods. We show that VIDMAP is a state-of-the-art detector of most artifact types, and is highly competitive otherwise, even using the same network architecture across all distortions. [1]

Within the VIDMAP model, we make the following contributions:

- The VIDMAP framework (Section 3.1) designed for use with ingested video sources that automatically detects possible distortions and assigns that video for further processing if a distortion is detected.

[1] This chapter appears in the following papers: T. R. Goodall and A. C. Bovik, "Detecting and Mapping Video Impairments" submitted to the IEEE Transactions on Image Processing, 2018; and T. R. Goodall and A. C. Bovik, "Artifact Detection Maps Learned Using Shallow Convolutional Networks." Southwest Symposium on Image Analysis and Interpretation, 2018. Todd Richard Goodall has designed the models, collected data, and performed full experimental analysis of the works described therein.

- An associated convolutional network (Section 3.2.2) designed to detect and localize artifacts.

- Performance comparisons (Section 3.4) between the VIDMAP detection framework and competing methods using both synthesized and real artifact data, showing VIDMAP as either superior to or at least highly competitive with leading distortion-specific models.

- A publicly available package of this framework is provided at [16], which includes the trained weights for the upscaling, combing, false contours, quantization, aliasing, video hits, compression, and dropped frames distortion categories.

We believe that no high-performance, practical video source inspection system similar to VIDMAP exists.

3.1 VIDMAP System

We present the VIDMAP system in Fig. 3.1, which incorporates pre-processing stages and convolutional network sub-components. For each artifact, the convolution network is rerun with artifact-specific weights. Any detections are then aggregated and delivered to the next stage, a quality assessment stage tasked with making a final decision regarding that input video content. This last stage may be fully automated using quality assessment prediction models, or it may be manual, depending on the level of scrutiny desired. If no artifacts are detected, then the video is deemed to be free of artifacts.

3.2 Models

3.2.1 Pre-Processing Model

Before applying VIDMAP processing to videos during either training or testing, the video is pre-processed by center-surround, isotropic bandpass filtering, followed by a non-linear divisive normalization process [106]. 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 [106, 82, 81]. The MSCN coefficients of image I are given by

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

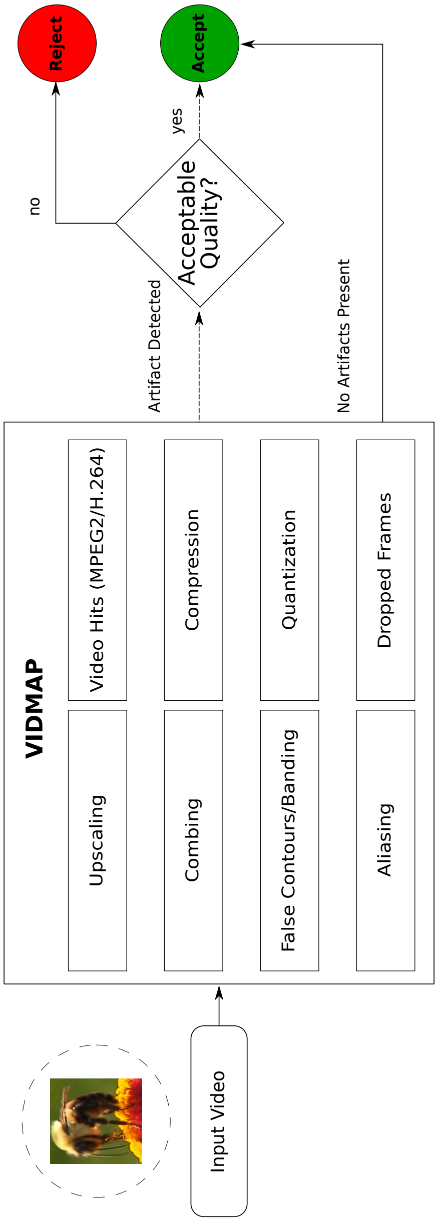


Figure 3.1: VIDMAP system design. An input video is submitted to VIDMAP for artifact analysis. If an artifact is detected, the video is flagged for either manual or automatic quality assessment. Videos with an acceptably low number of artifacts can be ingested. Otherwise, the video is rejected.

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.

The MSCN pre-processing stage reflects both a well-established NSS model [106], as well as simple center-surround retinal processing [31]. The BRISQUE IQA model [82] 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 [47] and NIQE [81] image quality models. We have found that using both the sigma field and the MSCN transformed image improve the prediction power and thus the generalizability of the VIDMAP model.

3.2.2 Convolutional Detection Map Network

3.2.2.1 Version 1

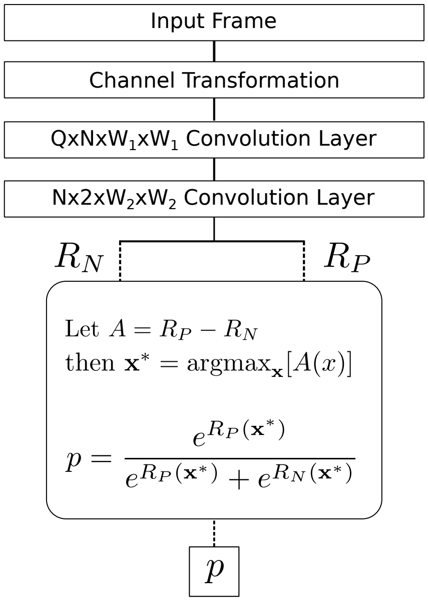

Input Frame
Channel Transformation
QxNxW$_1$xW$_1$ Convolution Layer
Nx2xW$_2$xW$_2$ Convolution Layer
R_N
R_P
Let $A = R_P - R_N$
then $\mathbf{x}^* = \text{argmax}_{\mathbf{x}}[A(x)]$
$p = \frac{e^{R_P(\mathbf{x}^*)}}{e^{R_P(\mathbf{x}^*)} + e^{R_N(\mathbf{x}^*)}}$
p

Figure 3.2: VIDMAP convolutional network architecture in the first configuration. 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 [39] (not shown) are present at the convolution layer outputs.

A visual summary the first version of the VIDMAP artifact detection network is provided in Fig. 3.2. 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) [39] 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

$$\bar{y}(\mathbf{x}) = \frac{e^{R_P(\mathbf{x})}}{e^{R_P(\mathbf{x})} + e^{R_N(\mathbf{x})}}, \tag{3.1}$$

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.

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. 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 [26], which use class labels at each coordinate of the training image. The dotted lines in Fig. 3.2 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 like VGG, [122] which can have greater than 100 million parameters. 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 pre-processing that computes the MSCN inputs. Without this pre-processing step, the network takes much longer to converge and performance suffers. 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 experiments related to this version of the convolution network.

As we will show, this first version yields great predictors of upscaling and combing. It does not, however, produce great predictors of video hits or dropped frames. For this reason, we extend this architecture in the next section.

3.2.2.2 Version 2

A visual summary of the second version VIDMAP artifact detection network is provided in Fig. 3.3. As before, each input frame is pre-processed into $Q = 2$ channels, the MSCN coefficients and $\sigma(\mathbf{x})$ maps. This network can accommodate multiple frames, by pre-processing individual frames, then concatenating these independent processed channels into a single multi-channel input frame. When N frames are input to the network, we set $Q = 2N$. Alternatively, the input frames may be differenced before the pre-processing stage, which would require setting $Q = 2(N - 1)$. Each frame or sequence of frames input to the network is reorganized and pre-processed into a single multichannel input before being applied to the network.

The first layers after pre-processing include both convolutional and bias weights. The size of the internal representation, i.e., the number of output channels for the first layer, is fixed at $N = 100$. Following this layer is an Exponential Linear Unit (ELU) [39] activation function, which avoids neuron death associated with the ReLu, while also reducing training time. The following two layers in both branches perform identical operations, albeit with different channel configurations, such the final output of both branches is a single response map. The lower path output labeled R_N serves as the inhibitory (negative) response, while the upper path output labeled R_P is the excitatory (positive) response. A final probability prediction map is formed using Eq. (3.1).

As in version 1, the ground-truth labels used to train 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.

Although propagating error through $\mathbf{x}^*$ as previously described was found to produce excellent performance, we found that the resulting probability maps did not label many distorted regions. This is a phenomenon similar to that observed by Singh and Yee [123], who proposed randomly hiding the most discriminative data during training. We tried this by sampling different discriminative points, which improved training time, but 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 find a discriminative point that takes a neighborhood of responses into account, and second, to backpropagate error through more than one point in the map. In some cases this improves the overall detection performance of VIDMAP, but in all cases it produces more complete probability maps.

The only learned parameters in the network are the convolutional and bias weights. The first layers contain $2N(QW_1^2 + 1)$ free parameters, the second layer contains $2N(NW_2^2+1)$ free parameters, and the last layer contains $2(NW_3^2+1)$ free parameters. We found that setting $N = 100$ and $W_1 = W_2 = W_3 = 11$ provided excellent generalizable performance, meaning there are about 2 million parameters. The complexity of version 2, like version 1, is still much lower than recent deep convolutional algorithms [122].

3.3 Dataset Preparation

3.3.1 Synthesized Datasets

We created a separate dataset for each artifact type: upscaling, video hits (MPEG2), video hits (H.264), aliasing, banding, false contours, interlacing, frame drops, and compression. 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 contents. We identified scene boundaries using [93], 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.

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.

Upscaled video was produced by using one of "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," or "Nearest Neighbor Upscaling." 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, 3.0]. We collected a total of 202,752 samples.

Quantized video was produced by first selecting a $q \in \{2, 4, 8, 16, 32\}$, 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. (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. (b). The negative samples in this contour dataset were supplemented with pristine video data. The final dataset contained 558,100 100x100 samples.

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.

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 with small TI [141]. A total of 63,030 samples were generated in this way.

Two video hits datasets were created, based on corrupting MPEG2 or H.264 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 H.264 and 30,043 MPEG2 hit samples were generated.

The compression dataset was created by considering the H.264 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.

3.3.2 Non-synthesized Datasets

While generated datasets provide an excellent baseline for how VIDMAP captures different artifacts, it is unclear how VIDMAP performs on data that is not synthesized. Videos were collected from Netflix that exhibited combing artifacts. For videos with these artifacts, we noticed that some contained additional compression from the transcoding process. We also gathered an dataset comprised of videos with aliasing and "jaggies" artifacts. We discovered that "jaggies" can appear even when a video signal is not aliased (i.e. in the case of upscaling or leftover patterns from a de-interlacing algorithm).

3.4 Artifact Detection Results

Table 3.1: Upscaling detection F1 scores computed on the test set for VIDMAP version 1. Upscaling type includes "Not Upscaled," "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," and "Nearest Neighbor Upscaling."

Algorithm	Bilinear	Bicubic	Lanczos	Nearest
VIDMAP	**0.9902**	**0.9916**	**0.9915**	0.9932
Vázquez-Padín [133]	0.9736	0.9706	0.9683	0.9929
Goodall *et al.* [50]	0.9872	0.9885	0.9941	**0.9977**
BRISQUE [82]	0.9331	0.8988	0.8847	0.8847
Feng *et al.* [44]	0.8609	0.9162	0.9577	0.9099

We evaluated two types of problems using version 1 of the VIDMAP network: the video upscaling (interpolation) detection and the combing (interlacing artifact) detection problems.

For upscaling, we numerically evaluated the performances of the models using the F1 score, which is the harmonic mean of precision and recall. Table 3.1 compares the performance of VIDMAP 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.* [44].

Table 3.2: F1 scores achieved by the compared combing detection models on the set of 150 video sequences for VIDMAP version 1.

Algorithm	F1
VIDMAP	**0.9868**
BRISQUE [82]	0.8718
FFmpeg	0.9167
Baylon [28]	0.8811

For combing, we also evaluated two existing state-of-the-art algorithms. The first is the FFmpeg 'idet' detector, which requires 3 frames. For progressive video, it assumes that the row in the current frame can be interpolated using two rows in either the previous or next frame. For interlaced video, it assumes the interpolated row will not match the corresponding row in the previous or next frames. A prediction is generated by applying threshold T_1 on these two measurements.

The second algorithm was developed to determine field order on known combed sequences [28]. We modified it to provide detection predictions. It counts the number of zipper artifacts T_0 of length Z in the top-field and the bottom field between two frames. If the difference between these counts exceeds a threshold T_1, then the two frames are labeled as combed. Thus, this algorithm requires two frames for detection. Both of these algorithms are provided in Appendix B.

Table 5.13 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.

Table 3.3: Detection results on validation sets. Top performers in boldface.

Distortion Category	Method	F1 Score	MCC
Upscaling	VIDMAP	**0.9899**	**0.9799**
	VIDMAP-D	0.9767	0.9549
	Goodall [50]	**0.9865**	**0.9728**
	BRISQUE [82]	0.9597	0.9185
	Feng et. al. [44]	0.8713	0.7330
	Vázquez-Padín *et al.* [133]	0.9767	0.9533
False Contours	VIDMAP	0.9762	0.9529
	BRISQUE [82]	**0.9996**	**0.9993**
	Luo *et. al.* [75]	0.9606	0.9240
	Ahn and Kim [25]	0.8554	0.7033
Quantization	VIDMAP	**0.9944**	**0.9887**
	VIDMAP-D	0.9753	0.9504
	BRISQUE [82]	**0.9954**	**0.9909**
	Luo *et. al.* [75]	0.9903	0.9806
Compression	VIDMAP	**0.9790**	**0.9580**
	VIDMAP-D	0.9487	0.8961
	BRISQUE [82]	**0.9765**	**0.9528**
	Luo *et. al.* [75]	0.8422	0.6708

Inspired by the performance of VIDMAP version 1, we evaluated the performance of VIDMAP version 2 against state-of-the-art methods on the aforementioned datasets in Section 3.3.1. Our evaluation included measuring the errors between predictions and ground truth binary labels, hence we assessed the binary classification to VIDMAP F1 scores, the harmonic mean between precision and recall, and Matthew's correlation coefficient (MCC), which is a balanced measure related to the chi-square statistic. Tables 3.3 and 3.4 list the performance results, where VIDMAP refers to VIDMAP version 2 performance using only single frames, and VIDMAP-D refers to VIDMAP performance using frame differences.

For the upscaling detection problem, VIDMAP using single frames matched the top performance of a recent sparsity-based model. BRISQUE performed surprisingly well on upscaling, although it was designed for artifacts that would affect an observer roughly 2 feet from the display. Upscaling factors smaller than 2 produce distortions that are difficult to see, especially when using Lanczos-4 interpolation.

A trivial quantization 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. We found that BRISQUE was able to effectively detect the presence/absence of quantization. VIDMAP also produced excellent quantization detection performance, with the ability to also localize areas of visible quantization artifacts.

On the detection of false contours, we observed that VIDMAP was slightly outperformed by BRISQUE. This is likely because the false contour detection problem is a subset of the quantization problem, which is easily detected in the spatial domain. We noticed that 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. 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 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.

On aliasing artifacts, VIDMAP delivered superior detection performance. The competing compared method, which uses the Signal-to-Aliasing ratio to measure aliasing, involves several steps that depend on implementation details that were not specified, such as energy masking parameters. As a result, the performance could possibly be improved. We retrained VIDMAP on a collection of 2000 video patches exhibiting jaggies, since we suspected that jaggies were not produced only by aliasing artifacts. A test set of 100 negative video segments and 51 positive video segments were used. Detection was performed per-frame, then averaged. To classify a segment as distorted in VIDMAP, the detection probabilities are averaged across frames, then a threshold on this average is learned using a separate validation set to binarize the final output detection prediction. Detection results on non-synthesized data are provided in Table 3.6. Again, VIDMAP and VIDMAP-D yielded superior performance.

Combing manifests as a vertical zipper artifact. VIDMAP again produced the best results, but 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. We supplemented the analysis with respect to the combing artifact by creating a distinct dataset from videos exhibiting artifacts found in a real video collection. As in aliasing case, per-frame predictions are averaged, then an entire video segment is classified as exhibiting combing by using a threshold learned on a validation set. Table 3.5 lists performance results, evaluated using 271 interlaced segments and 285 non-interlaced segments. VIDMAP outperforms the other methods. Interestingly, FFmpeg suffers most on real data.

When detecting dropped frames, both individual frame-based and frame-difference methods worked well. Upadhyay and Singh's detector gave the best results, using a threshold value of 30 in their algorithm in the frame-difference binarization step. The default parameters in the model suggested by Wolf yielded inadequate performance on the Netflix dataset. Surprisingly, BRISQUE also performed well for this task.

By defining video hits as corruptions that can change the bits in a stream at any point in the production process, the block positioning imposed by an intermediate codec is generally unknown, due to cropping and repositioning during the editing process. We found that VIDMAP produced the best detection results, and was able to capture the locality of the artifacts. Glavota *et. al.*'s features, which measured statistics related to structured block sizes of 8x8, 16x16, and 32x32 pixels, performed quite well when fed into an SVR for prediction. There is a gap in performance for BRISQUE between detection of H.264 versus MPEG2 artifacts. This is likely due to how the dataset was constructed, whereby H.264 artifacts were more numerous and more uniformly distributed across each frame, while the MPEG2 artifacts were fewer and much more isolated.

Compression artifacts were detected well by both VIDMAP and BRISQUE. Luo *et. al.*'s method that worked well for detecting quantization-based artifacts, does not work provide similar performance for compression.

3.5 Artifact Detection Maps

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. Figure 3.5 demonstrates predicted corruptions on exemplar H.264 and MPEG2 streams. Notice that nearly all of the visible artifacts are highlighted. Figure 3.6 shows detection of the combing artifact, where the map appears to capture all visible portions of the artifact. Notice that the detection probabilities nicely fall along the (gray) contours where the smoother content is free of combing. The aliased regions in Fig. 3.7

are detected with high certainty along edges. Figure 3.8 depicts the detection of false contours on a frame with film grain noise that was quantized. The contour lines were largely captured. As shown in Fig. 3.9, 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 3.10 depicts detection of H.264 compression artifacts. VIDMAP does not seem to measure edge strength, but rather characteristic smoothness in low contrast regions. Figure 3.11 depicts the results of several upscaling interpolation methods and corresponding artifact maps computed on a video of a traffic cone. The upscaling artifacts were easily detected. Figure 3.12 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.

3.6 Discussion and Conclusion

We proposed a new video source inspection system called VIDMAP, which is able to effectively learn how to detect and localize multiple types of video artifacts without using *a priori* models of the statistics or structures of the artifacts. We showed that VIDMAP achieves state-of-the-art detection performance in most categories tested, with competitive performance in the others. It is a practical tool that also assists a user in visualizing distortion types, locations, and severities. We envision that this model will be useful as a tool for source inspection of streaming video collections.

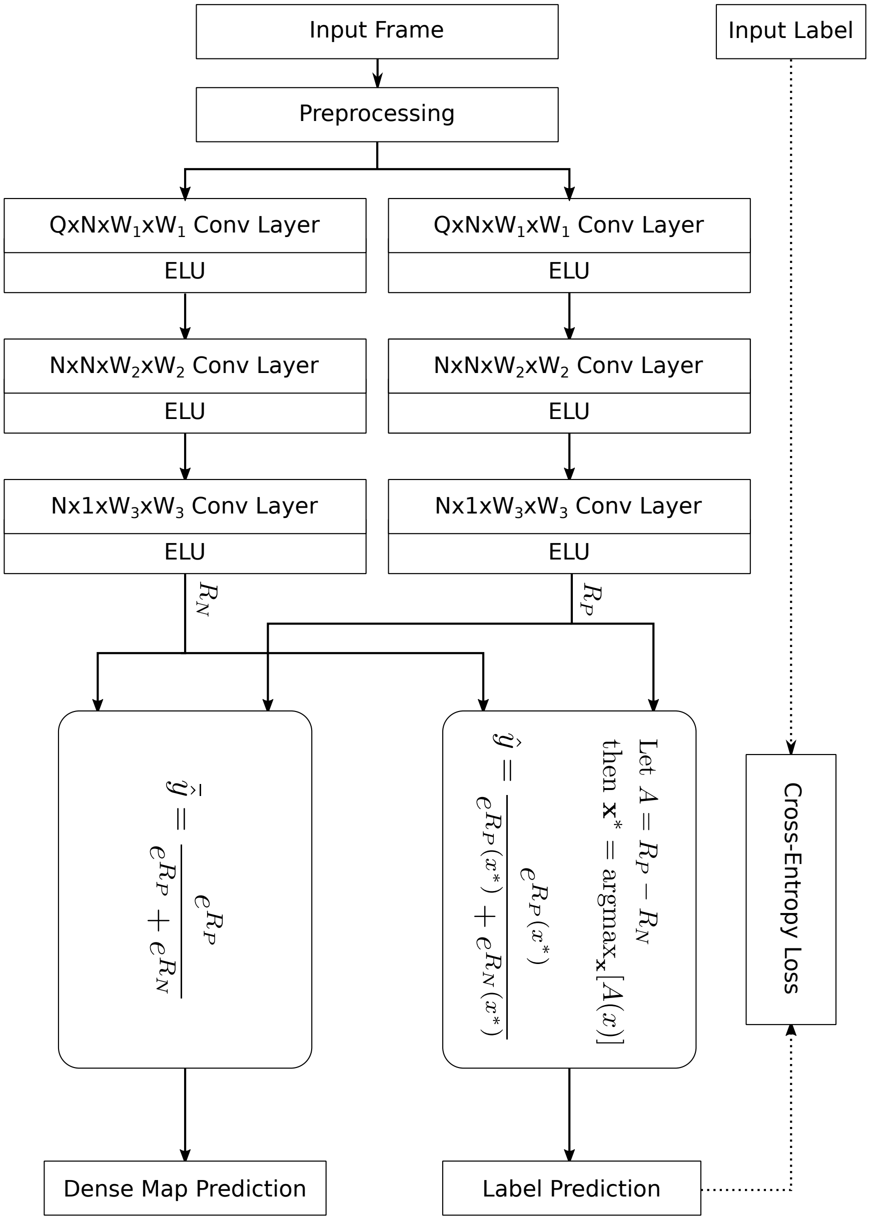
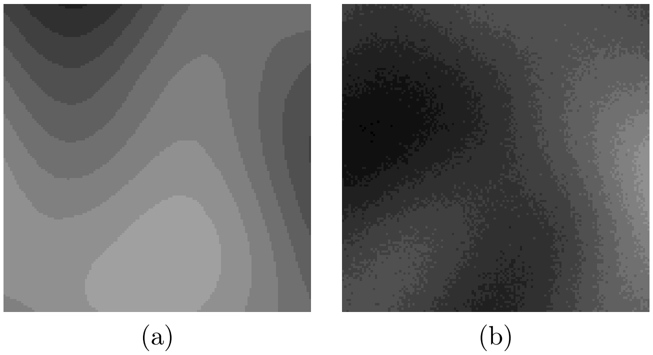

Input Frame
Input Label
Preprocessing
QxNxW$_1$xW$_1$ Conv Layer
ELU
QxNxW$_1$xW$_1$ Conv Layer
ELU
NxNxW$_2$xW$_2$ Conv Layer
ELU
NxNxW$_2$xW$_2$ Conv Layer
ELU
Nx1xW$_3$xW$_3$ Conv Layer
ELU
Nx1xW$_3$xW$_3$ Conv Layer
ELU
R_N
R_P
$\bar{y} = \frac{e^{R_P}}{e^{R_P} + e^{R_N}}$
Let $A = R_P - R_N$
then $\mathbf{x}^* = \text{argmax}_{\mathbf{x}}[A(x)]$
$\hat{y} = \frac{e^{R_P(x^*)}}{e^{R_P(x^*)} + e^{R_N(x^*)}}$
Cross-Entropy Loss
Dense Map Prediction
Label Prediction

Figure 3.3: Second version of the VIDMAP convolutional network architecture. Dotted lines indicate the portion of the network that exists only for training. No loss is propagated through the dense map prediction. The preprocessing layer computes σ and MSCN coefficient maps. Each input frame has a single associated binary label indicating whether the frame is distorted or not. Exponential Linear Units [39] are present at all convolution layer outputs.



(a) (b)

Figure 3.4: Examples of generated false contours. (a) False contours without noise; (b) False contours with noise.

Table 3.4: Detection results on validation sets. Top performers in boldface.

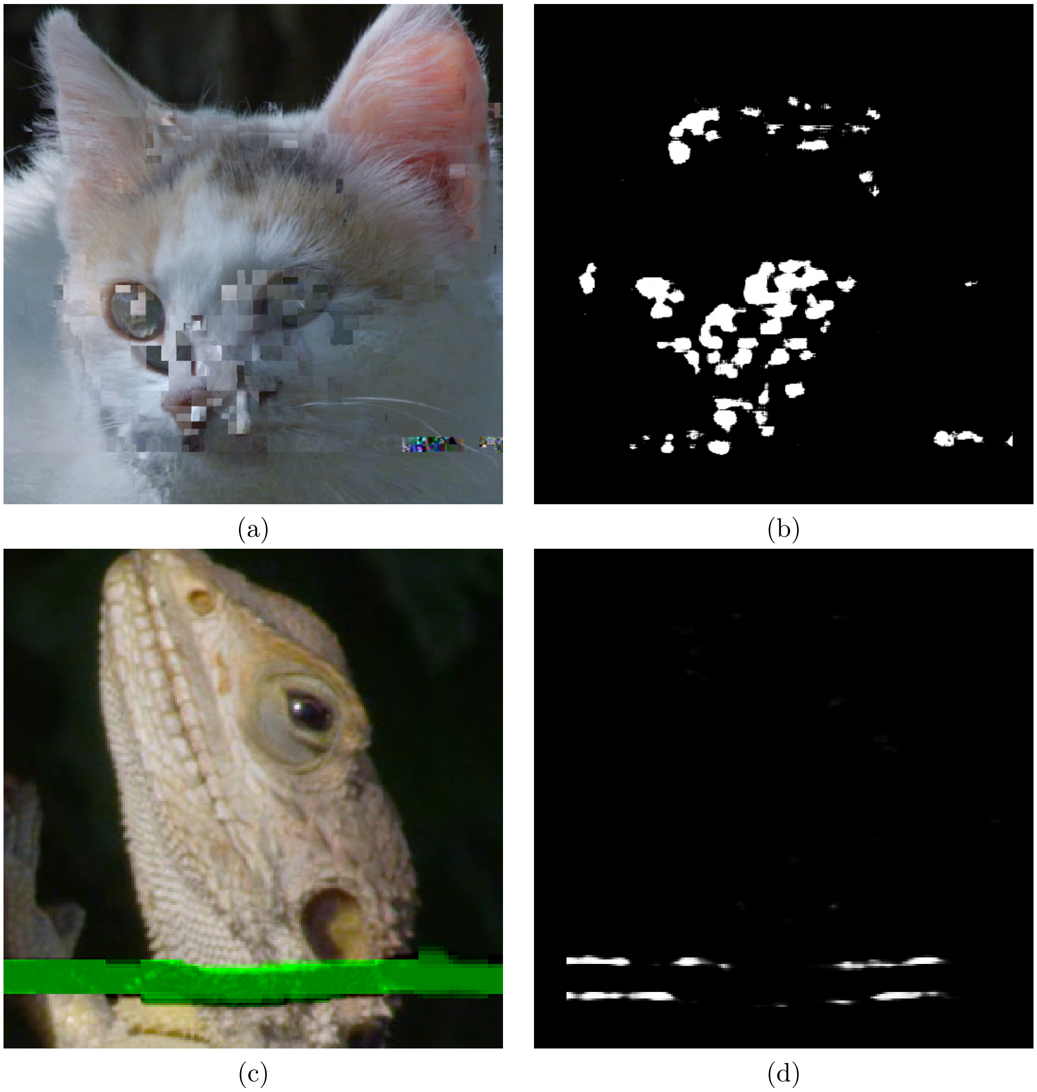
Distortion Category	Method	F1 Score	MCC
Aliasing	VIDMAP	**0.9728**	**0.9451**
	VIDMAP-D	0.9531	0.9056
	BRISQUE [82]	0.9615	0.9230
	Signal-to-Aliasing Ratio [103]	0.6859	0.2223
Combing	VIDMAP	**0.9693**	**0.9388**
	VIDMAP-D	**0.9682**	**0.9360**
	BRISQUE [82]	0.9599	0.9196
	FFmpeg [5]	**0.9645**	**0.9288**
	Baylon [28]	0.9288	0.8562
Dropped Frames	VIDMAP	0.9355	0.8687
	VIDMAP-D	0.9147	0.8250
	BRISQUE [82]	0.9142	0.8249
	Upadhyay and Singh [132]	**0.9510**	**0.9007**
	Wolf [144]	0.6827	0.2406
Hits (H264)	VIDMAP	0.9323	0.8672
	VIDMAP-D	**0.9406**	**0.8856**
	BRISQUE [82]	0.8273	0.6467
	AIDB [115]	0.7342	0.4867
	Glavota *et. al.* [49]	0.8794	0.7777
	Winter *et. al.* [142]	0.5521	0.2059
Hits (MPEG2)	VIDMAP	**0.9083**	**0.8193**
	VIDMAP-D	0.8734	0.7716
	BRISQUE [82]	0.6342	0.2959
	AIDB [115]	0.6413	0.3124
	Glavota *et. al.* [49]	0.8024	0.6296
	Winter *et. al.* [142]	0.5159	0.1070

Table 3.5: Detection results on videos exhibiting combing artifacts.

Method	F1 Score	MCC
VIDMAP	**0.9304**	**0.8663**
VIDMAP-D	0.8676	0.8055
BRISQUE [82]	0.9065	0.8141
FFmpeg [5]	0.9154	0.8316
Baylon [28]	0.8535	0.7122

Table 3.6: Detection results on videos exhibiting aliasing/jaggies artifacts.

Method	F1 Score	MCC
VIDMAP	**0.8807**	**0.8179**
VIDMAP-D	0.8772	0.8156
BRISQUE [82]	0.8571	0.7818
Signal-to-Aliasing Ratio [103]	0.5333	0.1758



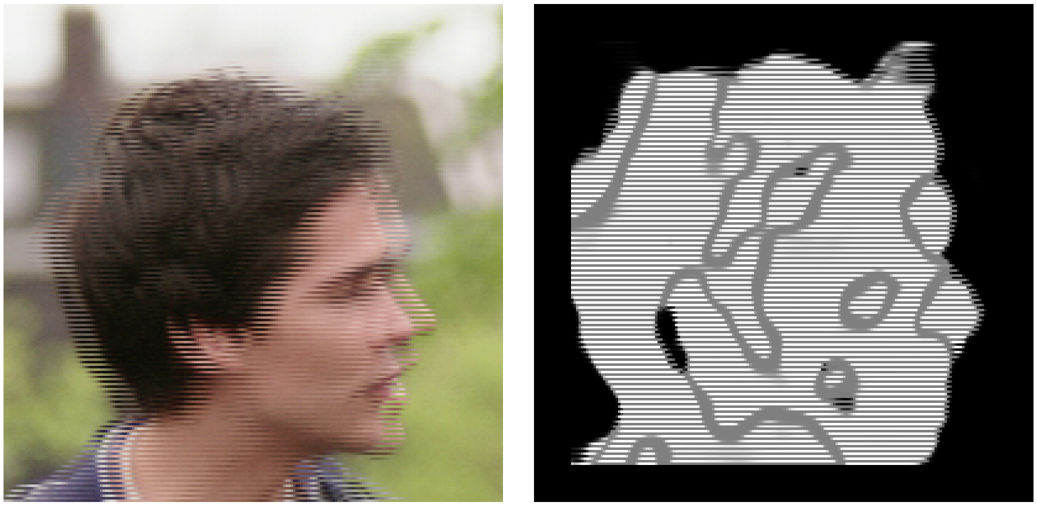
(a)

(b)

(c)

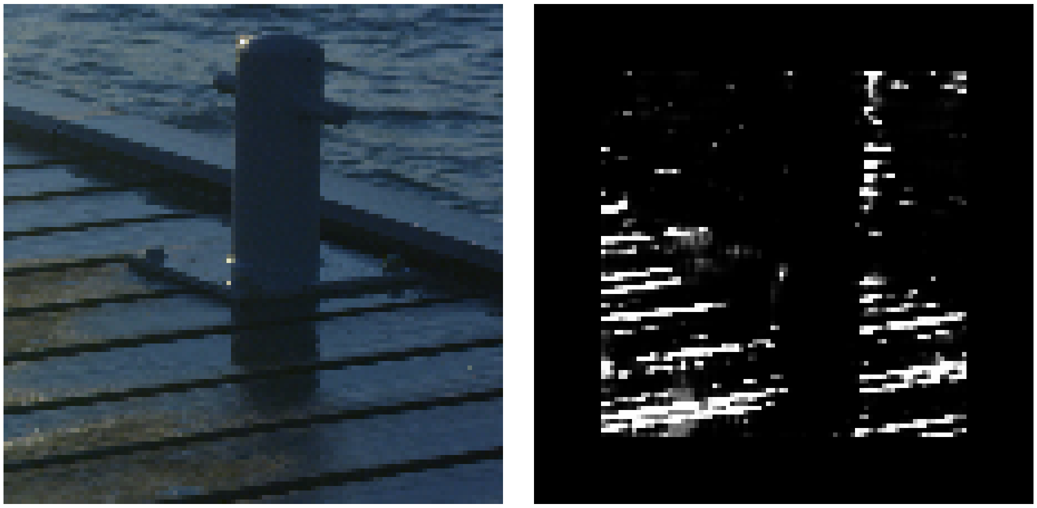
(d)

Figure 3.5: Video Hits Impairment Maps. (a) Video frame with H.264 video hits; (b) VIDMAP visualization of (a); (c) Video frame with MPEG2 video hits; (d) VIDMAP visualization of (c).



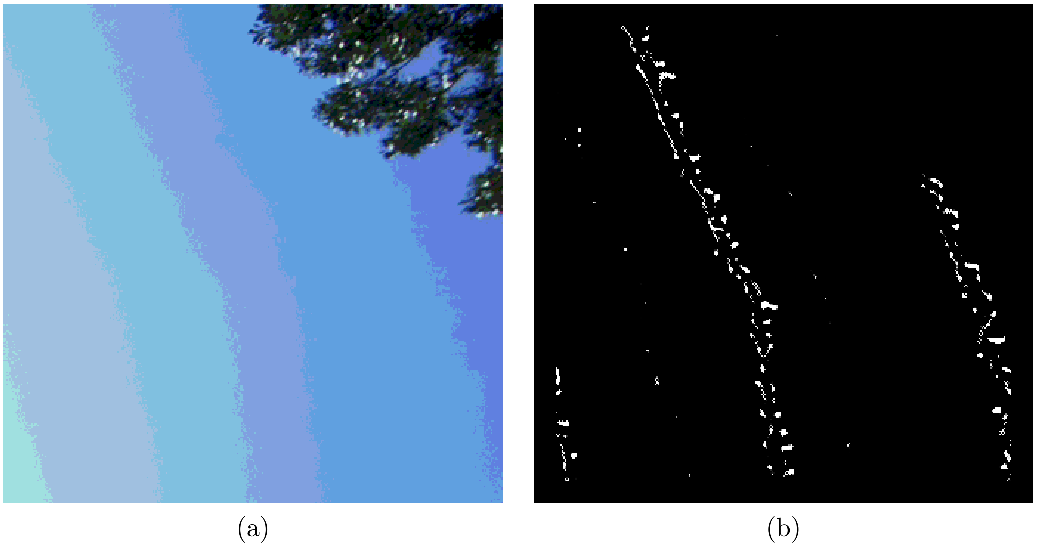
(a) (b)

Figure 3.6: Combing impairment map. (a) Video frame with combing distortion; (b) VIDMAP visualization of (a).



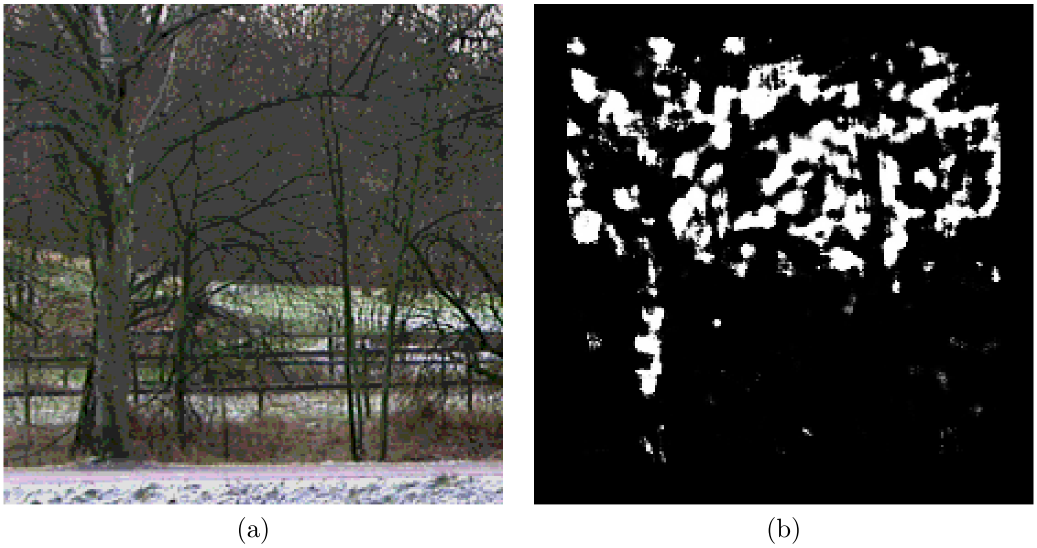
(a) (b)

Figure 3.7: Aliasing impairment map. (a) Video frame with aliasing distortion; (b) VIDMAP visualization of (a).



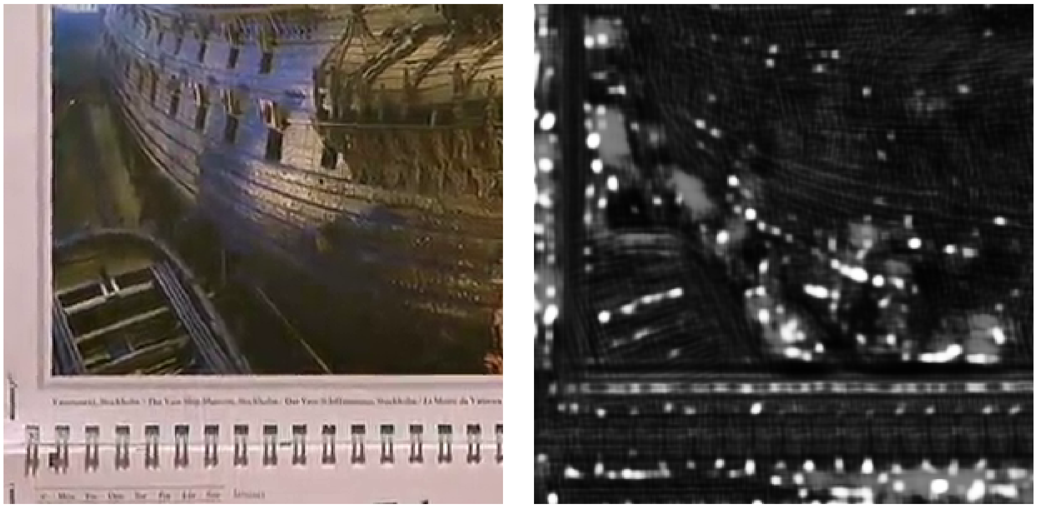
(a) (b)

Figure 3.8: False contour impairment map. (a) Video frame with false contour distortion; (b) VIDMAP visualization of (a).



(a) (b)

Figure 3.9: Quantization impairment map. (a) Quantized frame; (b) VIDMAP visualization of (a).



(a)

(b)

Figure 3.10: Compression impairment map. (a) Compressed frame; (b) VIDMAP visualization of (a).

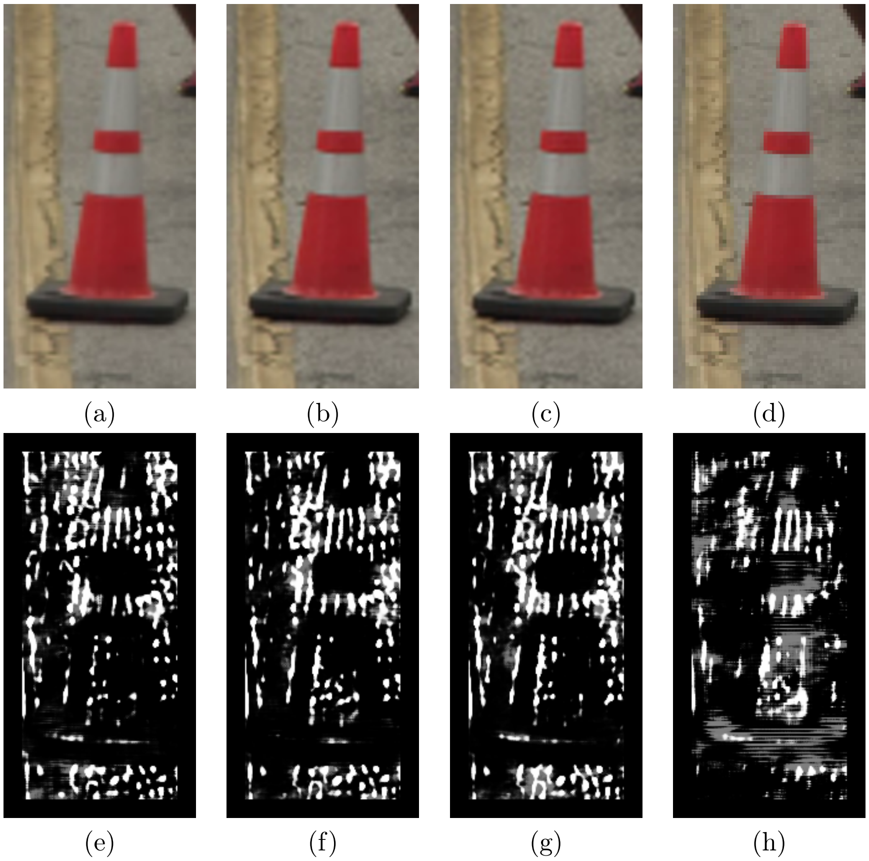
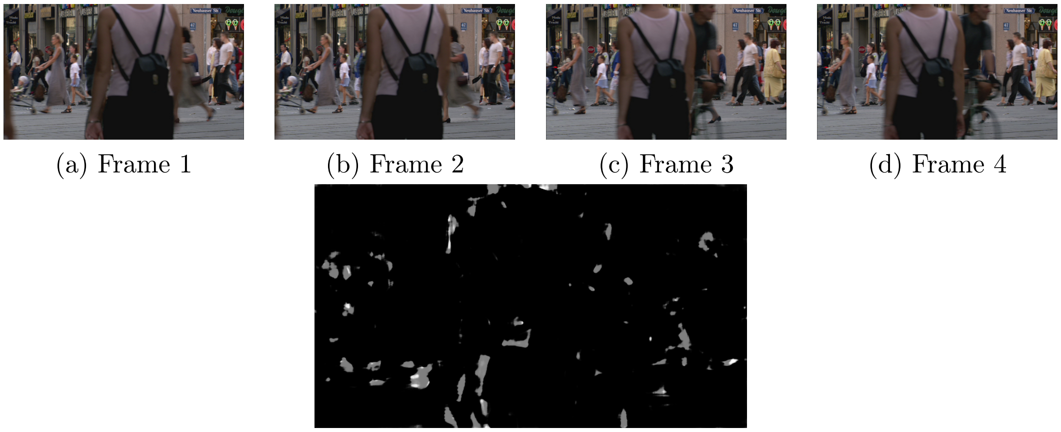


Figure 3.11: Upscaling impairment maps. (a) Bilinear upscaled; (b) Bicubic upscaled frame; (c) Lanczos upscaled frame; (d) Neighbor upscaled frame; (e) VIDMAP visualization of (a); (f) VIDMAP visualization of (b); (g) VIDMAP visualization of (c); (h) VIDMAP visualization of (d).



(a) Frame 1

(b) Frame 2

(c) Frame 3

(d) Frame 4

(e) Impairment map

Figure 3.12: Dropped frame impairment map. The drop of 9 frames occurred between frames 2 and 3.

Chapter 4

LIVE Video Masters Database

4.1 Video Synthesis

A sizable database of human subjective opinion scores is required for validating automated quality assessment methods on source videos. Towards this end, we collected a total of 24 high-quality reference videos from both public sources and from the closed Netflix collection, which was used as a cinematic content resource. We found that obtaining interesting high-quality cinematic video content from public sources was difficult. Representative thumbnails of the public content is provided in Fig. 4.1. Each video content is 10 seconds long, while the observed video frame rates included 23.98, 24, 25, 30, and 59.94 frames per second.

We distorted each of the pristine source videos using a total of 6 different distortion types. These types include "Video Hits (H.264)," "Video Hits (MPEG2)," "Upscaling," "Banding," "Dropped Frames," and "Incorrect Aspect Ratio." We produced H.264 and MPEG2 video hits by corrupting a 2 second Group of Pictures (GoP) that was randomly selected within the middle portion of an input video. To cause corruption, we used FFmpeg's '-bsf' noise flag to generate two severities of H.264 and two severities of MPEG2 hits distortions.

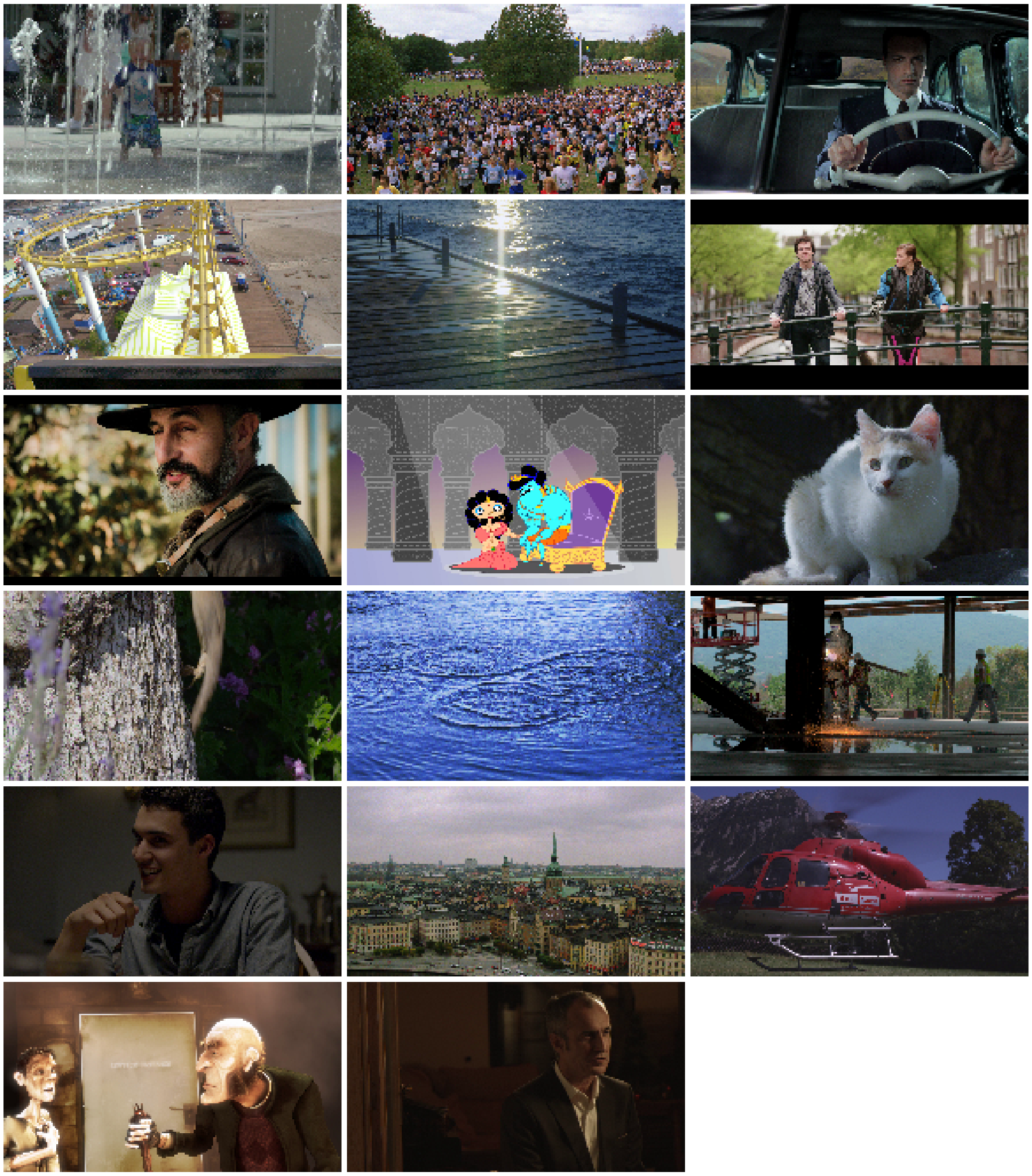


Figure 4.1: Thumbnails of free contents in Video Masters database.

To create videos having incorrect aspect ratios, we considered two extremes: stretching the video width by 25%, and shrinking the width by 25%. For the

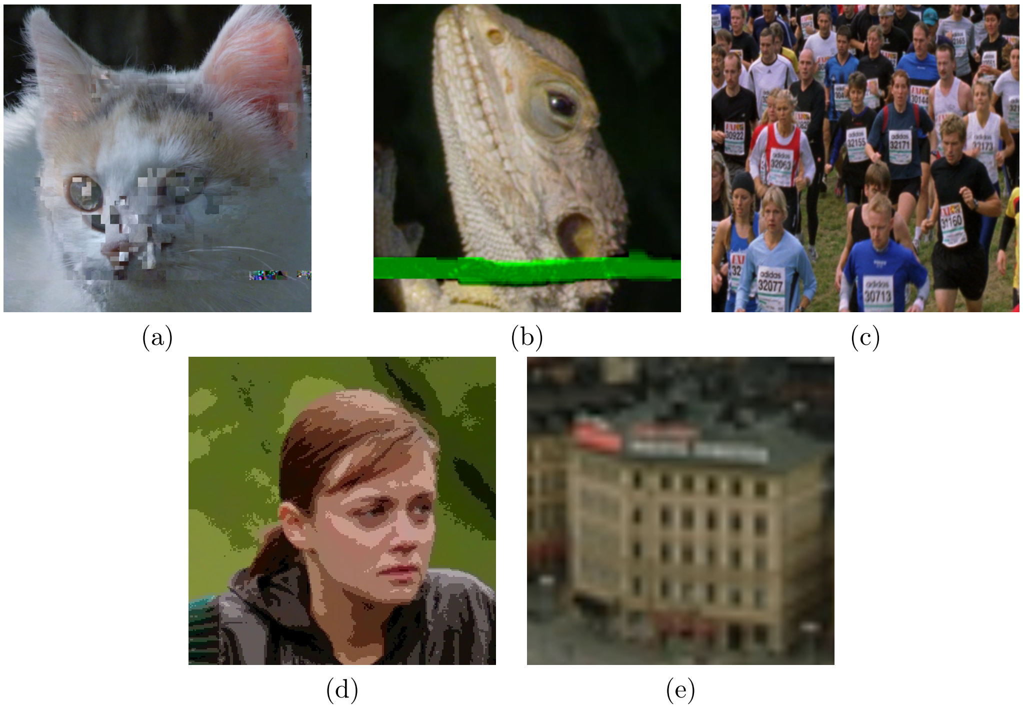


Figure 4.2: Examples of impairments that occur in source videos ingested by the streaming video industry. (a) H.264 hits; (b) MPEG2 hits; (c) Incorrect Aspect Ratio; (d) Quantization; (e) Upscaling.

upscaling distortion, we upscaled pristine videos by 2x, 4x, and 6x using bilinear interpolation. To simulate quantization, we quantized videos by zeroing the least significant 3$^{\text{rd}}$, 4$^{\text{th}}$, and 5$^{\text{th}}$ bits. For dropped frames, we carefully selected points in the video where a frame drop would be noticeable, then we dropped either 3, 6, or 9 frames at that point. In each case, severity levels were chosen by making them perceptibly separable under normal viewing conditions. Examples of these artifacts, excepting dropped frames, are depicted in Fig. 4.2.

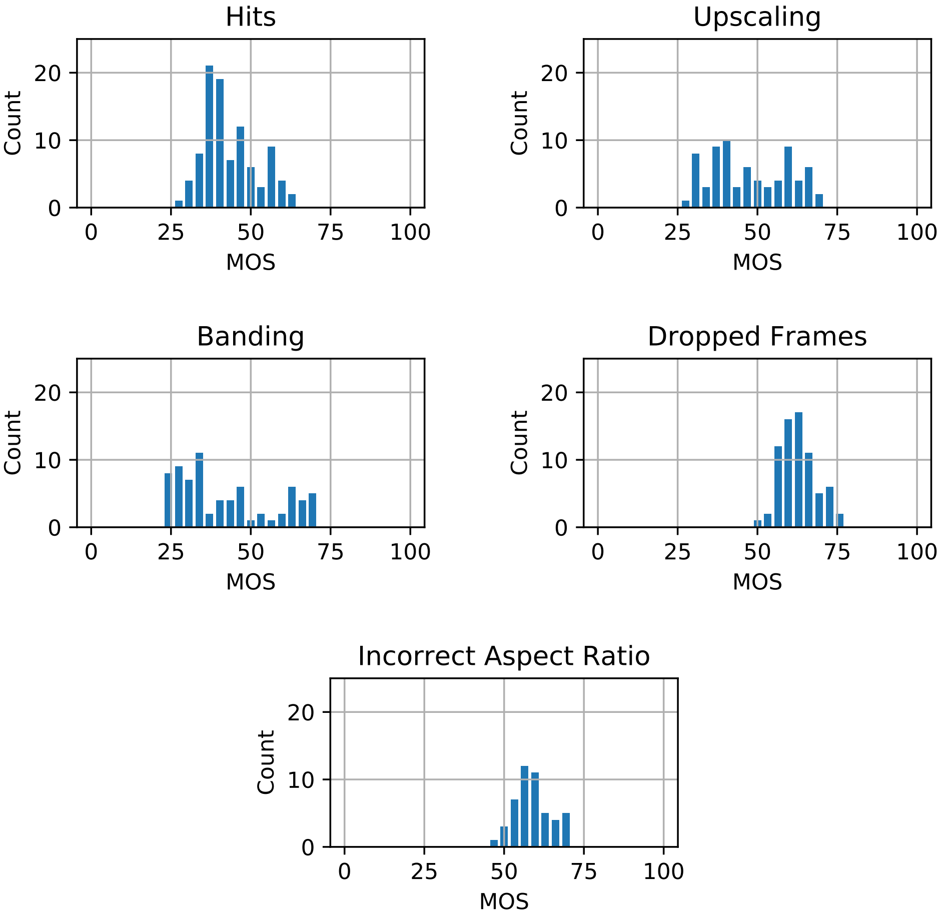


Figure 4.3: Distribution of mean opinion scores per distortion.

4.2 Subjective Study Design

Prior to testing, subjects were provided with brief descriptions regarding the types of artifacts they should expect to see. When informing each subject regarding their task, we stressed that their holistic opinion was most important, and that they should provide ratings that incorporated both seemingly intended and non-intended distortion sources. The exact instructions given to each subject is provided in Appendix C. During each of the three sessions, the subjects were seated with their eyes 2 feet away from the LCD screen.

Since in real viewing scenarios only a single video is watched, the subjects were presented content using a standard Single Stimulus Continuous Quality Evaluation (SSCQE). In other words, a full-screen video would play, and at the end of each video the subject would be asked to provide an overall quality score. This quality score was reported using a continuous sliding bar with qualitative Likert-like labels provided, ranging from "Worst" to "Excellent." In each of the three sessions, all of the contents and distortion types were presented, but the total ranges of distortion severities were spread across the sessions. We randomized the playout order, while also ensuring that identical distortion types and contents did not repeat between contiguous stimulus presentations.

A total of 30 subjects participated in the experiment. Subjects were obtained from the Image Processing class taught at the University of Texas at Austin, graduate students, and a few external participants. Subjects selected from the class were provided the option to participate in the study for course credit in place of a homework assignment. No reward provided to subjects otherwise. Each subject was determined to have either normal or corrected-to-normal vision, as evaluated by their ability to read the 20/20 line of a Snellen chart.

4.3 Subjective Data Analysis

The recorded scores were discretized to integers on [0, 100]. These scores collected from the study are histogrammed in Fig. 4.4. This distribution shows a decent spread of subject scores that span the entire range of quality scores. By averaging scores per video, we produced per-video scores, histogrammed in 4.5. We notice that averaged scores do not span the entire range, due to subject bias and variance. To reduce this bias and variance, we perform z-scoring.

Let s_{ijk} be the opinion score given by subject i, on video j during session $k = \{1, 2, 3\}$. Each score from each session was then converted to a Z-score:

$$z_{ijk} = \frac{s_{ijk} - \mu_{ik}}{\sigma_{ik}} \tag{4.1}$$

where

$$\mu_{ik} = \frac{1}{N_{ik}} \sum_{j=1}^{N_{ik}} s_{ijk} \tag{4.2}$$

and

$$\sigma_{ik} = \sqrt{\frac{1}{N_{ik} - 1} \sum_{j=1}^{N_{ik}} (s_{ijk} - \mu_{ik})^2}, \tag{4.3}$$

and where N_{ik} is the number of test videos seen by subject i in session k. This Z-score computation removes individual subject bias and variation within each session.

We used the rejection procedure specified in the ITU-R BT recommendation 500.13 for discarding scores from unreliable subjects. Z-scores were considered to be normally distributed if their kurtosis fell between the values of 2 and 4. The recommendation is to reject if more than 5 percent of the Z-scores are found to lie outside two standard deviations of the mean. Using this procedure, we found no significant outliers [113] [22].

After the subject rejection procedure, the values of z_{ijk} follow a normal distrution, where 99% of the variance falling in the range on $[-3, 3]$. Linear rescaling was used to remap this range onto $[0, 100]$ using

$$z'_{ij} = \frac{100(z_{ij} + 3)}{6}. \tag{4.4}$$

Finally, the z-scored Mean Opinion Score (MOS) of each video was computed as the mean of the $M = 30$ rescaled Z-scores:

$$\text{MOS}_j = \frac{1}{M} \sum_{i=1}^{M} z'_{ij}. \tag{4.5}$$

A plot of the histogram of the z-scored MOS is shown in Fig. 4.6, indicating a reasonably broad distribution of subjective opinions.

The per-distortion MOS histograms are also provided, in Fig. 4.3. We noticed that upscaling and banding provided wider histograms relative to the hits, incorrect aspect ratio, and dropped frames. We combined the two types of video hits into the same histogram, since the individual histograms for MPEG2 and H.264 hits were similar in appearance.

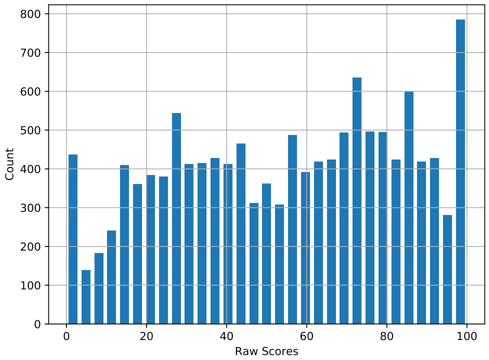


Figure 4.4: Distribution of overall raw scores.

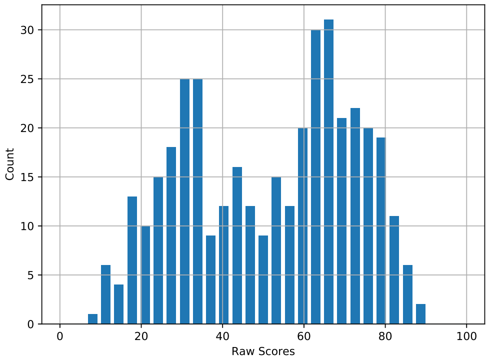


Figure 4.5: Distribution of per-video mean opinion scores.

We measured the degree of agreement between groups of subjects in Fig. 4.8, per distortion type. This was done to inform us regarding potential model prediction limitations. We split the collected subject data into two equally sized groups, computed the MOS of each group, and computed correlation between the two resulting MOS distributions. We repeated this experiment 1000 times, then computed the median SRCC and LCC, which we plotted in Fig. 4.8. Unsurprisingly, we found that the distortions Dropped Frames and

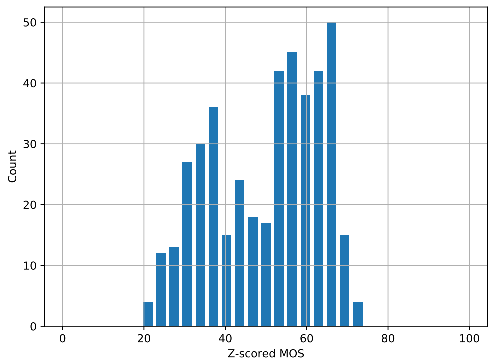


Figure 4.6: Distribution of z-scored per-video mean opinion scores.

Incorrect Aspect Ratio had the lowest median inter-subject group correlations. Figure 4.7 plots the average MOS values per distortion and distortion level along with 95% confidence intervals. Also shown in an overlay is the pristine distribution. As may be seen, significant overlap exists between distortion levels for dropped frames and incorrect aspect ratio, meaning these are the most difficult to predict. Specifically, videos with 3 dropped frames are not statistically separated from the pristine videos, further exhibiting the difficulty that subjects observed when rating this distortion category.

4.4 Discussion and Conclusion

We are releasing the LIVE Video Masters database, which contains a number of distortion types that are highly relevant to modern digital video streaming companies. We believe that this database will prove to be quite useful for developing and evaluating source inspection systems.

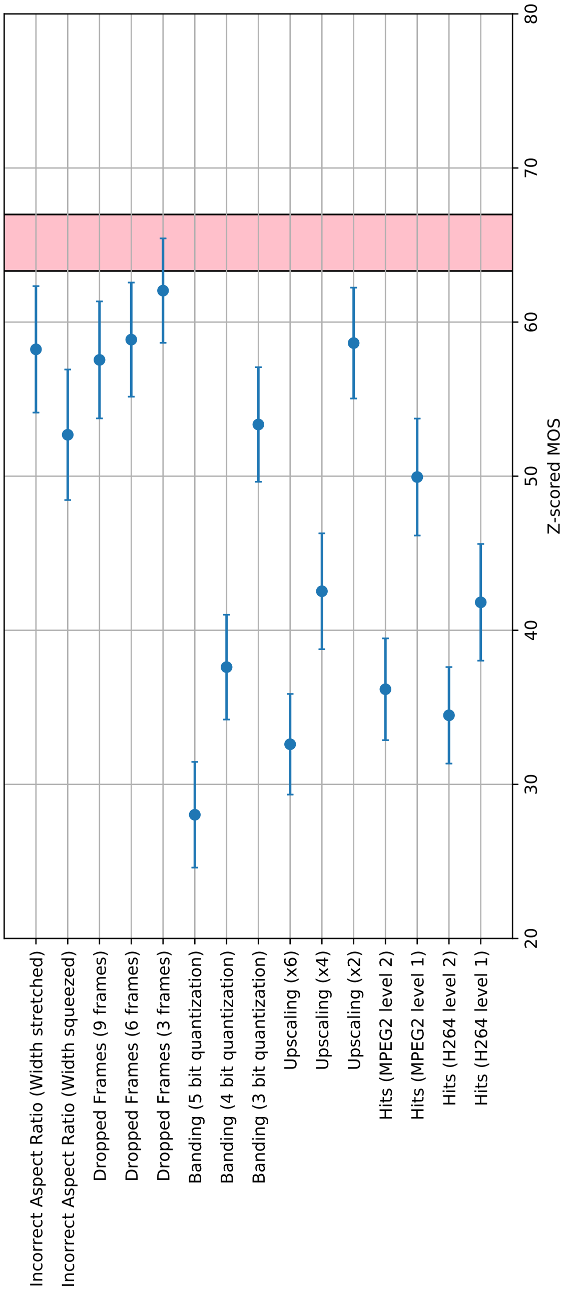


Figure 4.7: Distribution of z-scored mean opinion scores per distortion type.

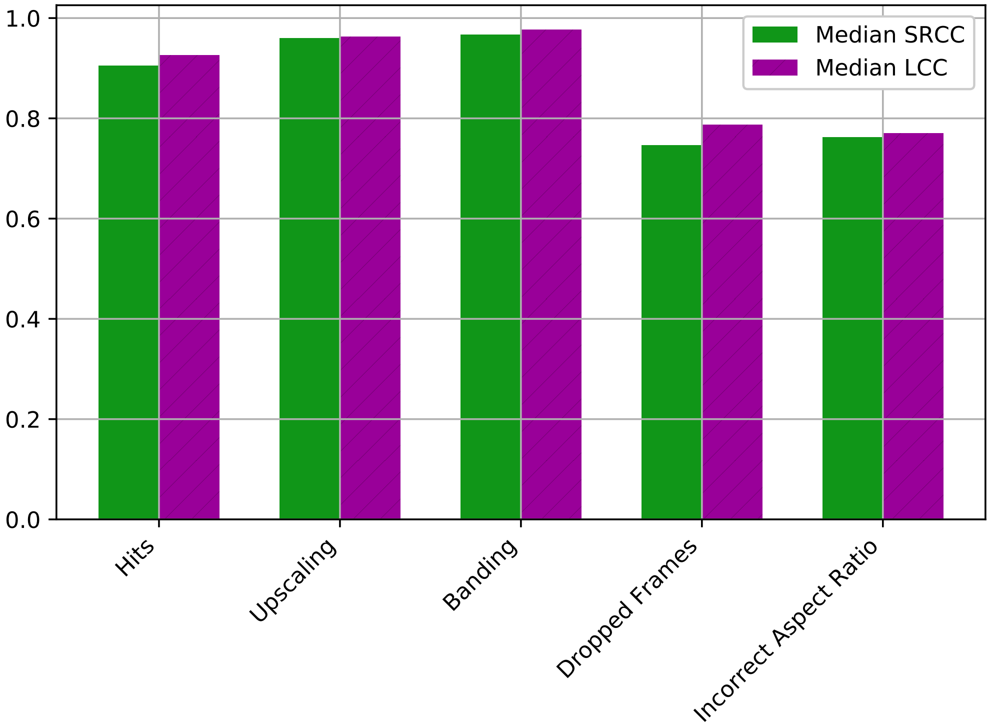


Figure 4.8: Subject to subject median agreement from 1000 random splits.

Chapter 5

VIDSPECT: Video Impairment Detection by SParse Error CapTure

Powerful predictors of picture quality have been developed based on models of human visual perception, which has had substantial time to evolve in response to the statistics of the natural world. Here we extend these principles to the problem of video source inspection, by coupling spatial divisive normalization with a filterbank tuned for artifact detection, and implemented using an augmented sparse functional form. We call this method the Video Impairment Detection by SParse Error CapTure (VIDSPECT). We configure VIDSPECT to create state-of-the-art detectors of 5 kinds of commonly encountered source video artifacts: upscaling, incorrect aspect ratio, dropped frames, video hits, and banding. We validate detection performance using a sizable video dataset, and we evaluate VQA performance using the LIVE Video Masters Database. [1]

[1] This chapter appears in the following papers: T. R. Goodall, I. Katsavounidis, Z. Li, A. Aaron, and A. C. Bovik, "Blind Picture Upscaling Ratio Prediction." IEEE Signal Processing Letters Vol. 23 No. 12, pp. 1801-1805; and T. R. Goodall and A. C. Bovik, "Detecting Source Video Artifacts with Supervised Sparse Filters." accepted by Picture Coding Symposium, 2018. Todd Richard Goodall has designed the models, collected data, and performed full experimental analysis of the works described therein.

5.1 Upscaling-Sensitive model

Before describing VIDSPECT, we first explore predictions that can be made on different aspects of the upscaling problem by developing natural-signal tuned set of basis functions. No-reference quality prediction models such as the Blind Reference-less Image Spatial QUality Evaluator (BRISQUE) [82] and Naturalness Image Quality Evaluator (NIQE) [81] use simple spatial-domain feature extraction strategies that correlate well with human opinions of multiple picture distortion types. Here, we follow this path by describing a new high-performance blind upscaling prediction model that combines a novel pre-filtering technique with the Mean-Subtracted Contrast-Normalized (MSCN) and "paired product" computations developed in BRISQUE.

5.1.1 Proposed Natural Scene-Based Model

By decomposing an input image frame using an orthogonal filter bank and locally normalizing the resulting responses, we show that the local energy terms can be used to predict the upscaling ratio. In fact, a simple linear regressor can be trained on these energy measurements, hence no hyper-parameter tuning is necessary. We compare the proposed model with other no-reference models using real-world data contained in the Netflix collection.

As described in [52], Principal Component Analysis (PCA), when applied to images, can find an orthogonal basis of natural image patches. We observed that these derived basis functions change as natural image patches are upscaled, leading us to explore how these changes can provide a useful

Figure 5.1: Exemplar pristine image selected from the Berkeley image segmentation database [79].

measurement on upscaling artifacts. Although different filter designs may be applied, we opt for a simple approach learned directly from natural images, differing from [134] in that the filters used are not specifically optimized for upscaled images.

We select a corpus of 500 natural luminance images, obtained from the Berkeley image segmentation database [79]. Each image is split into overlapping patches of size 5x5, from which we select 2000 random patches. Each patch is multiplied by a 5x5 Gaussian mask sampled to 2 standard deviations and normalized to unit maximum value to reduce energy at the patch boundaries. Accumulating the weighted patches from each image yields a total of 1 million patches. Given these 5x5 patches, PCA will produce at most 25 orthogonal basis functions, as depicted in Fig. 5.2, most of which exhibit

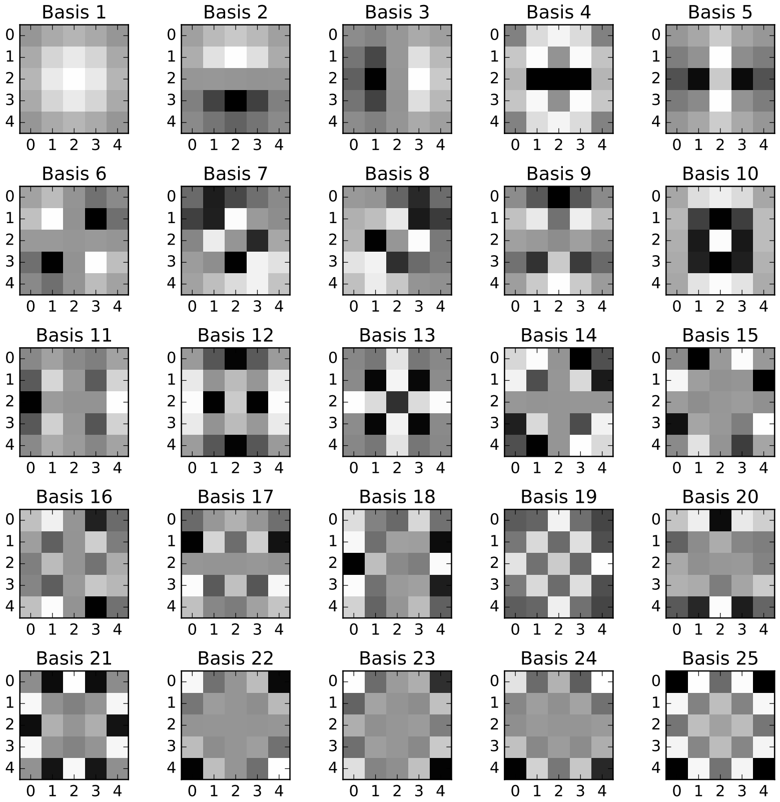


Figure 5.2: Basis functions computed using PCA on 5x5 patches. All patches were obtained on pristine images from the Berkeley image segmentation database [79].

sinusoidal-like properties.

We use these 25 orthogonal basis functions for image pre-filtering. Given an input luminance image, I, a total of 25 response images were produced after filtering with each of these basis functions, yielding $R^{(f)}$ where $f \in \{1, 2, ..., 25\}$. Next, each response image, $R^{(f)}$, undergoes divisive normalization to yield MSCN map $\widehat{R}^{(f)}$ for each f according to

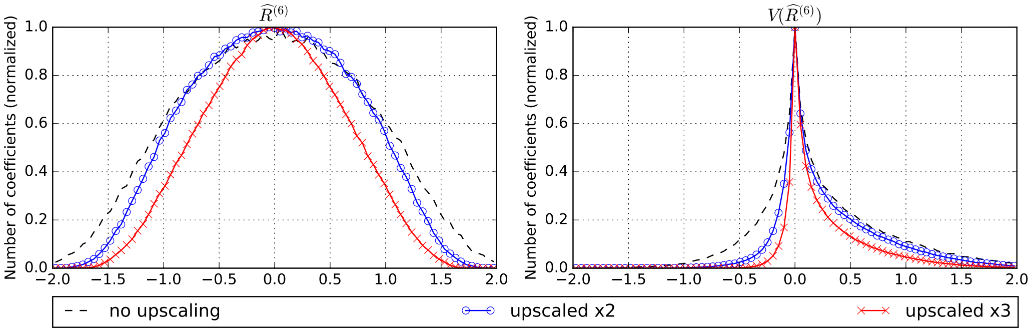


Figure 5.3: Histograms of MSCN and vertical paired product for basis filter 6 for different degrees of upscaling. These coefficients were computed using bicubic upscaling of the image in Fig. 5.1.

$$\widehat{R}^{(f)}(\mathbf{x}) = \frac{R^{(f)}(\mathbf{x}) - \mu(R^{(f)};\mathbf{x})}{\sigma(R^{(f)};\mathbf{x}) + \epsilon}$$

where

$$\mu(R^{(f)};\mathbf{x}) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w_{k,l} R^{(f)}_{k,l}(\mathbf{x})$$

and

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

where $K = L = 5$, $\mathbf{x}$ is the pixel location vector, and $w = \{w_{k,l}|k = -K, \cdots, K, l = -L, \cdots, L\}$ is a 2D circularly-symmetric Gaussian weighting function sampled out to 3 standard deviations and normalized to unit volume. Throughout, we fixed the saturation parameter $\epsilon = 1\text{x}10^{-9}$.

The coefficients $\widehat{R}^{(f)}$ are the MSCN versions of the basis filtered responses, like those obtained in BRISQUE. This MSCN transform is inspired by retinal models of divisive normalization in the human visual system. A total of 25 sample standard deviation features, $\sigma_m^{(f)}$, are computed on the 25 $\widehat{R}^{(f)}$ maps. To obtain measurements of local spatial correlations that may exist after normalization, "paired product" coefficient maps are computed for each $\widehat{R}^{(f)}$ according to

$$\begin{aligned} \mathrm{H}(\widehat{R}^{(f)}; i, j) &= \widehat{R}^{(f)}(i, j)\widehat{R}^{(f)}(i, j+1) \\ \mathrm{V}(\widehat{R}^{(f)}; i, j) &= \widehat{R}^{(f)}(i, j)\widehat{R}^{(f)}(i+1, j) \\ \mathrm{D1}(\widehat{R}^{(f)}; i, j) &= \widehat{R}^{(f)}(i, j)\widehat{R}^{(f)}(i+1, j+1) \\ \mathrm{D2}(\widehat{R}^{(f)}; i, j) &= \widehat{R}^{(f)}(i, j)\widehat{R}^{(f)}(i+1, j-1) \end{aligned}$$

yielding a total of 100 "paired product" maps. The sample standard deviations $pp_H^{(f)}$, $pp_V^{(f)}$, $pp_{D1}^{(f)}$, and $pp_{D2}^{(f)}$ are computed on $\mathrm{H}(\widehat{R}^{(f)})$, $\mathrm{V}(\widehat{R}^{(f)})$, $\mathrm{D1}(\widehat{R}^{(f)})$, and $\mathrm{D2}(\widehat{R}^{(f)})$ respectively. Thus, 25 MSCN features, $\sigma_m^{(f)}$, and 100 local correlation features, $pp_H^{(f)}$, $pp_V^{(f)}$, $pp_{D1}^{(f)}$, and $pp_{D2}^{(f)}$, are computed on each input image, for a total of 125 features.

To observe the behavior of the distributions from which our features are extracted, we plot the histograms of $\widehat{R}^{(6)}$ and $V(\widehat{R}^{(6)})$ in Fig. 5.3, for the case of the test image in Fig. 5.1. When upscaling by factors of 1x, 2x, and 3x, a direct relationship appears between the histogram width and upscaling factor, with higher upscaling resulting in narrower histograms.

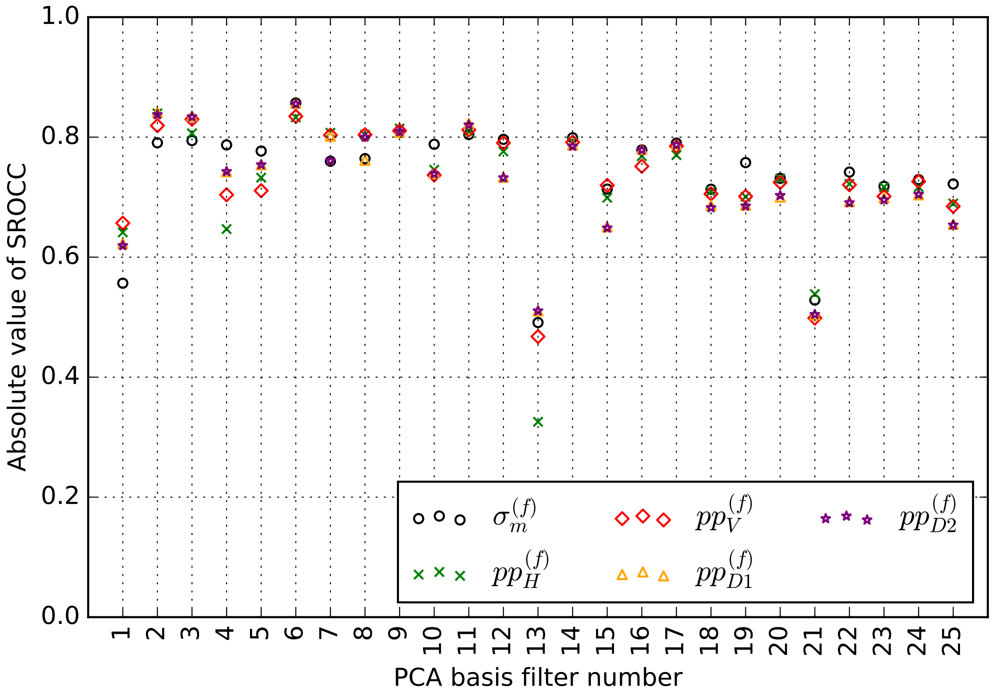


Figure 5.4: Absolute value SROCC between each basis function and the upscaling ratio. Images are upscaled using one of bilinear, bicubic, or Lanczos interpolation.

By measuring correlations between each feature and the upscaling ratio, we can better understand the contribution of each feature to a final prediction. Using the Berkeley dataset, we obtained 1500 images by upscaling the 500 images to upscaling ratios in the continuous range $[1, 3]$ with bilinear, bicubic, and Lanczos upscaling. Next, we observed the correlations between the 125 features and the upscaling ratio. Figure 5.4 shows the absolute Spearman's Rank-Order Correlation Coefficients (SROCC) between features and upscaling ratio.

From Fig. 5.4, the highest correlation occurs using Basis 6, which measures responses to a cross-like shape. A low correlation can be observed against the response to the low-pass Basis 1, since the upscaling artifact perturbs high-frequencies. Interestingly, the 5 features extracted from each basis have similar correlations, except $pp_H^{(13)}$.

5.1.2 General Prediction Performance

To compare performance amongst algorithms on a controlled dataset, the Berkeley segmentation dataset was used again. We upscaled 75% of the images in the dataset to have upscaling ratios in the continuous range $[1.25, 3]$, such that each upscaled image was assigned a unique ratio. The remaining 25% of the images were not upscaled. Each image then received one of three levels of compression: None, 90%, and 80% quality using the *imagemagick* [3] command line utility, which implements JPEG compression. Introducing both upscaling and compression allows for a more realistic test, since delivery of professional content can include both lossless and compressed images. Note that images in this dataset are likely downscaled, minimizing CFA interpolation artifacts.

For the proposed model, predictions of the upscaling ratio were made using both a linear regressor and a Support Vector Regressor (SVR). We compared performance between these regressors to show that a linear combination of the proposed features yields a competitive predictor. Moreover, comparing models using a linear regressor can provide a basis from which to start tuning more complex models. For the alternative models, the suggested predictors

Table 5.1: Median prediction performance across upscaling methods over 1000 train/test trials on "Berkeley" dataset.

Model	Bilinear		Bicubic	
	LCC	MSE	LCC	MSE
Gallagher	0.624	0.404	0.615	0.431
Pfennig and Kirchner (SVR)	0.910	0.079	0.860	0.132
BRISQUE (SVR)	0.956	0.034	0.975	0.021
Feng *et al.* (SVR)	0.973	0.023	0.982	0.015
Proposed (Linear)	0.965	0.030	0.972	0.024
Proposed (SVR)	**0.981**	**0.016**	**0.985**	**0.013**

were used. Note that Gallagher directly estimated upscaling without need for a regressor.

The Berkeley dataset was randomized, then partitioned into two sets, with 75% of the dataset for training and 25% for testing. Models were evaluated on the testing data using the Linear Correlation Coefficient (LCC) and Mean-Squared Error (MSE). This process was repeated 1000 times, each time re-randomizing the dataset order before partitioning. The median results of this testing are reported in Tables 5.1 and 5.2.

As may be seen, the proposed algorithm achieved top prediction results overall, except for Lanczos interpolation. When performance on all combined categories was measured, the prediction performance of all models was found to suffer. This could perhaps be overcome using a more complex machine learning model, as exemplified by the results obtained using the SVR.

Table 5.2: Median prediction performance across upscaling methods over 1000 train/test trials on "Berkeley" dataset. The presence of '*' indicates that all upscaling methods are present in the testing and training sets.

Model	Lanczos		*	
	LCC	MSE	LCC	MSE
Gallagher	0.629	0.476	0.420	0.495
Pfennig and Kirchner (SVR)	0.813	0.188	0.849	0.139
BRISQUE (SVR)	0.977	0.019	0.966	0.029
Feng *et al.* (SVR)	**0.994**	**0.005**	0.968	0.027
Proposed (Linear)	0.981	0.017	0.960	0.035
Proposed (SVR)	0.988	0.012	**0.979**	**0.018**

Table 5.3: Median prediction performance across upscaling methods over 1000 train/test trials on "Movie and TV Show" image dataset.

Model	Bilinear		Bicubic	
	LCC	MSE	LCC	MSE
Gallagher	0.267	0.477	0.029	0.674
Pfennig and Kirchner (SVR)	0.745	0.199	0.460	0.471
BRISQUE (SVR)	0.952	0.041	0.930	0.058
Feng *et al.* (SVR)	0.796	0.161	0.877	0.099
Proposed (Linear)	0.970	0.025	0.961	0.033
Proposed (SVR)	**0.979**	**0.018**	**0.978**	**0.019**

5.1.3 Movie and TV Show Upscaling Prediction Performance

Since the Berkeley dataset was used when training the pre-filters, there might be concern that performance on the Berkeley dataset may be inflated owing to some unseen bias (e.g., in the human selection of content). To address this concern, we collected 801 distinct video frames from the Netflix collection, from movie and TV show sequences that were encoded at resolutions of 480p, 720p, 1080p, and 2160p with extremely light compression. Next, each of these frames was subjected to upscaling as before, using bilinear, bicubic, or Lanczos

Table 5.4: Median prediction performance across upscaling methods over 1000 train/test trials on “Movie and TV Show” image dataset. The presence of ’*’ indicates that all upscaling methods are present in the testing and training sets.

Model	Bilinear		Bicubic	
	LCC	MSE	LCC	MSE
Gallagher	-0.069	0.772	0.416	0.500
Pfennig and Kirchner (SVR)	0.285	0.623	0.430	0.467
BRISQUE (SVR)	0.941	0.050	0.928	0.060
Feng *et al.* (SVR)	0.935	0.055	0.795	0.161
Proposed (Linear)	0.969	0.026	0.951	0.042
Proposed (SVR)	**0.981**	**0.016**	**0.969**	**0.026**

Table 5.5: Median classification accuracy across upscaling methods over 1000 train/test trials on “Berkeley” dataset. The presence of ’*’ indicates that all upscaling methods are present in the testing and training sets.

Model	None	JPEG 90%	JPEG 80%	*
BRISQUE (SVC)	0.872	0.816	0.752	0.768
Feng *et al.* (SVC)	0.968	**0.960**	**0.952**	**0.944**
Proposed (LDA)	**0.984**	0.928	0.856	0.880
Proposed (SVC)	0.976	0.912	0.856	0.872

upscaling. This time, JPEG compression was not applied, since, in practice, source inspection of content is applied only to high quality videos.

Using the same 75%/25% training/test split and 1000 trials, we evaluated the prediction performance of each model, as shown in Tables 5.3 and 5.4. The proposed algorithm delivered outstanding performance on both the 3 datasets containing only a single type of upscaling and on the dataset with multiple types of upscaling. For this particular use case, the energy-based Feng *et al.* features appear to have significant difficulty for both bicubic and

Table 5.6: Median classification accuracy across upscaling methods over 1000 train/test trials on “Movie and TV Show” dataset.

Model	Accuracy
BRISQUE (SVC)	0.776
Feng *et al.* (SVC)	0.672
Proposed (LDA)	**0.935**
Proposed (SVC)	0.915

bilinear upscaling techniques.

5.1.4 General Classification Performance

Determining the interpolation method used is important for both forensic artifact detection and for reporting source issues. At the same time, study of model classification performance can lead to further insights into the actual artifacts. For instance, if classification accuracy of a model is high, then information specific to each upscaling artifact is captured.

As listed in Table 5.5, several models were used to classify an image as having been upscaled using bilinear, bicubic, or Lanczos interpolation. Decisions were made using Linear Discriminant Analysis (LDA) and Support Vector Classifiers (SVCs) for the same reasons that we used linear regression. Again, a total of 1000 randomized 75%/25% train/test splits were used, and the median results reported in Table 5.5. Feng *et al.* largely outperformed the other models.

5.1.5 Movie and TV Show Upscaling Classification Performance

We also measured classification performance on the Netflix video frames as shown in Table 5.6. Here, Feng *et al.* largely underperformed, indicating that measurements on the frequency magnitude are more ambiguous for the given content. When compared to Table 5.5, more mis-classifications occurred for all models. The accuracies across all models are low, implying that classifying the interpolation function is a difficult problem.

5.1.6 Discussion

We proposed a natural scene statistics-based method of predicting the amount of upscaling that has been applied to a picture. We show it to be an accurate and monotonic predictor of upscaling, which can be trained using linear regressors. In addition, the proposed model is a general spatial model that is not necessarily limited to the upscaling artifact.

In fact, we show that this approach can be extended to other artifact types. Instead of estimating the basis functions using PCA, we solve for the basis functions that work best for the specific detection task.

5.2 VIDSPECT System Design

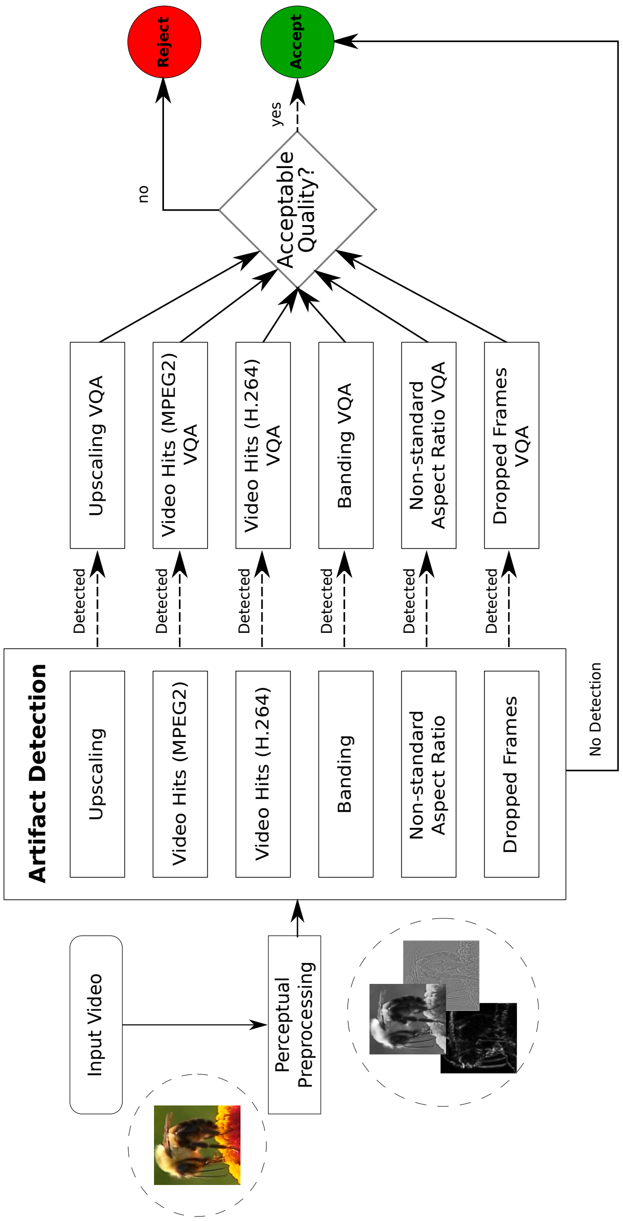


Figure 5.5: VIDSPECT system for detecting and assessing artifact severity.

Inspired by our study of natural scene statistics computed on basis function responses, we introduce an effective, holistic, and compute-efficient framework for detecting and assessing distortion artifacts in video masters, as depicted in Fig. 5.5. In creating this concept, we make the following contributions:

- We develop a first-of-a-kind video master inspection model and algorithm called Video Impairment Detection by SParse Error CapTure (VIDSPECT), which is designed to detect a set of the most common and annoying artifacts that occur in digital source masters, then it assesses the quality of the analyzed masters.
- We designed and built the LIVE Video Masters database described in Chapter 4, which includes opinions gathered on distortions that are relevant to the concerns of streaming digital video companies. Video sources were obtained from both Netflix and the public domain, allowing us to release a sizable subset of this database at [7].
- We devise a sparse dictionary learning model described in Sec. 5.1.1, which can be used to learn a discriminative set of basis functions for video impairment classification and quality prediction.
- We supply a free software release of VIDSPECT at [17].

Basis pursuit denoising (BPDN) is often used when modeling the data distribution of a video signal. BPDN balances a trade-off between data fidelity and sparsity, assuming that a higher dimensional video signal can be represented in a lower dimension with minimal loss of information. In BPDN, data is reconstructed using a weighted sum of basis functions. Limiting the participation of basis functions in the reconstruction is an ℓ_0 norm, for which finding an exact minimizing solution is NP-hard. Lasso [129] was developed as a way of relaxing ℓ_0 minimization by instead using the ℓ_1 norm, which also leads to sparse solutions in many instances.

5.2.1 Pre-Processing Model

VIDSPECT pre-processes each video frame by perceptually relevant spatial bandpass filtering and subsequent local non-linear divisive normalization [106]. Following the image quality literature, we will refer to this step as Mean-Subtracted Contrast Normalization (MSCN) [106, 82, 81]. MSCN is used in several successful image quality assessment (IQA) models as a preprocessing step prior to feature extraction, since it tends to strongly Gaussianize and decorrelate image pixels when applied to high-quality, undistorted images (or video frames). This is the same normalization procedure applied in VIDMAP It greatly reduces the image space to something resembling Gaussian white noise. When distortions are present, this property often becomes lost, hence statistical measurements made on MSCN-processed images are highly sensitive to distortions, *viz.*, are “quality-aware.” The MSCN coefficients of a video frame I are given by

$$\hat{I}(\mathbf{x}) = \frac{I(\mathbf{x}) - \mu(\mathbf{x})}{\sigma(\mathbf{x}) + C} \tag{5.1}$$

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.

The BRISQUE IQA model [82] deploys parametric fits of empirical probability distributions of the MSCN coefficients as the basis for extracting quality-aware picture features. As we explain in the next section, we will instead model local correlations using a set of basis functions as feature extractors. In this way, we will be able to characterize patterns that imply degradations in quality. We consider only the luminance channel when computing MSCN coefficients.

In Sections 5.2.2 and 5.2.3, two classes of modeling are described. In Section 5.2.2, a method of modeling basis functions using patch-based data is described. To overcome limitations associated with this approach, Section 5.2.3 discusses the transition to convolutional-based learning of discriminative basis functions.

5.2.2 Patch-based Sparsity Model

Neurons in human visual cortex decompose image signals into numerous bandpass channels that extract local spatio-temporal information. Classically, overcomplete wavelet transforms are often used to model this process. The dual nature of image statistics and the way the early visual system efficiently encodes information was highlighted by the discovery by Olshausen and Field [92, 120], that filters used to efficiently represent natural images mimic those found in visual cortex. Specifically, they learned a set of image basis functions by using a simple sparsity penalty applied on total activation energy. Sparsity priors have subsequently been successfully implemented in many image processing tasks [23], such as facial recognition [149, 145, 72], pattern modeling, denoising [43], and super-resolution [146].

We are interested in developing similar optimal encoding schemes for specific visual detection tasks. Just as visual cortex can be modeled as an overcomplete filterbank, we consider the possibility of learning embedded patterns in MSCN transformed images using an automatic feature extraction technique that utilizes such a learned filterbank. Towards this purpose, sparse dictionary learning can be used to discover those atoms which underlie pristine and distorted natural images.

Sparsity applied on image patches has shown utility in general recognition and denoising problems. The patch-based sparsity functional, which seeks to minimize the difference between batch of MSCN-transformed patches S and a small number of weighted basis functions ϕ is defined by

$$\underset{X,\phi}{\operatorname{argmin}} \frac{1}{2} \left\| S - \sum_k \phi_k X_k \right\|_2^2 + \lambda \left\| X \right\|_1 \tag{5.2}$$

subject to

$$\left\| \phi_k \right\|_2 = 1,\, X \geq 0.$$

Note that each basis function in ϕ is constrained to share the same dimension as the input MSCN patch S_i. Sparsity is achieved by penalizing the absolute sum of coding matrix X using an Lagrangian multiplier λ. This type of penalization of the coding matrix is known as the ℓ_1 norm.

Since this functional is unsupervised, it does not fully exploit additional information (such as labels). To overcome this, binary labels that indicate artifact presence may be added to the functional. The updated functional with labels is given by

$$\begin{aligned} \underset{X,\phi,p_c}{\operatorname{argmin}} \Bigg[\frac{1}{2} \left\| S - \sum_k \phi_k X_k \right\|_2^2 - \alpha \sum_c y_c \log(p_c) \\ + \lambda \left\| X \right\|_1 \Bigg], \end{aligned} \tag{5.3}$$

subject to

$$\left\| \phi_k \right\|_2 = 1,\, X \geq 0$$

where the first term penalizes the reconstruction error and the second penalizes non-discriminative codes, y is a matrix of binary class labels, and $\|X\|_1$ is the sparsity term. The codes in X are constrained non-negative to enforce an additive relationship among unit-normalized dictionary elements. The predicted class label vector p_i is computed using a linear projection followed by softmax normalization, using

$$p_c = \frac{e^{\sum A(SW_c\phi_c)+b_c}}{\sum_j e^{\sum A(SW_j\phi_j)+b_j}}$$

to project correlations between filters and the input signal onto probability estimates. The diagonal weight matrix W_c is constrained non-negative to enforce correlation between signal and ϕ while reweighing the contributions of each correlation to the overall prediction of class c. Finally, b is the class bias. The term $SW_c\phi$ measures correlation of reweighted dictionary elements $W_c\phi$ with the data S. The function $A(\cdot)$ is the ReLU activation function, the same function commonly used between layers in neural networks. When an element from ϕ correlates with the patch, the output should be a positive value that scales with the degree of correlation. When ϕ has a non-positive correlation, no response is passed through $A(\cdot)$. To isolate filters for particular classes, values in W_c can be set to 0 to disable elements in ϕ for a class.

Discriminative sparse feature learning has been studied previously [147, 77, 55, 102, 73]. Previous methods have attempted to increase the dictionary discrimination power by making the sparse codes more discriminative. Unfortunately, we find that, at least in our application, this approach can couple the sparse code solution directly to the input labels. In other words, discriminative codes are found based on the ground truth labels. Because of this, it turns out that supervised dictionaries learned by making codes more discriminative do no better than unsupervised dictionaries on the artifact detection task. Our approach is different from previous methods, since the sparse code is decoupled from the classification problem. Classification can be thought of as a projection from the input data to the class labels. This approach to incorporating labels into the sparse functional is closest to the work of Mairal *et. al.* [77], but unlike Mairal *et. al.*, there is no direct dependence between the sparse coding problem and the classification problem. As a result, the dictionary learned by minimizing Equation 5.3 will recover the same codes found by minimizing Equation A.1 with the same dictionary. We find that enforcing independence between the code update step and the dictionary update step is necessary for the artifact detection task.

Since equation A.1 is convex in X while ϕ is fixed, we can rewrite equation A.1 to take advantage of the Alternating Direction Method of Multipliers (ADMM) [32] to solve the ℓ_1 minimization. This derivation is further explained in Appendix A.

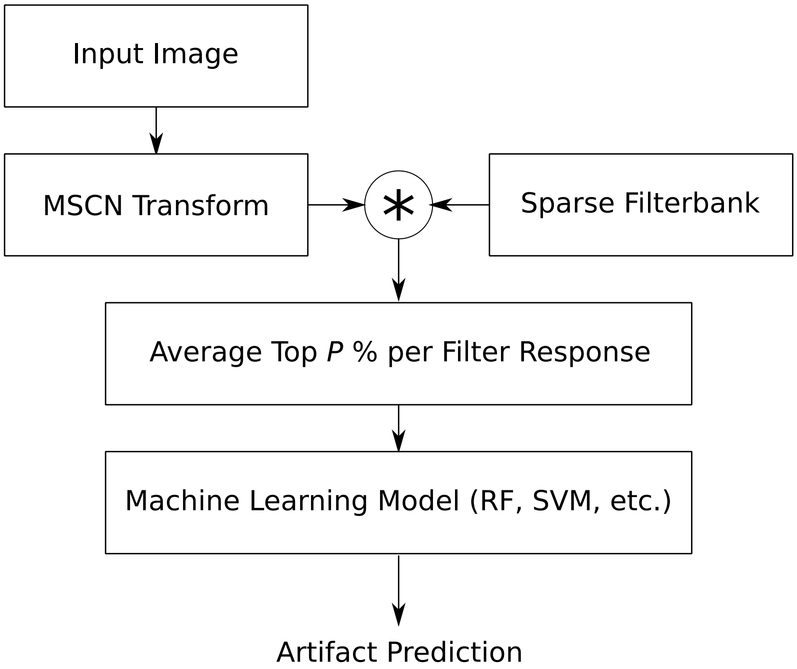

Input Image
MSCN Transform
*
Sparse Filterbank
Average Top *P* % per Filter Response
Machine Learning Model (RF, SVM, etc.)
Artifact Prediction

Figure 5.6: Processing stages of VIDSPECT used to compute an artifact prediction given any input image and trained sparse filterbank.

This system of detecting artifacts in patches can be extended to images and frames of videos, by applying basis functions as convolution templates. In order to expand from patch-based to whole-frame analysis of artifacts, we consider the sparse filterbanks learned by appropriate minimization of equations A.1 and 5.3 to be tuned for detecting artifacts and predicting artifact intensity. We developed a VIDSPECT extraction model which uses this filterbank as a set of feature extractors. The processing stages of VIDSPECT are: computing the MSCN transform on the input frame, using a pre-computed filterbank designed by appropriately minimizing equation 5.3, convolving the MSCN transformed video frame by that filterbank, averaging top responses, then mapping those averages to class labels, as depicted in Fig. 5.6. As we will show, patch-based VIDSPECT performance well across multiple upscaling tasks and for multiple configurations of sparsity and machine learning algorithms.

5.2.2.1 Upscaling Problem Analysis

One approach to studying upscaling would be to model the anti-aliasing filter kernel itself. Such an analysis might involve spatial/frequency analysis of the filter along with analysis of the image spatial/frequency statistics. However, such a global analysis might overlook any peculiarities regarding how real natural picture data is locally perturbed by upscaling. Of particular interest is how to approach perturbations of the natural statistics that occur when the amount of upscaling is arbitrary. To study how natural video frames are perturbed by upscaling, we learned a set of sparse discriminative filters using both upscaled and non-upscaled video frames.

We studied four upscaling interpolation schemes: bilinear, bicubic, Lanczos, and nearest neighbor upscaling, since these are all commonly used to resize, retarget, and otherwise edit video frames. To conduct the analysis, we collected a large dataset of more than 100,000 high quality Netflix video frames. We upscaled these frames using 1 of the 4 chosen interpolation functions. The upscaling ratios were randomly applied in the range 1.25 to 3.0. This range was chosen since we are interested in detecting a range of upscaling factors that includes the practical extreme case where a 720p film is upscaled to 2160p.

To generalize our upscaling analysis, we mixed two philosophies. First, we center cropped from within each video frame, then upscaled to the size of the frame, which ensures that pristine frame data is only perturbed by the upscaling artifact. In the second, alternative approach, we downscaled the frame using a Lanczos-4 filter such that the upscaling factor maintains the same size as the original frame, which ensures that content is held fixed across upscaling factors. We also consider frames downscaled using Lanczos-4 as a part of our non-upscaled frame data. These two scenarios were selected to alleviate concerns regarding scale in film content while also attempting to maintain upscaled film grain noise artifacts.

We then extracted several 25x25 patches from each frame. This size of 25x25 was determined based on the maximum interpolation kernel width, which happens to be Lanczos kernel with upscaling factor of 3. We split this collection of patches into training and testing halves, by dividing based on frame content. This yielded 60,000 patches for testing and 100,000 patches for training. A total of five classes are balanced in both patch datasets - "No Upscaling," "Bilinear," "Bicubic," "Lanczos," and "Nearest Neighbor."

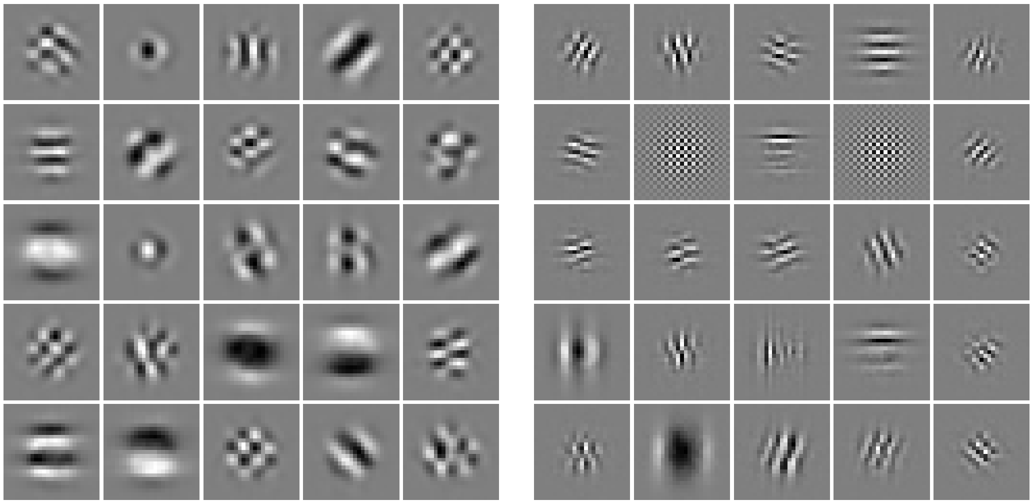
To explore the temporal aspect of videos, a separate dataset of frame-differences was created using the same methodology. Two consecutive video frames are differenced then processed using MSCN. Patches are extracted, taking special care not to extract patches near frame edges. This allows us to produce a second patch-based dataset of equal size to the single-frame dataset, allowing us to directly compare the difference in prediction performance between single-frame and frame-difference predictors.

Towards understanding how well the learned system can characterize upscaling artifacts, we devised five tasks, the first of which involved use of just the sparse code with the last four tasks involving only the use of VIDSPECT features. The first task was to predict the interpolation kernel from the image data using a sparse code. The second task was to discriminate between upscaled and non-upscaled frames. The third involved identifying the interpolation scheme used from among non-upscaled (pristine), bilinear, bicubic, Lanczos, and nearest neighbor upscaling. The fourth was to predict native resolution of both pristine and upscaled images. Lastly, the fifth was to study how effectively the sparse basis functions can be adapted to predict human opinion scores of upscaling.

5.2.2.1.1 Detection Shallow machine learning algorithms yield models that produce a final predictor following a process of feature extraction. For example, BRISQUE [82] and Feng [44] use a support vector machine (SVM) to produce a final mapping. Since it is not clear which model should be used for mapping algorithm features to either class labels or continuous labels, we evaluated two non-linear models, an SVM and a Random Forest (RF), and compared them against linear models including Linear Discriminant Analysis (LDA) and linear regression. Optimal parameters for each model are chosen by maximizing the median performance of 5-fold cross validation using just

Table 5.7: Upscaling detection performance measured on the test set of 20,000 patches, where upscaling type includes "Not Upscaled," "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," and "Nearest Neighbor Upscaling." The reported measure is the F1 score.

Algorithm	Bilinear	Bicubic	Lanczos	N. Neighbor	All
VIDSPECT ($\alpha = 1.0$)	**0.9950**	**0.9949**	**0.9952**	**0.9923**	**0.9909**
VIDSPECT ($\alpha = 0.0$)	0.9715	0.9843	0.9931	0.9810	0.9689
VIDSPECT-D ($\alpha = 10.0$)	0.9884	0.9909	0.9934	0.9914	0.9875
VIDSPECT-D ($\alpha = 0.0$)	0.9860	0.9894	0.9926	0.9884	0.9847
Goodall *et al.* [50]	0.9872	0.9885	0.9941	**0.9977**	**0.9893**
BRISQUE [82]	0.9331	0.8988	0.8847	0.8847	0.8730
Vázquez-Padín *et al.* [133]	0.9736	0.9706	0.9683	0.9929	0.9729
Feng *et al.* [44]	0.7207	0.8303	0.9155	0.8150	0.7206



(a) Evidence for upscaling

(b) Evidence against upscaling

Figure 5.7: Filter developed for positive and negative evidence categories.

the training subset. To assess the binary classification performance, we measured the F1 score, which is the harmonic mean of precision and recall, and the Matthews Correlation Coefficient (MCC), which is a balanced measure

Table 5.8: Upscaling detection performance measured on the test set of 20,000 patches, where upscaling type includes "Not Upscaled," "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," and "Nearest Neighbor Upscaling." The reported measure is Matthew's Correlation Coefficient (MCC).

Algorithm	Bilinear	Bicubic	Lanczos	N. Neighbor	All
VIDSPECT ($\alpha = 1.0$)	0.9899	0.9897	0.9904	0.9845	0.9818
VIDSPECT ($\alpha = 0.0$)	0.9427	0.9686	0.9862	0.9620	0.9379
VIDSPECT-D ($\alpha = 10.0$)	0.9767	0.9819	0.9868	0.9827	0.9750
VIDSPECT-D ($\alpha = 0.0$)	0.9719	0.9788	0.9853	0.9768	0.9693
Goodall *et al.* [50]	0.9744	0.9769	0.9882	0.9953	0.9786
BRISQUE [82]	0.8650	0.7949	0.7657	0.7639	0.7417
Vázquez-Padín *et al.* [133]	0.9469	0.9409	0.9361	0.9858	0.9454
Feng *et al.* [44]	0.7207	0.8303	0.9155	0.8150	0.7206

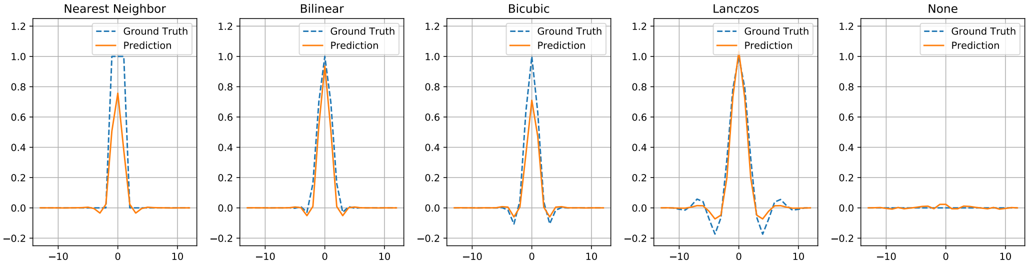


Figure 5.8: Upscaling kernel prediction for individual samples randomly selected from test set. The basis learned using $\lambda = 0.1$ and $\alpha = 1000.0$ was used.

related to the chi-square statistic. To assess multi-class classification performance, we measured the F1-macro score, which is the harmonic mean of the averaged precision and the average recall across classes. To assess regression performance, we measured Spearman's Rank-Ordered correlation Coefficient (SRCC) for monotonicity and Mean-Squared Error (MSE) for point-wise accuracy.

Tables 5.7 and 5.8 list the performance results of VIDSPECT and VIDSPECT-D on the upscaling detection task. VIDSPECT-D refers to VIDSPECT applied on frame differences. We tested both detection performance when only one interpolation method was present in the upscaled class, and also when all interpolation methods were present in the upscaled class. Among all machine-learning methods, the SVM classifier provided the best performance, although the Random forest classifier achieved nearly identical performance. In general, LDA yielded only slightly reduced performance. From these results, we conclude that VIDSPECT yielded the best upscaling detector.

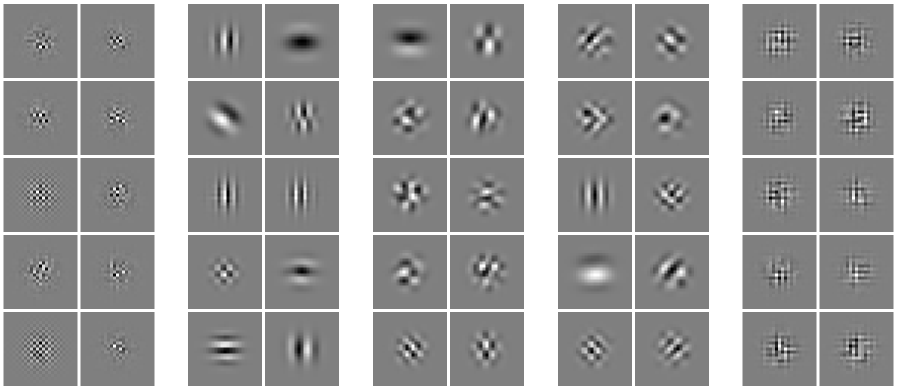
Evaluation can be done per-patch for whole frames using the model in Fig. 5.6. Much more evidence of upscaling can be found when scanning through a frame, since a frame is composed of many patches. A minimum of one patch in the frame needs to exhibit strong evidence of upscaling to classify the entire frame as upscaled.

We evaluated VIDSPECT by choosing parameters for each model that reasonably spanned the parameter space for α and λ. We considered $\alpha \in \{0.0, 1.0, 10.0\}$ and $\lambda \in \{0.1, 0.5, 1.0\}$. The best learned positive and negative evidence detection filters are provided in Fig. 5.7. From the positive evidence for upscaling in Fig. 5.12a, we can see checkerboard patterns and directional sinusoid-like patterns. From the negative evidence in Fig. 5.12b, we can see much higher frequencies.

Table 5.5 lists the performance results of VIDSPECT on the upscaling detection task. We tested both detection performance when only one interpolation method was present in the upscaled class, and also when all interpolation methods were present in the upscaled class. Among all machine-learning methods, the SVM classifier provided the best performance, although the Random forest classifier achieved nearly identical performance. In general, LDA yielded only slightly reduced performance. From these results, we conclude that VIDSPECT yielded the best upscaling detector, and that using different sparsity methods in it provided only small differences in performance.

5.2.2.1.2 Method Discrimination The filters for the discrimination problem are provided in Fig. 5.9. These basis functions all exhibit directional high frequency patterns, which intuitively follows since upscaling artifacts mostly affect high-frequencies. Evidence against upscaling exhibits the highest frequencies, which intuitively follows from how frequency spectra falloff more rapidly for each of the different interpolation methods. If high frequencies are well-represented in a patch, then it likely not upscaled. The progression in interpolation order can be clearly seen across Bilinear, Bicubic, and Lanczos basis classes. In other words, bilinear basis functions exhibit patterns with 1-2 cycles, bicubic basis functions exhibit 2-3 cycles, and Lanczos exhibits at least two cycles, all at different orientations. Nearest neighbor is visually distinct, picking up on different numbers of high-frequency cardinal edges.

Table 5.9 compares performance across methods for the upscaling type discrimination task. Again, the task was to discriminate amongst the "Not Upscaled," "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," and



(a) Not upscaled (b) Bilinear (c) Bicubic (d) Lanczos (e) Nearest Neighbor

Figure 5.9: Dictionaries learned for each evidence category, when assigning 10 filters to each. Filter size is held constant at 25x25.

"Nearest Neighbor Upscaling" classes. VIDSPECT performed well when compared against other models.

5.2.2.1.3 Kernel Estimation Using sparse filters, we can predict the original kernel from the sparse coding matrix, using a small fully connected neural network with a single hidden layer of 25 units to map from the sparse code to the kernel function. We used

$$\|G - W_2 f(W_1 X + b_1) + b_2\|_2^2 \tag{5.4}$$

where G is the 25xN kernel matrix where N is number of samples and f is the logistic function given by

Table 5.9: Upscaling type discrimination performance on the test set of 20,000 video frame patches when classifying upscaling type among "Not Upscaled," "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," and "Nearest Neighbor Upscaling." Reported values are F1-macro scores, since the classes are well-balanced.

Algorithm	F1-Macro
VIDSPECT ($\alpha = 1.0$)	0.9225
VIDSPECT ($\alpha = 0.0$)	0.9206
VIDSPECT-D ($\alpha = 10.0$)	0.8965
VIDSPECT-D ($\alpha = 0.0$)	0.8838
Goodall *et al.* [50]	0.8753
BRISQUE [82]	0.4921
Feng *et al.* [44]	0.7519

Table 5.10: MSE between predicted kernel and true kernel for different α and λ evaluated on the test set of patches. The number of basis functions used is 100.

	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1.0$
Single-frame $\alpha = 0.0$	0.0129	0.0119	0.0134
Single-frame $\alpha = 1000.0$	0.0086	0.0094	0.0120
Frame-diff $\alpha = 0.0$	0.0165	0.0162	0.0169
Frame-diff $\alpha = 1000.0$	0.0134	0.0130	0.0143

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5.5}$$

. We used gradient descent to find the weights for W_2, W_1, b_1, and b_2 in Equation (5.4). Although we use this perceptron for mapping from code responses to upscaling kernel, we believe any machine learning technique can be used.

Estimating the upscaling kernel allows for direct identification of the impulse function used for interpolating the image data. OpenCV was used to compute both the interpolation kernels, as it was also used for generating

Table 5.11: Native resolution prediction on patches that were not upscaled, and upscaled using “Bilinear Upscaling,” “Bicubic Upscaling,” “Lanczos Upscaling,” and “Nearest Neighbor Upscaling” using upscaling ratios chosen from the range 1.25x to 3x. The metric being measured is SRCC.

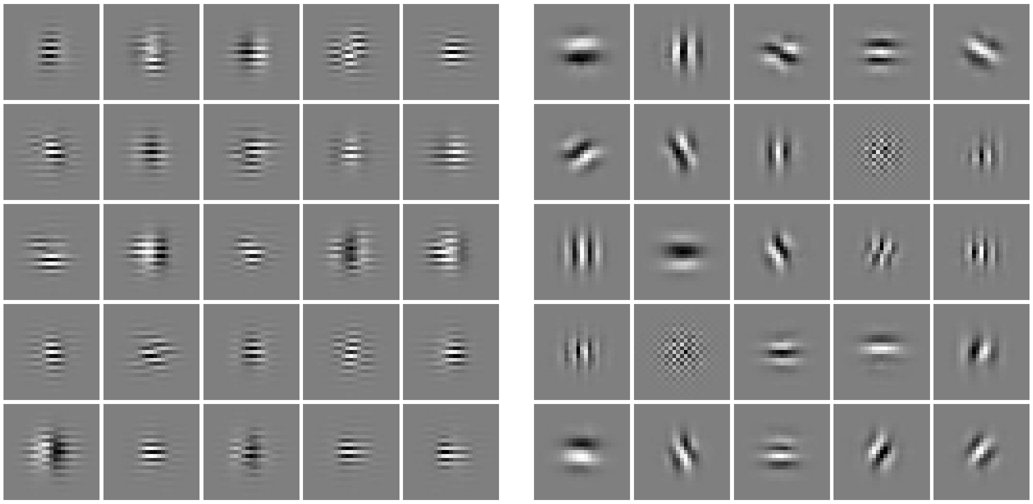
Algorithm	Bilinear	Bicubic	Lanczos	N. Neighbor	All
VIDSPECT ($\alpha = 0.1$)	**0.9684**	**0.9669**	**0.9665**	**0.9357**	**0.9445**
VIDSPECT ($\alpha = 0.0$)	**0.9678**	**0.9668**	**0.9667**	**0.9353**	0.9353
VIDSPECT-D ($\alpha = 10.0$)	0.9469	0.9578	0.9567	0.9260	0.9250
VIDSPECT-D ($\alpha = 0.0$)	0.9383	0.9458	0.9581	0.9194	0.9179
Goodall *et al.* [50]	0.9076	0.9021	0.9092	**0.9325**	0.9055
BRISQUE [82]	0.8412	0.8235	0.8343	0.7933	0.7663
Vázquez-Padín *et al.* [133]	0.8713	0.8541	0.8460	0.8638	0.8591
Feng *et al.* [44]	0.8017	0.8702	0.9037	0.8647	0.8048
Pfennig and Kirchner [96]	0.6734	0.7142	0.7486	0.5546	0.6184

the upscaled images. As can be seen in Fig. 5.8, the upscaling kernel can be estimated with great accuracy. In Table 5.10, we see improved kernel prediction performance when using labels, by comparing $\alpha = 0$ and $\alpha = 1000$ cases.

5.2.2.1.4 Native Resolution Prediction Tables 5.11 and 5.12 list native resolution prediction performances across algorithms. VIDSPECT delivered much better predictions of native resolution than the other models, with each sparsity configuration being close in performance. When evaluating different machine learning algorithms, Random Forest Regression performed best for this task.

Table 5.12: Native resolution prediction on patches that were not upscaled, and upscaled using "Bilinear Upscaling," "Bicubic Upscaling," "Lanczos Upscaling," and "Nearest Neighbor Upscaling" using upscaling ratios chosen from the range 1.25x to 3x. The metric being measured in MSE.

Algorithm	Bilinear	Bicubic	Lanczos	N. Neighbor	All
VIDSPECT ($\alpha = 0.1$)	**20.59**	**14.30**	**14.22**	20.83	**26.28**
VIDSPECT ($\alpha = 0.0$)	21.73	14.99	14.63	21.36	27.20
VIDSPECT-D ($\alpha = 10.0$)	35.14	24.82	21.26	29.97	37.85
VIDSPECT-D ($\alpha = 0.0$)	39.92	27.46	21.18	46.57	46.13
Goodall *et al.* [50]	63.83	77.21	63.87	**15.58**	70.70
BRISQUE [82]	168.44	202.19	189.64	250.26	282.86
Vázquez-Padín *et al.* [133]	201.81	234.94	250.88	227.76	227.66
Feng *et al.* [44]	250.54	141.57	76.88	147.70	238.02
Pfennig and Kirchner [96]	466.51	445.45	431.40	578.59	505.33



(a) Positive evidence

(b) Negative evidence

Figure 5.10: Sparse filters learned for interlacing.

5.2.2.2 Interlace Detection

Combing artifacts can be much more visually obvious than upscaling effects when viewed on progressive displays. Combing manifests as annoying jigsaw patterns, typically along edges, arising from interleaved rows shared between original frames offset slightly in time. The artifact often becomes increasingly obvious on scenes containing rapid motion.

We collected a training/validation dataset of 581 interlaced combed sequences, where sequences consist of 3 frames. A combed sequence is one where the middle frame exhibits visible combing (the others may also). To balance these positive samples, an equally sized set of 581 non-interlaced video sequences was gathered as negative examples. A negative sequence is one where no frames exhibit visible combing. We collected a separate content-distinct test dataset containing 75 interlaced three-frame sequences and 75 undistorted three-frame sequences.

The functional defined in Equation (5.10) was used to predict the combing artifact. Thus, for 338 basis functions, W is a 1x338 matrix and b is a scalar. We evaluated parameters for $\alpha \in \{0.0, 0.1, 1.0, 10.0\}$ to understand how classification capability is impacted. Sparsity parameters of $\lambda \in \{0.1, 0.5, 1.0\}$ were also evaluated.

Figure 5.10 depicts a set of basis functions that are correlated and a set of basis functions that are uncorrelated with the interlacing artifact label, based on values learned in W. In Fig. 5.10a, the zigzag pattern of combing is apparent. In Fig. 5.10b, low-frequencies and vertical edges dominate, which indeed do not indicate presence of combing.

Table 5.13 lists the F1 and MCC performances for the selected algo-

Table 5.13: Combing detection results computed on the test set of 150 video sequences.

Algorithm	F1	MCC
VIDSPECT ($\alpha = 1.0$)	**0.9730**	**0.9470**
VIDSPECT ($\alpha = 0.0$)	**0.9730**	**0.9470**
VIDSPECT-D ($\alpha = 10.0$)	0.9306	0.8695
VIDSPECT-D ($\alpha = 0.0$)	0.9241	0.8552
BRISQUE [82]	0.8718	0.7357
FFmpeg	0.9167	0.8427
Baylon [28]	0.8811	0.7761

rithms. VIDSPECT performance is reported using a Random Forest classifier, and we noticed little difference when testing SVM and LDA classification performance. We observe that the each sparse configuration of the VIDSPECT detector yielded much higher accuracy than the other compared detectors, while requiring only a single frame. Of course, the single-frame sparse detector involves significantly higher computational load to achieve the increase in performance. Optimized using 5-fold cross validation, the optimal threshold parameter for FFmpeg's detector is $T_1 = 1.0551$, and the optimal parameters for Baylon's detector are $T_0 = 75$, $T_1 = 1.113$, and $Z = 10$.

5.2.3 Convolutional Sparsity Model

We find that a dictionary learned using convolution learns more specific structures of distortions. We start with the BPDN sparsity inducing ℓ_1 functional [129], in convolutional form

$$\underset{x,\phi}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{I} - \sum_k \hat{\phi_k} * x_k \right\|_2^2 + \lambda \left\| x \right\|_1 \tag{5.6}$$

subject to

$$\left\| \hat{\phi_k} \right\|_2 = 1,\, x \geq 0.$$

where $\hat{I}$ is the pre-processed image, $\hat{\phi}$ are normalized dictionary elements, x is the coding tensor, and x_k is the spatial coding map for dictionary element k. Sparsity is achieved by penalizing the coding tensor x. The idea behind this minimization problem is that the optimized basis functions will reflect sparse distortions from the otherwise very regular structure of natural images.

Given that we wish to detect a finite set of domain-specific distortions, we instead learn a set of sparse basis functions by learning them on labeled sets of distorted videos. Thus, as in the patch-based functional, we impose binary labels that indicate the presence or absence of a distortion, to optimize a discriminative set of basis functions for each distortion. Thus, modify (5.6) with labels as

$$\underset{x,\phi,p}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{I} - \sum_k \hat{\phi_k} * x_k \right\|_2^2 - \alpha MN \sum_c y_c \log(p_c) + \lambda \left\| x \right\|_1 , \tag{5.7}$$

subject to

$$\left\| \hat{\phi_k} \right\|_2 = 1,\, x \geq 0$$

where α is the classification weight, y_c are the binary class labels that indicate the presence or absence of a given distortion, and p is the predicted distortion probability. All values in the coding map must be non-negative to model feature correlations with the distortion artifacts.

The normalized filterbank $\hat{\phi}$ is made more discriminative by applying a logistic non-linearity

$$p = \frac{1}{1 + e^{-\sum_k \max(A(S * \hat{\phi_k})) * W_k - b}}, \tag{5.8}$$

where W_k is a scalar which remaps the filter activations to weighted evidence, b is a scalar classification bias, $A(\cdot)$ is the ReLU activation function, and $\max(\cdot)$ outputs the maximum response over spatial indices. W is forced to be strictly positive to reduce dependencies among the filters. For the artifact detection problem, the two-class problem weighs evidence for, not against, the detection of an artifact. Given this, filters learn to promote artifact detection, while the bias balances the scales of evidence for detection.

To minimize (5.6) and (5.10), one could leverage the proximal methods recently developed to compute efficient convolutional sparsity [143, 27]. Since the data variable S is assumed to be much larger than available memory, the learning must necessarily be done in batches, as in [138]. We opt for a simple approach that can leverage any popular convolutional network framework. We optimize x using a convolutional autoencoder, learned using gradient descent, instead of optimizing for the sparse coding tensor x directly. We reformulate the functionals to take the form of an autoencoder, by letting $x = A(S * \phi^T)$.

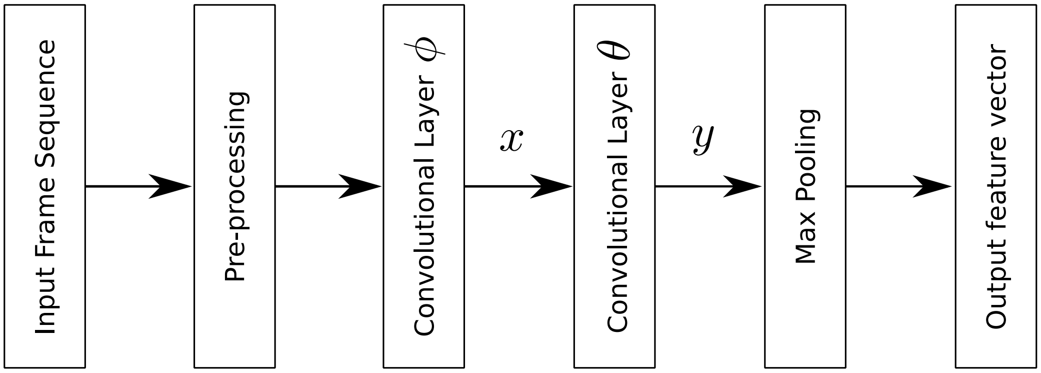


Figure 5.11: Convolutional sparsity feature extraction flowchart.

Note that ϕ^T is the non-normalized transposed version of ϕ.

This approach allows multiple layers of sparse filters to be used. We also define one additional layer, extending this approach using $y = A(x * \theta^T)$, where θ is a new filterbank, and y are the response outputs of this new layer. With this layer, Eq. (5.8) becomes

$$p = \frac{1}{1 + e^{-\sum_k \max(y_k) * W_k - b}} \tag{5.9}$$

to enforce that response outputs contribute to the classification problem. We find that this additional layer allows for a more compact representation of artifacts, significantly improving artifact detection and subjective quality prediction.

The revised supervised sparse functional that incorporates the autoencoder optimization is provided as

$$\operatorname*{argmin}_{x,\phi,p_{ck}} \frac{1}{2} \left\| \hat{I} - y * \hat{\theta} * \hat{\phi} \right\|_2^2 - \alpha MN \sum_c y_c \log(p) + \lambda \left[\|x\|_1 + \|y\|_1 \right]. \tag{5.10}$$

These sparse filterbanks ϕ and θ can be used for feature extraction according to the flow depicted in Fig. 5.11. An input frame or frame difference is pre-processed using Eq. (5.1), convolved with filters in ϕ, then passed through a ReLU activation layer to generate the response map tensor x. The responses in x are convolved with θ and passed through a ReLU activation to compute the response map tensor y. The responses are pooled by choosing the maximum of each filter response to develop a final feature vector.

5.3 Distorted Video Datasets

Toward training and validating our artifact detection model, we developed a collection of independent distortion-specific video datasets. On consultation with colleagues in the streaming video industry, we studied the following important artifacts: upscaling, banding, video hits (MPEG2) and (H.264), dropped frames, and incorrect aspect ratio. For each dataset, we collected a number of pristine videos from the Netflix collection. To isolate scenes, we segmented the pristine videos along scene boundaries using [93], which compares luminance distributions between frames.

For the upscaled video dataset, we upscaled pristine video sources by randomly choosing one from among the "Bilinear," "Bicubic," "Lanczos," and "Nearest neighbor" interpolation methods, and choosing a uniform random number in the range [1.25, 6.0] as the upscaling factor. To complement this first collection, we created another group of upscaled videos, by first downscaling an input video by a uniform random factor chosen in the range [1.25, 6.0] using

Lanczos interpolation, then upscaling it back to native resolution using one of the randomly chosen interpolation types. We did this to simulate realistic occurrences whereby videos may have been downscaled, then later upscaled to fit evolving display technologies. After upscaling a video, a number of 256x256 patches were extracted from regions exceeding a minimum variance threshold. Without applying any threshold on minimum variance, pristine patches were selected from random locations within the pristine video set, and additional pristine patches were selected from the downscaled versions of the pristine video set. In this way, we collected a total of 129,561 samples.

The quantized video dataset was produced from pristine video sources by first selecting a quantization factor $q \in \{8, 16, 32\}$, then for a given randomly selected patch P, applying

$$Q = q\left\lfloor \frac{P}{q} \right\rfloor \tag{5.11}$$

to yield a quantized patch Q. We selected the same 256x256 patch size as in the upscaling dataset, for both quantized and non-quantized patches. For quantized patches, a small threshold was used to reject low contrast patches. A total of 64,925 quantized samples were produced in this manner.

The dataset for videos with dropped frames was created from pristine videos by dropping N consecutive frames, where $N \in \{3, 6, 9\}$. To collect positive examples of dropped frames, we captured the 2 frames preceding and the 2 frames following the dropped frames, then concatenated them to form a 4 frame sequence. Next, we selected random spatial locations to extract four

256x256 patches centered about each location. A minimum value of 5 of the widely-used temporal activity index TI [141] was required across all frames to ensure that enough motion was present such that a frame drop would be visible. We did not threshold for negative samples. A total of 51,562 samples were generated in this way.

Two video hits datasets were created by corrupting MPEG2 and H.264 bitstreams. To corrupt videos from the pristine corpus, we used FFmpeg's 'bsf' noise flag, which sets the corruption ratio, which is defined as the proportion of correct bits relative to distorted bits. The lower this ratio, the more corruptions that appear. We set the ratio to 1:2000000 for H.264 hits and 1:100000 for MPEG2 hits. These values were selected such that both small and large-scale artifacts would appear in the corrupted videos. We then extracted 256x256 patches from corrupted videos, rejecting patches that did not exceed a small threshold on the absolute difference between the patch and its pristine version. 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 62,417 H.264 hit samples, and 59,941 MPEG2 hit samples were generated.

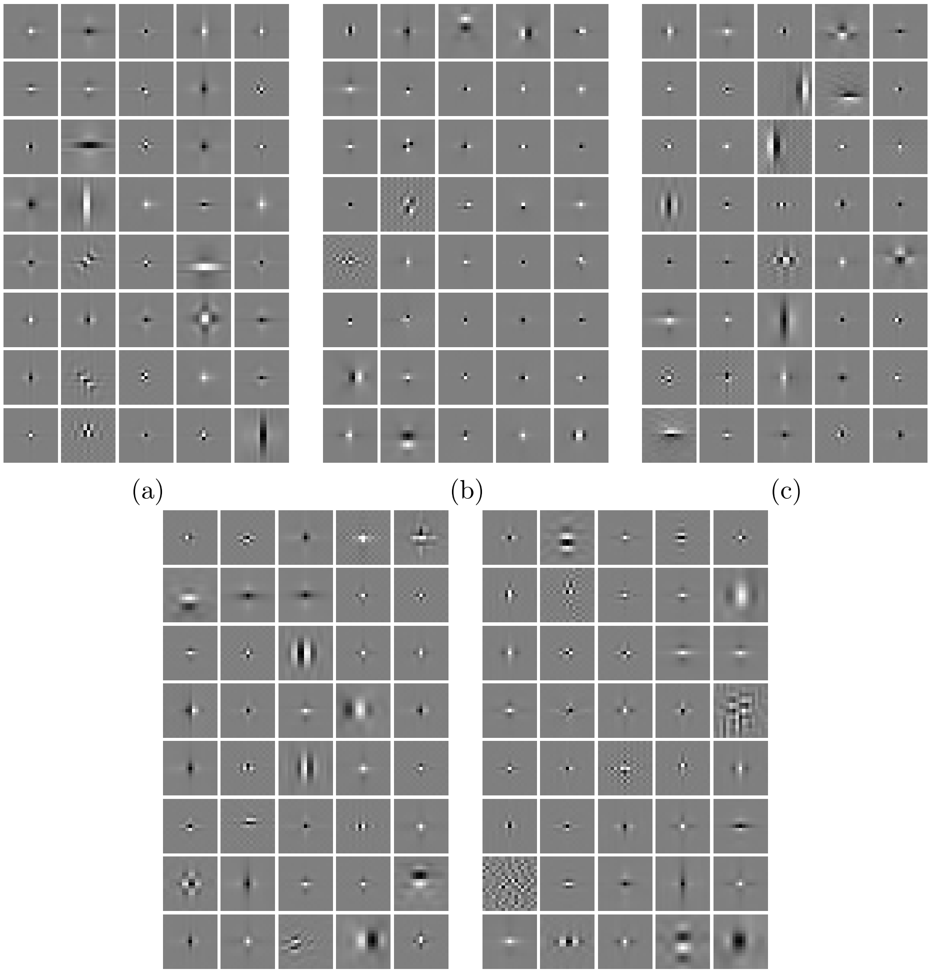
The incorrect aspect ratio dataset was generated by either squeezing or stretching the width of an input pristine video by a factor uniformly randomly chosen in the range [1.15, 2.0]. This range of aspect ratio manipulation was selected to cause visible distortions spanning barely noticeable to very noticeable. All pristine videos was assumed to be of correct aspect ratio. Patches of size 256x256 were selected, each centered on a random spatial location in a random frame. For patches selected from frames with incorrect aspect ratio, a small threshold was used to reject regions of low contrast. A total of 38,670 patches of size 256x256 were collected in this way.

To facilitate detection of frame-differenced input data, we also extracted spatially corresponding patches in the previous frame corresponding to the patches extracted for incorrect aspect ratio, video hits, banding, and upscaling datasets.

5.4 Model Analysis

We trained the VIDSPECT model on each dataset, using values of $\alpha = 10.0$ and $\lambda = 1.0$. For training, we used a batch size of 50 and a learning rate of 1e-4. By measuring overall loss as shown in Fig. 5.14, we found that VIDSPECT converged after 40,000 batch iterations.

Figure 5.12 depicts basis function sets ϕ, where each set is trained on one of the five artifact types. The ϕ that were tuned for detecting upscaling are shown in Fig. 5.12a. These exhibit a mixture of sinusoidal patterns at various scales, which mimic the appearance of upscaling interpolation kernels. Fig. 5.12b shows the banding basis functions, which contain highly localized center-surround and edge patterns. For the hits basis functions in Figs. 5.12c and 5.12d, more complex patterns appear, including patterns resembling corner and edge detectors. Lastly, the aspect ratio basis functions in Fig. 5.12e exhibit some stretching and squeezing in addition to both high and



(a) (b) (c)

(d) (e)

Figure 5.12: Basis functions in ϕ, when training on (a) Upscaling; (b) Banding; (c) Hits (H.264); (d) Hits (MPEG2); and (e) Incorrect Aspect Ratio.

low frequency details. Basis functions tuned for detecting dropped frames are provided at [17], since they must be viewed as videos.

We further analyzed VIDSPECT by developing an input signal that maximizes detection performance. The input signal was initialized using ran-

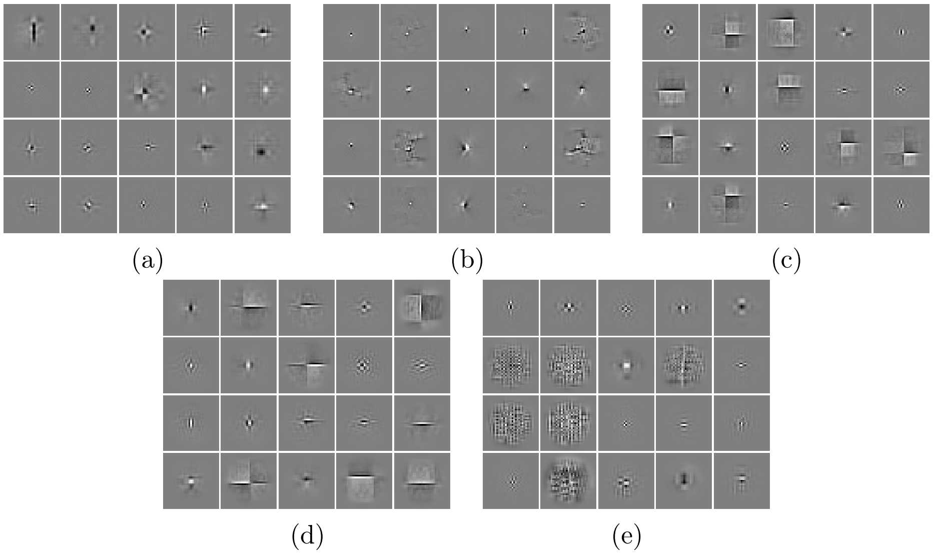


Figure 5.13: Hallucinated distortion patterns. (a) Upscaling; (b) Banding; (c) Hits (H.264); (d) Hits (MPEG2); and (e) Incorrect Aspect Ratio.

dom Gaussian noise. Next, this input signal was iteratively refined to maximize one feature in θ. After enough iterations, the input signal will mimic, or "hallucinate," the artifact. This method has been used to visualize the classification responses of deep convolutional networks [121]. Figure 5.13 depicts hallucination patterns for each distortion type. In Fig. 5.13a, the patterns exhibit the rippling effect associated with upscaling artifacts. Semicircles indicating a high-contrast and a flat region are observed in Fig. 5.13b. Figures 5.13c and 5.13d depict blocking artifacts associated with video hits, where larger blocks are more useful for detecting MPEG2 hits. The hallucinations observed for incorrect aspect ratio in Fig. 5.13e exhibit stretching and squeezing, in addition to more complex patterns. The dropped frames artifact is further analyzed in [17].

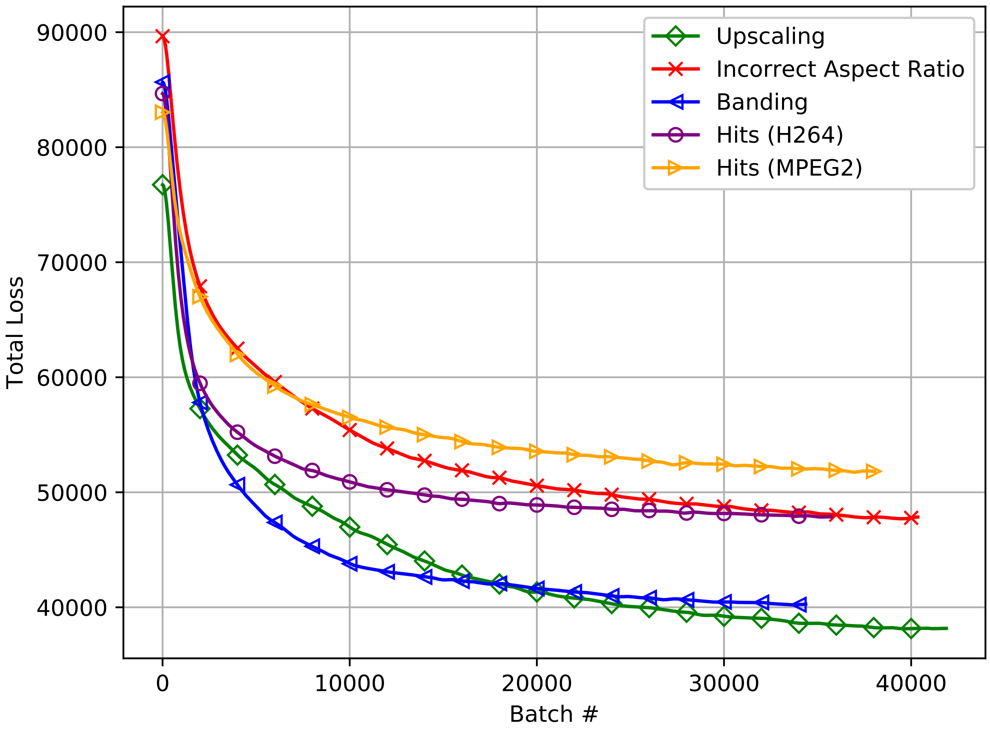


Figure 5.14: Measurement of total loss of VIDSPECT during training.

5.5 Detection Analysis

We now assess the performance of the trained VIDSPECT model for each detection task. To assess binary classification performance, we measured the F1 score, which is the harmonic mean of precision and recall, and the Matthews Correlation Coefficient (MCC), which is a balanced measure related to the chi-square statistic. We opted to have forty 25x25x1 filters in ϕ and twenty 25x25x40 filters in θ, which effectively limits the output feature vector length to twenty elements. To optimize this feature vector for detection, we used a Support Vector Classifier (SVC). Each generated dataset was divided equally into training and testing subsets, all methods are trained on the training subset, and performance is evaluated on the testing subset.

According to the upscaling results in Table 5.14, VIDSPECT was competitive with another specialized domain-specific filtering-based approach in [50]. The generic IQA algorithm BRISQUE was next in order, which performed surprisingly well, followed by Vázquez-Padín *et al.*'s method. Interestingly, the frequency magnitude measurement method [44] performed worst, although upscaling artifacts are theoretically simple to characterize in the frequency domain.

Testing on banding artifacts showed that all of the compared methods performed well, achieving very high F1 and MCC scores. This is not unexpected, since these artifacts are highly distinctive, and generally disrupt very smooth, homogeneous regions along isolated spatial contours.

When evaluated on H.264 hits and MPEG2 hits, the performance of VIDSPECT was the best, followed by that of VIDSPECT-D. The third-best in both cases was the distortion-specific method by Glavota *et. al.*. BRISQUE performed surprisingly well when detecting H.264 hits, but delivered poor performance on MPEG2 hits.

When testing on dropped frame distortions, the detector developed by Upadhyay and Singh [132] delivered the best results. Surprisingly, a modified form of BRISQUE applied to frame differences instead of frames (denoted

BRISQUE-D), was the second best detector of dropped frames, followed by VIDSPECT. Wolf [144] performed worst.

On videos impaired by incorrect aspect ratios, VIDSPECT performed significantly better than the other methods.

5.6 Video Quality Prediction Model

When using VIDSPECT for video quality prediction, we extract a total of 20 features per frame, which represent maximum responses. To reduce the computational burden, we process every 16 frames for Upscaling, Banding, and incorrect aspect ratio. For video hits, we process every two frames, and for dropped frames, we process every frame. To pool these frame-based features for a video sequence, we simply average these features across frames, except for video hits for which we compute the maximum across frames. We have a final feature vector of 20 for each video sequence.

We used the Spearman's Rank Ordered Correlation Coefficient (SRCC), Pearson's Linear Correlation Coefficient (LCC), and Root Mean Squared Error (RMSE) to measure prediction performance, as recommended in [35]. We considered each distortion separately when assessing algorithm performance. For each distortion category, the data was randomly split into 80% training and 20% testing subsets, meaning 20 contents were used for training and the remaining 4 were used for testing. We measured performance on the test set, then randomized the selected contents. Measurements were aggregated over 1000 trials, and the median values computed. For each model that required a machine learning step to remap features to scores, we used a Support Vector Regressor (SVR) with the 'rbf' kernel. The SVR hyper-parameters C and γ were found using grid search and 5-fold cross-validation. We remapped the score predictions using the nonlinear function

$$y = \beta_1(0.5 - \frac{1}{1 + e^{\beta_2(x-\beta_3)}}) + \beta_4 x + \beta_5 \tag{5.12}$$

where x is the subjective score and $\beta_i, i = 1...5$ are fitted parameters solved by minimizing the squared error between y and the ground truth scores [71, 137]. This function was used prior to computing the various performance measures.

As in Table 5.15, we found that VIDSPECT and VIDSPECT-D achieved top performance according to all metrics on Upscaling, Banding, Hits (H.264), Hits (MPEG2), and Incorrect Aspect Ratio, with competitive performance for Dropped Frames. Each of the competing methods performed poorly on Hits (H.264), Hits (MPEG2), and incorrect aspect ratio distorted video.

To determine how well quality can be predicted when severity is known, we included a method named "Oracle," which uses only the distortion severities as features. We found that the performance of this oracle is often high, meaning that a model that can accurately classify distortion severity can be expected to correlate well with subjective quality. VIDSPECT is an excellent distortion detector, and thus an excellent quality predictor for most distortion types.

Since the sparse filter responses are used to detect distortions, and since each filter in each distortion specific filterbank tends to be sensitive to a different aspect of a distortion, we also plotted the correlations of the filter responses against the human opinions in Fig. 5.15. In the plot, the basis functions are sorted according to their absolute correlation with MOS. The plots are quite interesting. For some distortions, many basis responses contribute nearly equally (e.g. Hits (MPEG2)), while for others (e.g. Banding) only a few do. To understand the relatively poor predictive performance of aspect ratio with respect to the oracle, we analyzed VIDSPECT trained on stretching

$$\frac{TargetWidth}{SourceWidth} > 1$$

and separately trained on squeezing

$$\frac{TargetWidth}{SourceWidth} < 1$$

the aspect ratio. We find that even when simplifying the detection task, the correlation between VIDSPECT features and MOS remains low overall. Fig. 5.15 nicely clarifies the relative difficulties of the distortions.

5.7 VIDSPECT System Analysis

Finally, we analyzed the end-to-end performance of the VIDSPECT system, which operates by first ingesting a video, determining which distortion dominates the video, then assessing the video quality given that the distortion is known. We evaluated the first step by using the video database to map sparse filter responses to whole-video detection labels. We combined the responses from each filterbank to perform a 7-class discrimination problem, and divided

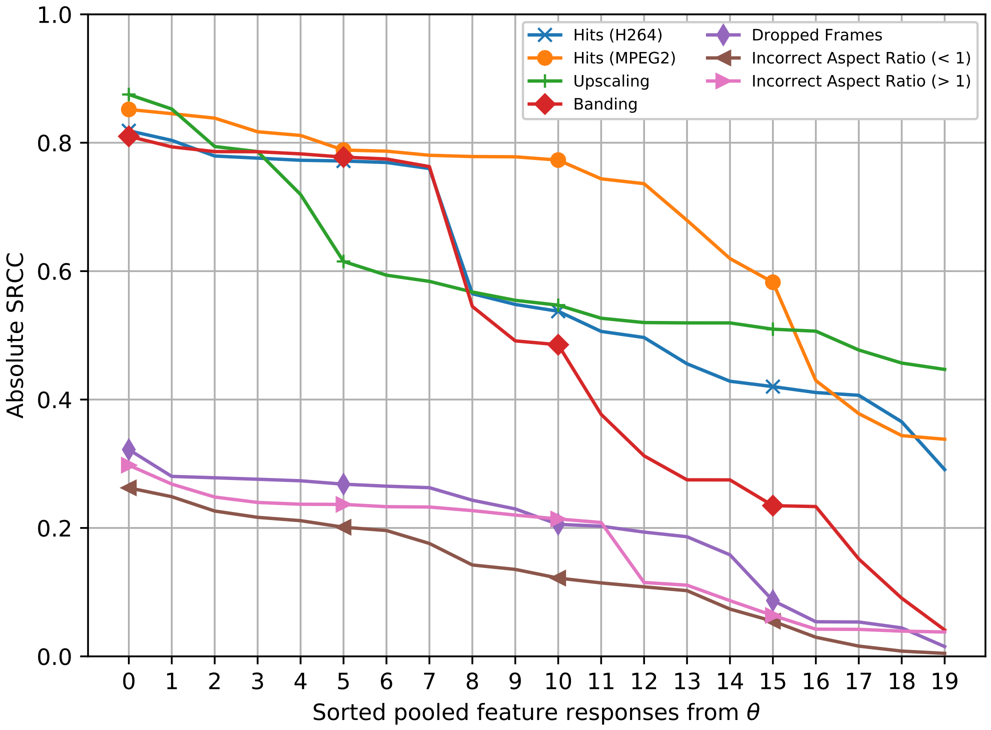


Figure 5.15: Correlations of pooled θ output to human opinion scores. Responses are sorted by correlation magnitude.

the database into training and testing parts, by randomly splitting on content. We left out 4 contents for testing, and trained the SVC discriminator on the rest. The discrimination performance of the overall distortion detection system reached an F1 macro score of 0.7203. When leaving out the two most difficult distortions (dropped frames and incorrect aspect ratio), the problem is reduced to a 5-class problem. For this reduced problem, we observed an F1 macro score of 0.9597.

To evaluate the misclassification rates on each distortion class, we computed a confusion matrix by collecting the test set predictions over 1000 randomized train-test split iterations. We then normalized the confusion matrix row-wise, as shown in Table 5.16. We can see that distortion-free video is difficult to classify and is confused mostly with dropped frames, which are spatially distortion-free. There is some confusion in classifying the two types of video hits, but less confusion than might be expected, since these artifacts overlap somewhat in appearance.

To measure the overall VIDSPECT quality prediction performance, we combined the class discrimination segment with the 6 per-distortion quality prediction modeling segment. We evaluated performance on the test set after training the entire VIDSPECT pipeline on the training set. After 1000 train/test trials, we computed the median performance, obtaining an SRCC of 0.8072, an LCC of 0.9017, and an RMSE of 9.4499. We repeated this test by leaving out the two most difficult distortions (dropped frames and incorrect aspect ratio), obtaining an SRCC of 0.8568, an LCC of 0.8791, and RMSE of 9.0596. These are very promising results, particularly in view of the subtlety of some of the distortions.

5.8 Discussion and Conclusion

We proposed a new, integrated framework for detecting distortions and rating videos based on quality. It effectively uses the responses of tuned filters to detect artifacts with across-the-board state-of-the-art performance. We also showed that the filter responses are excellent indicators of video quality. The distortions can be effectively characterized using the same filterbank in both stages (detection and quality prediction) of the VIDSPECT system.

Future work might include investigating cases where source videos have been multiply distorted. Examples of this include combinations like compression and rescaling to achieve a specific compression ratio, the appearance of interlacing alongside VHS artifacts in legacy content, and combinations of aliasing with aspect ratio changes.

Table 5.14: Detection results evaluated on test datasets. Boldface indicates best performing method.

Distortion Category	Method	F1 score	MCC
Upscaling	VIDSPECT	**0.9902**	**0.9804**
	VIDSPECT-D	0.9789	0.9583
	Goodall [50]	0.9885	0.9769
	BRISQUE [82]	0.9794	0.9585
	Feng et. al. [44]	0.8956	0.7844
	Vázquez-Padín *et al.* [133]	0.9774	0.9546
Banding	VIDSPECT	0.9933	0.9866
	VIDSPECT-D	0.9851	0.9708
	BRISQUE [82]	**0.9954**	**0.9909**
	Luo *et. al.* [75]	0.9903	0.9806
Hits (H.264)	VIDSPECT	**0.9240**	**0.8552**
	VIDSPECT-D	0.9196	0.8478
	BRISQUE [82]	0.8273	0.6467
	AIDB [115]	0.7342	0.4867
	Glavota *et. al.* [49]	0.8794	0.7777
	Winter *et. al.* [142]	0.5521	0.2059
Hits (MPEG2)	VIDSPECT	**0.8420**	**0.6999**
	VIDSPECT-D	0.8081	0.6425
	BRISQUE [82]	0.6342	0.2959
	AIDB [115]	0.6413	0.3124
	Glavota *et. al.* [49]	0.8024	0.6296
	Winter *et. al.* [142]	0.5159	0.1070
Dropped Frames	VIDSPECT-D	0.9033	0.8115
	BRISQUE-D [82]	0.9142	0.8249
	Upadhyay and Singh [132]	**0.9510**	**0.9007**
	Wolf [144]	0.6827	0.2406
Incorrect Aspect Ratio	VIDSPECT	**0.9848**	**0.9700**
	VIDSPECT-D	0.8880	0.7792
	BRISQUE [82]	0.8796	0.7543
	Feng et. al. [44]	0.7177	0.4805

Table 5.15: Median quality prediction results evaluated on 1000 randomized train/test splits.

Distortion Category	Method	SRCC	LCC	RMSE
Upscaling	Oracle	0.9338	0.9518	6.9231
	VIDSPECT	**0.9029**	**0.9317**	**8.1234**
	VIDSPECT-D	**0.9091**	**0.9494**	**7.6651**
	BRISQUE [82]	**0.9029**	**0.9312**	**8.4999**
	Video BLIINDS [108]	0.8811	0.9032	9.5642
	Li *et. al.* [69]	0.8324	0.8442	11.9187
Banding	Oracle	0.8853	0.8803	13.0261
	VIDSPECT	**0.9118**	**0.9186**	**10.4019**
	VIDSPECT-D	0.8824	0.8812	12.8926
	BRISQUE [82]	0.8765	0.8974	12.9704
	Video BLIINDS [108]	**0.9029**	**0.9293**	**11.0157**
	Li *et. al.* [69]	0.8588	0.8762	13.2335
Hits (H.264)	Oracle	0.8574	0.9252	8.7628
	VIDSPECT	0.7972	0.8909	10.8210
	VIDSPECT-D	**0.8531**	**0.8966**	**10.1576**
	BRISQUE [82]	0.2697	0.1388	22.4274
	Video BLIINDS [108]	0.4333	0.3596	21.1707
	Li *et. al.* [69]	0.1189	0.0614	22.3638
Hits (MPEG2)	Oracle	0.9461	0.9617	5.9580
	VIDSPECT	**0.9091**	**0.9363**	**7.8349**
	VIDSPECT-D	**0.8951**	**0.9200**	**7.8652**
	BRISQUE [82]	0.3147	0.1848	19.2417
	Video BLIINDS [108]	0.3636	0.2718	18.9376
	Li *et. al.* [69]	0.3497	0.2742	18.8895
Dropped Frames	Oracle	0.6427	0.6361	6.1218
	VIDSPECT-D	0.2108	0.1747	7.3637
	BRISQUE-D [82]	**0.3912**	**0.3955**	**7.1888**
	Video BLIINDS [108]	0.1054	-0.0143	7.3345
	Li *et. al.* [69]	0.1054	0.1227	7.3218
Incorrect Aspect Ratio	Oracle	0.8278	0.8129	6.2237
	VIDSPECT	**0.4930**	**0.5015**	**8.6907**
	VIDSPECT-D	0.1861	0.1491	10.2052
	BRISQUE [82]	0.1538	0.1393	10.3758
	Video BLIINDS [108]	0.1608	0.1574	10.2469
	Li *et. al.* [69]	0.0280	0.0121	10.3309

Table 5.16: Normalized confusion matrix computed using VIDSPECT to classify videos in the Video Masters database. Class prediction probabilities are averaged over 1000 trials. Boldface indicates highest predicted value. Aspect Ratio is abbreviated as AR.

		Predicted Distortion						
		Banding	Hits (H.264)	Hits (MPEG2)	Upscaling	Dropped Frames	Incorrect AR	Distortion-Free
Ground Truth	Banding	**0.984**	0.011	0.003	0.000	0.002	0.000	0.000
	Hits (H.264)	0.000	**0.883**	0.055	0.000	0.056	0.007	0.000
	Hits (MPEG2)	0.000	0.096	**0.869**	0.000	0.034	0.002	0.000
	Upscaling	0.000	0.000	0.000	**1.000**	0.000	0.000	0.000
	Dropped Frames	0.002	0.023	0.013	0.000	**0.759**	0.203	0.000
	Incorrect AR	0.000	0.019	0.012	0.003	0.344	**0.621**	0.000
	Distortion-Free	0.001	0.017	0.013	0.000	**0.752**	0.218	0.000

Chapter 6

Conclusion and Future Work

In this dissertation, we proposed a No-Reference (NR) video source inspection concept called VIDMAP, which is able to effectively learn how to detect and localize multiple types of video artifacts without using *a priori* models of the statistics or structures of the artifacts. We showed that VIDMAP achieves state-of-the-art detection performance in most categories tested, with competitive performance in the others. It is a practical tool that also assists a user in visualizing distortion types, locations, and severities. We envision that this model will be useful as a tool for source inspection of streaming video collections.

We also proposed VIDSPECT, a new, integrated framework for detecting distortions and rating videos based on quality. This framework is developed in two stages, a detection stage and a quality assessment stage. We showed that both stages performed well, and we also showed that the VIDSPECT output responses are excellent indicators of video quality. Distortions can be effectively characterized using the same set of filterbanks in both stages (detection and quality prediction) of the VIDSPECT system.

Last, we proposed the LIVE Video Masters database, which contains a number of distortion types that are highly relevant to modern digital video streaming companies. We believe that sharing this database will provide an invaluable resource for those developing and evaluating source inspection systems similar to VIDSPECT.

A number of changes to VIDMAP can be explored. First, the convolutional network can easily extended to multi-class problems, to predict amongst an array of distortions, like VIDSPECT. Second, it can be extended to regression problems to offer subjective quality prediction, noise severity predictions, or even a measure of distance between two groups of videos. Third, this convolutional network architecture can be rearranged to take advantage of the first filter layer, which holds many low level filter operators common amongst artifacts. This would serve to reduce computational overhead when predicting responses to more than one artifact, and it could be possible that the amount of training data can be effectively reduced, while maintaining high performance.

Both VIDSPECT and VIDMAP prediction models can be extended using deeper spatio-temporal pre-processing models that better decorrelate data. The perceptual pre-processing of the human visual system begins at the complex retinal layers of the eye, which has been observed to reduce the entropy of signals both spatially and temporally before information is delivered through Magnocellular and Parvocellular pathways post-retina. Training both detection models after this pre-processing should improve representational capacity, leading to reductions in complexity. A richer pre-processing stage that mimics low-level human vision allows for stronger assumptions regarding natural scenes statistics in the visual signal, which should be also be more informative when learning to detect artifacts.

Future work related to source inspection might include investigating cases where source videos have been multiply distorted. Examples of this include combinations like compression and rescaling to achieve a specific compression ratio. Another real-world example includes modeling the appearance of interlacing alongside VHS artifacts in legacy content, and combinations of aliasing with aspect ratio changes. Along the same lines, a particular manifestation of a distortion may mimic another type of distortion, meaning that there can be inherent ambiguity when discriminating these two distortion types. Representing both similarities and differences amongst distortion representations should yield more effective predictors.

Studying other modalities is also of interest. Artifacts that appear in hand-drawn cartoon videos do not necessarily follow the natural scene statistics that we previously described. It has been observed that existing detection models, such as those designed for detecting upscaling or combing, fail for these hand-drawn contents. During ingest, streaming companies often observe these contents, making their study an important next step. In addition, images which are formed using completely different capture methods may not follow usual NSS models, thus it would be of interest to investigate how VIDMAP/VIDSPECT might be used in these modalities.

Appendices

Appendix A

Sparse Functional in ADMM Form

The basis pursuit denoising (BPDN) functional is given by

$$(X^*, \phi^*) = \underset{X,\phi}{\operatorname{argmin}} \frac{1}{2} \left\| S - \sum_k \phi_k X_k \right\|_2^2 + \lambda \left\| X \right\|_1 \tag{A.1}$$

with constraints

$$\left\| \phi_k \right\|_2 \leq 1$$

and

$$X_i \geq 0,$$

where X is the coding matrix, the Lagrangian scalar multiplier λ, S is the input signal, and ϕ is the dictionary. The codes X are constrained positive and each element in the dictionary is normalized using the ℓ_2 norm. This functional cannot be optimized directly with the use of gradient descent since the ℓ_1 norm is not differentiable at every point.

Since equation A.1 is convex in X while ϕ is fixed, we can rewrite equation A.1 to take advantage of the Alternating Direction Method of Multipliers (ADMM) [32] to solve the ℓ_1 minimization with constraints, which is designed to solve problems in the form

$$\begin{aligned} &\text{minimize } f(x) + g(z) \\ &\text{subject to } Ax + Bz = c \end{aligned}$$

where $f(x)$ and $g(z)$ are convex.

The ADMM form of equation A.1 is

$$\begin{aligned} &\min \left[\frac{1}{2} \left\| S - \sum_k \phi_k X_k \right\|_2^2 + \lambda \left\| Z_1 \right\|_1 \right] \\ &\text{subject to } X = Z_2, X = Z_1, Z_2 \geq 0 \end{aligned} \tag{A.2}$$

where Z_1 and Z_2 are new variables upon which we apply the sparsity and nonnegative constraints respectively.

The optimization in equation A.2 can be rewritten as an augmented Lagrangian

$$\begin{aligned} L(X, Z_1, Z_2, U_1, U_2) = &\frac{1}{2} \left\| S - \sum_k \phi_k X_k \right\|_2^2 + \lambda \left\| Z_1 \right\|_1 \\ &+ \langle U_1, X - Z_1 \rangle + \frac{\rho}{2} \left\| X - Z_1 \right\|_2^2 \\ &+ \langle U_2, X - Z_2 \rangle + \frac{\rho}{2} \left\| X - Z_2 \right\|_2^2 \end{aligned} \tag{A.3}$$

where U_1 and U_2 are dual variables that correspond to Z_1 and Z_2 respectively, and ρ is the step-size parameter. The ADMM algorithm specifies how to perform the minimization of Eq. A.3 for each variable. The iterative update for $X^{(k+1)}$ is given by

$$X^{(k+1)} = \min_X L(X, Z_1, Z_2, U_1, U_2) \tag{A.4}$$

which has the closed form solution

$$X^{(k+1)} = \left(\phi\phi^T + 2\rho I\right)^{-1} \left(\phi S^T + \rho\left(Z_1 + Z_2\right) - U_1 - U_2\right)$$

where variables Z_1 and Z_2 are subsequently updated as

$$\begin{aligned} Z_1^{(k+1)} =& \min_{Z_1} L(X, Z_1, Z_2, U_1, U_2) \qquad \text{(A.5)} \\ =& S_{\lambda/\rho}\left(X^{(k+1)} + \frac{1}{\rho}U_1^{(k)}\right) \\ Z_2^{(k+1)} =& \min_{Z_2} L(X, Z_1, Z_2, U_1, U_2) \qquad \text{(A.6)} \\ =& \max\left(X^{(k+1)} + \frac{1}{\rho}U_2^{(k)}, 0\right) \qquad \text{(A.7)} \end{aligned}$$

where S_λ is the soft thresholding operator defined as

$$S_\lambda(v) = (v - \lambda)_+ - (-v - \lambda)_+$$

and the dual variables are updated according to

$$\begin{aligned} U_1^{(k+1)} &= U_1^{(k)} + \rho\left(X^{(k+1)} - Z_1^{(k+1)}\right) \qquad \text{(A.8)} \\ U_2^{(k+1)} &= U_2^{(k)} + \rho\left(X^{(k+1)} - Z_2^{(k+1)}\right). \qquad \text{(A.9)} \end{aligned}$$

Convergence is reached when the primal and dual residuals are each sufficiently small (≤ 0.001). The primal residuals are defined as

$$\begin{aligned} p_1 &= \left\|X^{(k+1)} - Z_1^{(k+1)}\right\|_2^2 \\ p_2 &= \left\|X^{(k+1)} - Z_2^{(k+1)}\right\|_2^2 \end{aligned}$$

and the dual residuals are defined by

$$\begin{aligned} d_1 &= \left\| \rho \left(Z_1^{(k+1)} - Z_1^{(k)} \right) \right\|_2^2 \\ d_2 &= \left\| \rho \left(Z_2^{(k+1)} - Z_2^{(k)} \right) \right\|_2^2 . \end{aligned}$$

The above algorithm can be simplified by observing that the nonnegative and sparse constraints can be combined. The sparsity constraint can inorporate non-negativity by dropping the negative component of the soft thresholding operator. The terms related to Z_2 and U_2 are thus redundant and can be removed from the algorithm.

Appendix B

Combing Detection Algorithms

We provide a description of the compared combing detection algorithms used in the context of our analysis, which involves identification of combing within a single frame i. Algorithm 1 is derived from the source code for FFmpeg's "idet" filter [5]. Although Algorithm 2 is described by Baylon [28] (in terms of separate top-field-first/bottom-field-first detection), we include the algorithm here for clarity and for the sake of comparison.

Algorithm 1 FFmpeg's combing detector

```
Given 3 frames $V_{i-1}$, $V_i$, and $V_{i+1}$ of shape $(M, N)$
$M :=$ height and $N :=$ width
$T_1 :=$ detection threshold

$t \leftarrow 0$, $b \leftarrow 0$
for $k \in \{1, M-1\}$ do
    if $k \mod 2$ is 0 then
        $t = t + V_i(k-1) + V_i(k+1) - 2V_{i-1}(k)$
        $b = b + V_i(k-1) + V_i(k+1) - 2V_{i+1}(k)$
    else
        $t = t + V_i(k-1) + V_i(k+1) - 2V_{i+1}(k)$
        $b = b + V_i(k-1) + V_i(k+1) - 2V_{i-1}(k)$
    end if
end for
if $t > T_1 b$ or $b > T_1 t$ then
    Detect positive
else
    Detect negative
end if
```

Algorithm 2 Baylon's combing detector

```
Given 2 frames $V_{i-1}$ and $V_i$ of shape $(M, N)$
$M :=$ height and $N :=$ width
$Z :=$ zipper filter length
$T_0$, $T_1 :=$ thresholds
$h[j] = (-1)^j \ \forall \ j \in \{-Z/2+1, Z/2\}$

Construct $x_0$ and $x_1$:
    for $k \in \{0, \frac{M-1}{2}\}$ do
        $x_0(2k) = V_{i-1}(2k)$
        $x_0(2k+1) = V_i(2k+1)$
        $x_1(2k) = V_i(2k)$
        $x_1(2k+1) = V_{i-1}(2k+1)$
    end for
Convolve across rows using zipper filter:
    $y_0 = |h * x_0|$
    $y_1 = |h * x_1|$
$C_0 = \sum \mathbb{1}_{y_0 > T_0}$ and $C_1 = \sum \mathbb{1}_{y_1 > T_0}$
if $C_0 > T_1 C_1$ or $C_1 > T_1 C_0$ then
    Detect positive
else
    Detect negative
end if
```

Appendix C

Instructions for Subjects

C.1 General Instructions

Your task is to judge the quality of each video sequence and not the content of the sequence. There is no right answer in this experiment. Please rely on your own judgment. This study is divided into three separate test sessions. Each test session will be preceded by a short training session and lasts approximately 40 minutes. At the end of each video you will be asked to provide a quality score between bad and excellent, where labels are shown in Table C.1.

During the study, please select a comfortable viewing distance of about 2 feet. Please remain upright in your seat and look directly at the monitor. You can move around a little to stay comfortable, but try to keep your viewing distance and angle as constant as possible, because the videos might look a little different from different positions, and weâĂŹd like everyone to judge the videos from about the same position. You might reposition your chair to achieve this comfortably.

Table C.1: Qualitative levels of the continuous video rating scale.

Level	Video Quality
2	Excellent
1	Bad

Table C.2: Distortion types

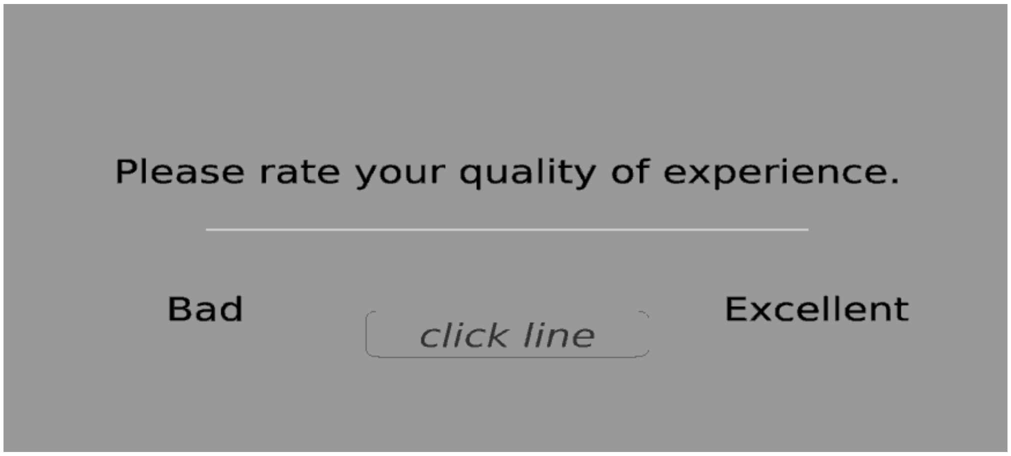
Distortion Type	Description
Upscaling	The video appears to be of low resolution.
Incorrect Aspect Ratio	The width and/or height of the video is not appropriately scaled resulting in objects that appear stretched and/or squished.
Video Hits	The video appears to contain corruptions.
Banding	Smooth areas (e.g. sky) appear to contain false contours. Gradients are not well represented.

C.2 How to Score the Videos

The system will guide your viewing and rating of the videos. For each session, you will be asked to evaluate quality of experience based on the possible presence of different types of distortions. The distortion types you will see are listed in Table C.2.

Note that we will not ask you to identify any distortions in the video. However, we ask that you rate the video quality of each presentation qualitatively, based on your perception, holistically. Your opinion will evaluated at the end of each video using a sliding bar, as shown in the screenshot below.

You should take approximately 10 seconds to decide and enter your response (your opinion on the quality of the video). The evaluation screen will only be displayed after a video has been completely viewed. After you submit



Please rate your quality of experience.

the quality score and are ready to continue, the system will guide you to the next video. Remember, your quality score should include everything that they see in the video, even if you think that a distortion was not intended by us.

C.3 Training

The training videos shown at the beginning of each session are meant to give you practice viewing and rating videos with particular distortions. In each training session, you will get a sense of how videos with artifacts appear as well as how to use the GUI. Please ask any questions during and after the training session.

Bibliography

[1] 2015 Will Be the Year Netflix Goes âĂŸFull HBOâĂŹ. http://time.com/collection-post/3675669/netflix-hbo/. accessed Mar 2017.

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

[3] ImageMagick. http://www.imagemagick.org/script/index.php.

[4] Interlace Detection. http://avisynth.nl/index.php/Interlace_detection. AVISynth, accessed Mar 2017.

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

[6] List of Error Types. https://backlothelp.netflix.com/content/issue-types#overlay-context=content/issue-types.

[7] LIVE Video Masters Database. http://live.ece.utexas.edu/research/Quality/live_video_masters.html.

[8] Netflix Backlot Pages. https://backlothelp.netflix.com/hc/en-us/sections/203516117-Manual-QC-Error-Messages?page=1#articles.

[9] Netflix Producing More Hours Of Original Content Than HBO. http://sanfrancisco.cbslocal.com/2016/04/19/netflix-producing-more-hours-original-c accessed Mar 2017.

[10] Netflix Q1 2015 Earnings Letter to Shareholders. http://files.shareholder.com/downloads/NFLX/47469957x0x821407/DB785B50-90FE-44DA-9F5B-37DBF0DCD0E1/Q1_15_Earnings_Letter_final_tables.pdf.

[11] Netflix will release 1,000 hours of original programming in 2017. http://bgr.com/2016/10/19/netflix-originals-1000-hours-programming/. accessed Mar 2017.

[12] Srestore. http://avisynth.nl/index.php/Srestore. Avisynth, accessed Jan 2016.

[13] Study: Amazon Video is now the third-largest streaming service, behind Netflix and YouTube. http://www.geekwire.com/2016/study-amazon-video-now-third accessed Mar 2017.

[14] The Number of Movies on Netflix Is Dropping Fast. http://time.com/4272360/the-number-of-movies-on-netflix-is-dropping-fast/. accessed Mar 2017.

[15] Video hits.

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

[17] VIDSPECT code. http://live.ece.utexas.edu/research/VIDSPECT/VIDSPECT.html.

[18] You know whatâĂŹs cool? A billion hours. https://youtube.googleblog.com/2017/02/you-know-whats-cool-billion-hours.html. accessed Mar 2017.

[19] YouTube Could Be About to Overtake TV as AmericaâĂŹs Most Watched Platform. http://fortune.com/2017/02/28/youtube-1-billion-hours-television/. accessed Mar 2017.

[20] Youtube Official Blog. https://youtube.googleblog.com/. accessed Mar 2017.

[21] YouTubeâĂŹs Pay Video Service Is Already Storming Past Other New Offerings. http://fortune.com/2016/08/31/youtube-red-streaming-video/. accessed Mar 2017.

[22] ITU-BTR13, 2017.

[23] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[24] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11), 2006.

[25] Wonseok Ahn and Jae-Seung Kim. Flat-region detection and false contour removal in the digital tv display. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1338–1341. IEEE, 2005.

[26] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[27] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. L0 norm based dictionary learning by proximal methods with global convergence. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3858–3865, 2014.

[28] David M Baylon. On the detection of temporal field order in interlaced video data. *IEEE Int'l. Conf. Image Process.*, 6:VI–129, 2007.

[29] Loris Bazzani, Alessandra Bergamo, Dragomir Anguelov, and Lorenzo Torresani. Self-taught object localization with deep networks. In *2016 Winter Conference on Applications of Computer Vision*. IEEE, 2016.

[30] Oren Boiman and Michal Irani. Detecting irregularities in images and in video. *International Journal of Computer Vision*, 74(1):17–31, 2007.

[31] A. C. Bovik. Automatic prediction of perceptual image and video quality. *Proc. IEEE*, 101(9):2008–2024, 2013.

[32] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[33] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[34] Eunjung Chae, Eunsung Lee, Wonseok Kang, Hejin Cheong, and Joonki Paik. Spatially adaptive antialiasing for enhancement of mobile imaging system using combined wavelet-fourier transform. *IEEE Transactions on Consumer Electronics*, 59(4):862–868, 2013.

[35] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J Karam. Objective video quality assessment methods: A classification, review, and performance comparison. *IEEE Transactions on Broadcasting*, 57(2):165–182, 2011.

[36] Lark Kwon Choi, Jaehee You, and A.C. Bovik. Referenceless prediction of perceptual fog density and perceptual image defogging. *IEEE Trans. Image Process.*, 24(11):3888–3901, Nov 2015.

[37] Vincent Christlein, Christian Riess, Johannes Jordan, Corinna Riess, and Elli Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Trans. Info. Forensics Sec.*, 7(6):1841–1854, 2012.

[38] M. Clark and A. C. Bovik. Experiments in segmenting texture patterns using localized spatial filters. *Pattern Recognition*, 22(6):707–717, 1989.

[39] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[40] Baptiste Coulange and Lionel Moisan. An aliasing detection algorithm based on suspicious colocalizations of fourier coefficients. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2013–2016. IEEE, 2010.

[41] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[42] T. Dumas, A. Roumy, and C. Guillemot. Image compression with stochastic winner-take-all auto-encoder. *ICASSP*, 2017.

[43] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[44] Xiaoying Feng, Ingemar J Cox, and Gwenael Doerr. Normalized energy density-based forensic detection of resampled images. *IEEE Trans Multimedia*, 14(3):536–545, 2012.

[45] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Amer. A*, 4(12):2379–2394, 1987.

[46] Andrew C Gallagher. Detection of linear and cubic interpolation in jpeg compressed images. *Canadian Conf. Computer Robot Vision*, pages 65–72, 2005.

[47] Deepti Ghadiyaram and Alan 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.

[48] Deepti Ghadiyaram and Alan C Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1):372–387, 2016.

[49] Ivan Glavota, Mario Vranješ, Marijan Herceg, and Ratko Grbić. Pixel-based statistical analysis of packet loss artifact features. In *Zooming Innovation in Consumer Electronics International Conference (ZINC)*, pages 16–19, 2016.

[50] T. Goodall, I. Katsavounidis, Z. Li, A. Aaron, and A. C. Bovik. Blind picture upscaling ratio prediction. *IEEE Signal Process Lett*, 23(12):1801–1805, 2016.

[51] Anthony 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.

[52] Peter JB Hancock, Roland J Baddeley, and Leslie S Smith. The principal components of natural images. *Network: Computation in Neural Systems*, 3(1):61–70, 1992.

[53] Y.W.L. Hui. Progressive/interlace and redundant field detection for encoder, March 22 2005. US Patent 6,870,568.

[54] Dai-Kyung Hyun, Seung-Jin Ryu, Hae-Yeoun Lee, and Heung-Kyu Lee. Detection of upscale-crop and partial manipulation in surveillance video based on sensor pattern noise. *Sensors*, 13(9):12605–12631, 2013.

[55] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1697–1704. IEEE, 2011.

[56] Ioannis Katsavounidis, Anne Aaron, and David Ronca. Native resolution detection of video sequences. In *Annual Technical Conference and Exhibition*, pages 1–20, 2015.

[57] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[58] Sune Keller, Kim S Pedersen, and François Lauze. Detecting interlaced or progressive source of video. In *Multimedia Signal Processing*, pages 1–4. IEEE, 2005.

[59] Jin-Tae Kim and Chang-Hee Joo. Analysis method of digital forgeries on the filtered tampered images. *Journal of information and communication convergence engineering*, 9(1):95–99, 2011.

[60] Jongyoo Kim, Hui Zeng, Deepti Ghadiyaram, Sanghoon Lee, Lei Zhang, and Alan C Bovik. Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment. *IEEE Signal Processing Magazine*, 34(6):130–141, 2017.

[61] Matthias Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. *ACM Wkshp. Multimedia Sec.*, pages 11–20, 2008.

[62] Matthias Kirchner and Rainer Böhme. Tamper hiding: Defeating image forensics. *Int'l Wkshp Info Hiding*, pages 326–341, 2007.

[63] Robert 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.

[64] Eric C Larson and Damon M Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of*

Electronic Imaging, 19(1):011006–011006, 2010.

[65] Eric C. Larson and Damon M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006, 2010.

[66] Ji Won Lee, Bo Ra Lim, Rae-Hong Park, Jae Seung Kim, and Wonseok Ahn. Two-stage false contour detection algorithm using re-quantization and directional contrast features and its application to adaptive false contour reduction. In *Consumer Electronics, 2006. ICCE'06. 2006 Digest of Technical Papers. International Conference on*, pages 377–378. IEEE, 2006.

[67] Qiaohong Li, Weisi Lin, Jingtao Xu, and Yuming Fang. Blind image quality assessment using statistical structural and luminance features. *IEEE Transactions on Multimedia*, 18(12):2457–2469, 2016.

[68] Renxiang Li, Bing Zheng, and Ming L Liou. Reliable motion detection/compensation for interlaced sequences and its applications to deinterlacing. *IEEE Trans. Circ. Syst. Video Technol.*, 10(1):23–29, 2000.

[69] X. Li, Q. Guo, and X. Lu. Spatiotemporal statistics for video quality assessment. *IEEE Transactions on Image Processing*, 25(7):3329–3342, July 2016.

[70] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[71] Joe Yuchieh Lin, Rui Song, Chi-Hao Wu, TsungJung Liu, Haiqiang Wang, and C-C Jay Kuo. Mcl-v: A streaming video quality assessment database. *Journal of Visual Communication and Image Representation*, 30:1–9, 2015.

[72] Bao-Di Liu, Bin Shen, Liangke Gui, Yu-Xiong Wang, Xue Li, Fei Yan, and Yan-Jiang Wang. Face recognition using class specific dictionary learning for sparse representation and collaborative representation. *Neurocomputing*, 204:198–210, 2016.

[73] Bao-Di Liu, Yu-Xiong Wang, Yu-Jin Zhang, and Yin Zheng. Discriminant sparse coding for image classification. In *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 2193–2196, 2012.

[74] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[75] Weiqi Luo, Yuangen Wang, and Jiwu Huang. Detection of quantization artifacts and its applications to transform encoder identification. *IEEE Transactions on Information Forensics and Security*, 5(4):810–815, 2010.

[76] Babak Mahdian and Stanislav Saic. Blind authentication using periodic properties of interpolation. *IEEE Trans. Info. Forensics Sec.*, 3(3):529–538, 2008.

[77] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, pages 1033–1040, 2009.

[78] Alireza Makhzani and Brendan J Frey. Winner-take-all autoencoders. *Adv Neural Info Process Syst*, pages 2791–2799, 2015.

[79] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings IEEE International Conference on Computer Vision*, 2:416–423, 2001.

[80] A. Mittal, M. A. Saad, and A. C. Bovik. A completely blind video integrity oracle. *IEEE Trans. Image Process.*, 25(1):289–300, 2016.

[81] A. Mittal, R. Soundararajan, and Alan C. Bovik. Making a completely blind image quality analyzer. *IEEE Signal Process. Lett.*, 20(3):209–212, 2013.

[82] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.*, 21(12):4695–4708, 2012.

[83] Anish Mittal, Gautam S Muralidhar, Joydeep Ghosh, and Alan C Bovik. Blind image quality assessment without human training using latent quality factors. *IEEE Signal Processing Letters*, 19(2):75–78, 2012.

[84] Anish Mittal, Ravi Soundararajan, and Alan C Bovik. Making a âĂIJ-completely blindâĂİ image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[85] A. K. Moorthy and A. C. Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, 2010.

[86] Anush Krishna Moorthy and Alan Conrad Bovik. Visual quality assessment algorithms: what does the future hold? *Multimedia Tools and Applications*, 51(2):675–696, 2011.

[87] Anush Krishna Moorthy, Anish Mittal, and Alan Conrad Bovik. Perceptually optimized blind repair of natural images. *Signal Processing: Image Communication*, 28(10):1478–1493, 2013.

[88] Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 2011.

[89] Tian-Tsong Ng and Shih-Fu Chang. A data set of authentic and spliced image blocks. Technical report, ADVENT Technical Report, #203-2004-3, Columbia University, 2004.

[90] Jiquan Ngiam, Zhenghao Chen, Sonia A Bhaskar, Pang W Koh, and Andrew Y Ng. Sparse filtering. *Adv. Neural Info. Process. Syst.*, pages 1125–1133, 2011.

[91] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In

IEEE Conf. on Computer Vision and Pattern Recog., pages 427–436, 2015.

[92] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

[93] Kiyotaka Otsuji and Yoshinobu Tonomura. Projection-detecting filter for video cut detection. *Multimedia Systems*, 1(5):205–210, 1994.

[94] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Mach. Learning Res.*, 12:2825–2830, 2011.

[95] Sara Perez. A sneak peek at Disney's streaming service lineup. `https://techcrunch.com/2018/02/09/disneys-streaming-service-wont-have-r-rated-fil`
accessed Mar 2018.

[96] Stefan Pfennig and Matthias Kirchner. Spectral methods to determine the exact scaling factor of resampled digital images. *Int'l Symp. Comm. Control Signal Process.*, pages 1–6, 2012.

[97] Manish Pindoria and Tim Borer. Automatic interlace or progressive video discrimination. *Annual Technical Conference & Exhibition, SMPTE 2012*, pages 1–8, 2012.

[98] Nikolay Ponomarenko, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Lina Jin, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, et al. Color image database tid2013: Peculiarities and preliminary results. In *Visual Information Processing (EUVIP)*, pages 106–111. IEEE, 2013.

[99] Nikolay Ponomarenko, Vladimir Lukin, Alexander Zelensky, and Karen Egiazarian. Tid2008-a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics*, pages 30–45, 2009.

[100] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Trans. Signal Process.*, 53(2):758–767, 2005.

[101] Santasriya Prasad and KR Ramakrishnan. On resampling detection and its application to detect image tampering. *IEEE Int'l Conf Multimedia Expo*, pages 1325–1328, 2006.

[102] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3501–3508, 2010.

[103] Amy R Reibman and Shan Suthaharan. A no-reference spatial aliasing measure for digital image resizing. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1184–1187. IEEE, 2008.

[104] Adriana Romero, Petia Radeva, and Carlo Gatta. No more meta-parameter tuning in unsupervised sparse feature learning. *arXiv preprint arXiv:1402.5766*, 2014.

[105] Adriana Romero, Petia Radeva, and Carlo Gatta. Meta-parameter free unsupervised sparse feature learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1716–1722, 2015.

[106] Daniel L Ruderman. The statistics of natural images. *Network: Comput Neural Syst*, 5(4):517–548, 1994.

[107] Seung-Jin Ryu and Heung-Kyu Lee. Estimation of linear transformation by analyzing the periodicity of interpolation. *Pattern Recog. Lett.*, 36:89–99, 2014.

[108] M. A. Saad, A. C. Bovik, , and C. Charrier. Blind prediction of natural video quality. *IEEETIP*, 23(3):1352–1365, 2010.

[109] R. Sakurai, S. Yamane, and J. H. Lee. Restoring aspect ratio distortion of natural images with convolutional neural network. *IEEE Transactions on Industrial Informatics*, PP(99):1–1, 2018.

[110] Ronald C Schloss. Bee on Flower. `https://archive.org/details/BeeOnFlowerHd1080i`. accessed Apr 2017.

[111] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, local-

ization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[112] Kalpana Seshadrinathan and Alan C Bovik. Motion-based perceptual quality assessment of video. In *IS&T/SPIE Electronic Imaging*, pages 72400X–72400X. International Society for Optics and Photonics, 2009.

[113] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan C Bovik, and Lawrence K Cormack. Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing*, 19(6):1427–1441, 2010.

[114] Muhammed Shabeer P., Saurabhchand Bhati, and Sumohana S. Channappayya. Modeling sparse spatio-temporal representations for no-reference video quality assessment. *GlobalSIP*, 2017.

[115] Dor Shabtay, Nissan Raviv, and Yair Moshe. Video packet loss concealment detection based on image content. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE, 2008.

[116] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. Live image quality assessment database release 2. `http://live.ece.utexas.edu/research/quality`.

[117] Hamid R Sheikh, Muhammad F Sabir, and Alan C Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Process.*, 15(11):3440–3451, 2006.

[118] Xiaohui Shen and Ying Wu. A unified approach to salient object detection via low rank matrix recovery. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 853–860. IEEE, 2012.

[119] Natalie Sherman. Netflix tunes into subscriber surge. `http://www.bbc.com/news/business-42779953`. accessed Feb 2018.

[120] Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Ann. Rev. Neurosci.*, 24(1):1193–1216, 2001.

[121] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[122] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[123] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. *arXiv preprint arXiv:1704.04232*, 2017.

[124] Rajiv Soundararajan and Alan C Bovik. Rred indices: Reduced reference entropic differencing for image quality assessment. *IEEE Transactions on Image Processing*, 21(2):517–526, 2012.

[125] Rajiv Soundararajan and Alan C Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):684–694, 2013.

[126] Todd Spangler. Netflix Eyeing Total of About 700 Original Series in 2018. `http://variety.com/2018/digital/news/netflix-700-original-series-2018-` accessed Mar 2018.

[127] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Mach. Learning Res.*, 15(1):1929–1958, 2014.

[128] Nikola Teslic, Vladimir Zlokolica, Vukota Pekovic, Tarkan Teckan, and Miodrag Temerinac. Packet-loss error detection system for dtv and set-top box functional testing. *IEEE Transactions on Consumer Electronics*, 56(3), 2010.

[129] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[130] DJ Tolhurst, Y Tadmor, and Tang Chao. Amplitude spectra of natural images. *Ophthalmic and Physiological Optics*, 12(2):229–232, 1992.

[131] B. Turkus. Drop video file(s) here: the emergence of free quality control tools for video preservation, 2015.

[132] Saurabh Upadhyay and Sanjay K Singh. Learning based video authentication using statistical local information. In *International Conference on Image Information Processing (ICIIP)*, pages 1–6, 2011.

[133] David Vázquez-Padín, P. Comesaña, and Fernando Pérez-González. An SVD approach to forensic image resampling detection. *EUSIPCO*, pages 2067–2071, 2015.

[134] David Vázquez-Padín and Fernando Pérez-González. Prefilter design for forensic resampling estimation. *IEEE Int'l Wkshp Info Forensics Sec.*, pages 1–6, 2011.

[135] Toni Virtanen, Mikko Nuutinen, Mikko Vaahteranoksa, Pirkko Oittinen, and Jukka Häkkinen. Cid2013: a database for evaluating no-reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 24(1):390–402, 2015.

[136] Cuong Vu, Thien Phan, and Damon Chandler. Can current image quality assessment algorithms predict visual quality of enhanced images?

[137] Wei Wang, Yuanlin Zheng, Kaiyang Liao, Li Liu, and Zhisen Tang. A novel blind jpeg image quality assessment based on blockiness and the low frequency feature in dct domain. In *Applied Sciences in Graphic Communication and Packaging*, pages 169–174. Springer, 2018.

[138] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Online convolutional sparse coding. *arXiv preprint arXiv:1706.06972*, 2017.

[139] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[140] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. IEEE, 2003.

[141] Arthur A Webster, Coleen T Jones, Margaret H Pinson, Stephen D Voran, and Stephen Wolf. Objective video quality assessment system based on human perception. In *Human Vision, Visual Processing, and Digital Display IV*, volume 1913, pages 15–27. International Society for Optics and Photonics, 1993.

[142] Martin Winter, Peter Schallauer, Albert Hofmann, and Hannes Fassold. Efficient video breakup detection and verification. *Info. Extract. Media Product.*, pages 63–68, 2010.

[143] Brendt Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Trans. Image Process.*, 25(1):301–315, 2016.

[144] Stephen 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.

[145] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[146] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, Nov 2010.

[147] Jianchao Yang, Kai Yu, and Thomas Huang. Supervised translation-invariant sparse coding. In *IEEE Conf Computer Vision Pattern Recogn*, 2010.

[148] M. Yang, L. Chen, and J. Tian. A training data cleaning framework for video distortion diagnosis. In *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 214–217, 2017.

[149] M. Yang, L. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *CVPR 2011*, pages 625–632, June 2011.

[150] Hojatollah Yeganeh, Mohammad Rostami, and Zhou Wang. Objective quality assessment of interpolated natural images. *IEEE Trans. Image Process.*, 24(11):4651–4663, 2015.

[151] Anđela Zarić, Nenad Tatalović, Nikolina Brajković, Hrvoje Hlevnjak, Matej Lončarić, Emil Dumić, and Sonja Grgić. VCL@FER image quality assessment database. *AUTOMATIKA*, 53(4):344–354, 2012.

[152] Lin Zhang, Lei Zhang, and Alan C Bovik. A feature-enriched completely blind image quality evaluator. *IEEE Trans. Image Process.*, 24(8):2579–2591, 2015.

[153] Nan Zhu, Xinbo Gao, and Cheng Deng. Image scaling factor estimation based on normalized energy density and learning to rank. *IEEE Int'l. Conf. Sec., Pattern Anal., and Cybern.*, pages 197–202, 2014.