How to Assess the Quality of Compressed Surveillance Videos using Face Recognition

Wen Heng, Tingting Jiang, Member, IEEE, and Wen Gao, Fellow, IEEE

Abstract—Video surveillance plays an important role in public security. To store the growing volume of surveillance videos, video compression is beneficial for reducing video volume; however, it is simultaneously harmful to the video quality. Video quality assessment (VQA) methods help to achieve a tradeoff between the data volume and perceptual quality of compressed surveillance videos. Generally speaking, surveillance video quality assessment (SVQA) is different from conventional VQA because surveillance videos are usually used for specific tasks, e.g., pedestrian recognition, rather than for entertainment purposes. Therefore, in this work, we propose two full-reference SVQA methods based on the concept of Quality of Recognition (QoR). We first design two new tasks, distorted face verification (DFV) and distorted face identification (DFI), based on which we further propose two SVQA methods, DFV-SVQA and DFI-SVQA, and corresponding quality metrics. The core components of the DFV-SVQA and DFI-SVQA methods are feature extractors (a DFV model and a DFI model), which we construct using convolutional-neural-network-based face recognition models. In addition, we construct a real-world surveillance video dataset, based on which we analyze how various factors, including the video codec, compression level, face resolution and light intensity, affect the quality of compressed surveillance videos. We find that compared with conventional VQA methods, our methods are more effective in measuring the quality of surveillance videos while maintaining an acceptable time efficiency.

Index Terms—surveillance videos, quality assessment, deep learning, face recognition.

I. INTRODUCTION

In recent years, public security problems have raised widespread concern. Surveillance video systems play an important role in public security and are widely used for monitoring public incidents. For comprehensive monitoring, an increasing number of surveillance cameras are being deployed in public areas, e.g., airports, train stations and urban roads, and higher resolution cameras, e.g., full HD and 4K cameras, are gradually beginning to be used in real scenarios. However, with this situation comes the attendant problem of how to store the enormous amounts of video data being generated every day. Video compression technology can greatly reduce the volume of video data but will inevitably cause distortions in the compressed videos. Therefore, proper video quality assessment (VQA) methods are key for achieving a tradeoff between the data volume and perceptual quality of compressed surveillance videos.

W. Heng, T. Jiang and W. Gao are with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: wenheng@pku.edu.cn; tijiang@pku.edu.cn; wgao@pku.edu.cn).

Copyright 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

In this work, we explore how to effectively assess the quality of compressed surveillance videos. Strictly speaking, surveillance video quality assessment (SVQA) is similar but not identical to conventional VQA. Most conventional VQA methods, e.g., SSIM [1], PVM [2], VMAF [3], and the methods presented in [4]–[6], measure video quality from the perspective of the human perception of signal fidelity. This means that the measured quality is expected to be consistent with users’ perceptions of video distortions with respect to the reference videos. This type of measured quality is known as the quality of experience (QoE) [7]. By contrast, SVQA methods should measure video quality by whether it is favorable for helping humans to perform specific tasks, e.g., face recognition. At present, enormous numbers of surveillance videos are shot every day, and most of them are viewed by people, such as security officers or policemen. When these people watch surveillance videos, they are performing certain tasks, such as face recognition. While performing these tasks, they make subjective evaluations of the video quality. For example, if they can easily identify a person in a video, the video quality is regarded as good; otherwise, it is poor. Therefore, our goal is to simulate this process and to design objective quality assessment methods for surveillance videos that are consistent with human performance on these tasks. Unfortunately, conventional VQA methods cannot capture how well tasks can be performed using surveillance videos; consequently, the quality of surveillance videos as measured using these methods may not satisfy the real demands of SVQA in practice. Petroviv et al. [8] proposed an objective SVQA method that outperforms PSNR and SSIM, but it still measures the QoE of surveillance videos. As a useful reference, [9] provides guidance for selecting cameras, network infrastructures, and display devices for different situations based on practical experience.

Quality of recognition (QoR) is a research topic that focuses on how to measure video quality based on recognition tasks. The Video Quality Expert Group (VQEG) has created a project named “Quality Assessment for Recognition and Task-based multimedia applications” (QART), which aims to “study the quality of video used for recognition tasks and task-based multimedia applications” in order to drive research on task-driven VQA methods. In addition, several works [10]–[12] related to SVQA have been proposed based on the car license plate recognition (LPR) task. Leszczuk et al. [10] developed critical quality thresholds for videos streamed under constrained networking conditions based on subjective LPR. Furthermore, Janowski et al. [11] extended [10] by incorporating automatic LPR (ALPR) methods and performed a
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2866701, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 1: Illustrations of the conventional face verification and face identification tasks as well as the DFV and DFI tasks. DFV and DFI models are used as feature extractors in the DFV-SVQA and DFI-SVQA methods. (Best viewed in color.)

comparison between subjective and objective LPR methods. Ukhanova et al. [12] used a logistic (logarithmic) function to model the relationship between the recognition rate and the video compression ratio and then applied that model to guide the reasonable compression of surveillance videos. However, the dataset used in their experiments was captured under controlled conditions, and not many factors were considered.

In addition to license plates, human faces are also important sources of semantic information in surveillance videos that can be used as important clues for SVQA. Moreover, face recognition is widely applicable for public security purposes. Therefore, SVQA based on the face recognition task is important for real-world scenarios. Korshunov and Ooi [13], [14] measured the quality of surveillance videos based on several face-related tasks. Their analysis focused on how image compression and scaling in surveillance videos affect the performance of face detection, recognition and tracking algorithms. However, the surveillance video data they used were not captured from real scenes, and not all factors that may affect the quality of surveillance videos were accounted for. In addition, they adopted classic but not state-of-the-art algorithms that cannot achieve high performance on benchmark datasets. For example, in [14], the Viola-Jones [15] and Rowley [16] algorithms were used for face detection, an ODA-based [17] face recognition algorithm was used for face recognition, and the CAMSHIFT [18] algorithm was used for face tracking. These algorithms are classic methods for their respective tasks, but their performances are poor compared with those of state-of-the-art algorithms. The adoption of these outdated objective algorithms probably decreased the reliability of the VQA results.

Unlike in previous works, we propose full-reference SVQA methods from the perspective of face recognition. Pursuing high reliability, we adapt state-of-the-art face recognition models to construct the feature extractors used in our SVQA methods. To ensure real-world applicability, we collect video data from real scenes and consider a larger number of factors that may affect the quality of surveillance videos. Most importantly, we design two new tasks for compressed SVQA based on conventional face recognition that can truly reflect the quality degradation of compressed surveillance videos.

Conventional face recognition includes two tasks: face verification [19] and face identification [20]. Face verification is the task of verifying whether a pair of face images show the same person. Based on this task, we propose the distorted face verification (DFV) task for SVQA. The difference is that we directly compare a compressed face image to its corresponding reference (undistorted) face image and judge whether the face in the compressed image can be still recognized as the same face shown in the reference image. It is obvious that the compressed face will become more unrecognizable as the compression level increases. This task reveals how video compression affects the identifiability of faces in surveillance videos, which is important for SVQA. The DFV task is illustrated in Fig. 1. By contrast, face identification is quite a different task, in which the aim is to find the identity of a given face from among the images in a gallery. Based on this task, we propose the distorted face identification (DFI) task. We construct the gallery from the reference video and use the distorted faces from the compressed video as the test data, as illustrated in Fig. 1. The conventional face identification task can still be performed on the reference video; the corresponding recognition rate is denoted by $R_{ref}$. $R_{dis}$ denotes the recognition rate corresponding to the newly defined DFI task performed on the compressed video. For SVQA, we are concerned with the difference between $R_{ref}$ and $R_{dis}$. It is expected that the corresponding drop in the recognition rate should reflect the quality of the compressed surveillance video. As the compression level increases, the drop in the recognition rate should correspondingly increase.

Based on the DFV and DFI tasks, we propose two SVQA methods, called DFV-SVQA and DFI-SVQA, respectively. As shown in experiments, these two methods are suitable for different practical applications involving surveillance videos.

In addition, we construct a real-world surveillance video dataset, based on which we further construct a face dataset for training and selecting the models to be used in our methods. Moreover, we consider four factors that can affect the quality of surveillance videos, including the video codec, compression level, face resolution and light intensity, and we experimentally analyze how they affect the quality of compressed surveillance videos. We adopt two state-of-the-art face recognition models to build our DFV and DFI models, as shown in Fig. 1. Experimental results show that both methods achieve high accuracy on the proposed tasks, thereby demonstrating the high reliability of the DFV-SVQA and DFI-SVQA methods.

The remainder of the paper is organized as follows. In Section II, we present the preliminaries, including the construction of the dataset and the building of the DFV and DFI models. In Sections III and IV, we describe the DFV-SVQA and DFI-SVQA methods in detail. In Section V, we report experiments conducted using the proposed methods and analyze the effects of the various quality-influencing factors. In Section VI, we compare the proposed methods with conventional VQA methods in terms of performance and time efficiency, and we discuss the different application scenarios of DFV-SVQA and DFI-SVQA. We conclude the paper in...
Section VII. To clarify the overall organization of the paper, we present a block diagram in Fig. 2.

II. PRELIMINARIES

In this section, we present the preliminaries for the proposed SVQA methods. This section is divided into two parts: the construction of the real-world surveillance video dataset and the building of the DFV and DFI models.

In the model building part, we introduce how the DFV and DFI models, as shown in Fig. 1, were designed based on existing face recognition methods.

A. Dataset Construction

First, we describe the construction of the real-world surveillance video dataset. Based on this video dataset, we constructed a set of face images to be used for the training (fine-tuning) of the DFV and DFI models. The steps of the dataset construction pipeline were source video collection, video compression, face detection and light intensity estimation.

1) Source Video Collection: We collected 6 source surveillance video sequences from real-world surveillance cameras in Beijing, which capture real scenes at major road intersections or on pavement. We refer to these source videos as reference videos (RVs). The RVs were captured at four different times of day with different lighting conditions: midday, afternoon, sunset and night. The daylight videos include front and back lighting conditions. At night, the street lamps are turned on. The RVs record pedestrians walking toward or away from the cameras, where the cameras were set at different distances from the pedestrians, allowing them to capture faces at different resolutions. In addition, all RVs were directly recorded from the surveillance cameras without any postprocessing. All videos are in YUV420 format and have a resolution of 1920 × 1080 pixels with a frame rate of 25 fps. The lengths of the RVs range from 3650 to 5530 frames. The information on the different RVs is summarized in TABLE I.

2) Video Compression: We explored how video compression affects video quality from two perspectives: video codec selection and video compression level. To investigate the effect of the video codec, we selected three representative codecs, namely, H264, H265 and AVS2 [21], which are widely used for surveillance video compression. We used the real-time versions\(^1\) of these codecs in our experiments. To investigate the effect of the compression level, we generated distorted videos (DVs) from all 6 RVs with each selected video codec at a series of compression levels. To align the compression levels across the different video codecs, we predefined 17 bit rate levels for each RV and then compressed each RV with each of the three video codecs based on these bit rates. Thus, the bit rate deviations among the three compressed videos at each level were kept within a limited range of ±5%. Finally, we obtained 306 DVs (6 RVs × 3 codecs × 17 levels). Thus, the constructed surveillance video dataset contains 6 RVs and 306 DVs in total.

3) Face Detection: Face detection is a key step that must be performed before face recognition. We conducted manual face detection on the Baidu Crowdsourcing Platform\(^2\), which is similar to Amazon Mechanical Turk. Manual face detection can supply accurate information about face locations, sizes and identity clustering, which is beneficial for model training. We considered only frontal face images shown in the videos and ignored profile images. This is because most face recognition methods perform extremely poorly on profile images. Frontal faces are defined as faces with angular deviations of no more than 45 degrees to the left, to the right, and in the upward and downward directions. We detected 16452 frontal faces in total, with window sizes ranging from 30 to 104 pixels. From among these detected faces, we selected 1340 representative detected faces with a resolution of 32 × 32, 1282 faces with a resolution of 48 × 48, and 1164 faces with a resolution of 64 × 64.

Based on the detection results obtained on the RVs, we

\(^1\)The H264 and H265 codecs were obtained from FFmpeg (http://ffmpeg.org/).
\(^2\)http://zhongbao.baidu.com/
cropped the corresponding compressed faces from the DVs. For convenience, we define the concept of a face group (FG), where each FG consists of one reference face from an RV and its 17 corresponding compressed faces from the DVs for each codec. Thus, each reference face corresponds to three FGs, each compressed with a different codec (H264, H265 or AVS2). We refer to the three FGs that correspond to the same reference face as related FGs. In addition, if the reference face for an FG was collected from RVx ($x = 1, \ldots, 6$), we say that this FG belongs to RVx for simplicity.

For DFI, it is first necessary to know the identity of each cropped face. We manually clustered all of the faces into different identities. The numbers of FGs and identities (IDs) belonging to the different RVs are shown in TABLE II.

### TABLE II: Numbers of FGs and identities belonging to different RVs. FG-oe denotes outlier-eliminated FGs, which are defined in Sec. III-B. ID denotes identity.

<table>
<thead>
<tr>
<th>RVx</th>
<th>FG</th>
<th>FG-oe</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>RV1</td>
<td>611</td>
<td>478</td>
<td>65</td>
</tr>
<tr>
<td>RV2</td>
<td>176</td>
<td>161</td>
<td>25</td>
</tr>
<tr>
<td>RV3</td>
<td>1373</td>
<td>1035</td>
<td>78</td>
</tr>
<tr>
<td>RV4</td>
<td>1307</td>
<td>1137</td>
<td>78</td>
</tr>
<tr>
<td>RV5</td>
<td>118</td>
<td>77</td>
<td>40</td>
</tr>
<tr>
<td>RV6</td>
<td>201</td>
<td>142</td>
<td>21</td>
</tr>
</tbody>
</table>

Table: Detailed information on the reference videos.

<table>
<thead>
<tr>
<th>Video ID</th>
<th>Frame No.</th>
<th>Resolution</th>
<th>Frame Rate</th>
<th>Time of Day</th>
<th>Lighting Conditions</th>
<th>Scene</th>
<th>Light Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RV1</td>
<td>5530</td>
<td>1080p</td>
<td>25 fps</td>
<td>Afternoon</td>
<td>Back Lighting</td>
<td>Pavement</td>
<td>0.381</td>
</tr>
<tr>
<td>RV2</td>
<td>4094</td>
<td>1080p</td>
<td>25 fps</td>
<td>Sunset</td>
<td>Diffuse Lighting</td>
<td>Pavement</td>
<td>0.490</td>
</tr>
<tr>
<td>RV3</td>
<td>4768</td>
<td>1080p</td>
<td>25 fps</td>
<td>Sunset</td>
<td>Diffuse Lighting</td>
<td>Road Intersection</td>
<td>0.513</td>
</tr>
<tr>
<td>RV4</td>
<td>4396</td>
<td>1080p</td>
<td>25 fps</td>
<td>Sunset</td>
<td>Diffuse Lighting</td>
<td>Road Intersection</td>
<td>0.525</td>
</tr>
<tr>
<td>RV5</td>
<td>3650</td>
<td>1080p</td>
<td>25 fps</td>
<td>Night</td>
<td>Lamp Lighting</td>
<td>Pavement</td>
<td>0.620</td>
</tr>
<tr>
<td>RV6</td>
<td>4291</td>
<td>1080p</td>
<td>25 fps</td>
<td>Midday</td>
<td>Front Lighting</td>
<td>Road Intersection</td>
<td>0.642</td>
</tr>
</tbody>
</table>

In summary, we constructed a real-world surveillance video dataset as described above. We conducted manual face detection on this video dataset to construct a face image dataset, called the MFD dataset. The MFD dataset was used only to train the DFV and DFI models to be used in our proposed methods. To improve the practical applicability of the proposed SVQA methods, we will introduce an automatic procedure for constructing a face image set in Sec. V-A; this procedure can be used to enable practical SVQA testing without human intervention.

### B. DFV and DFI Model Building

Generally speaking, face recognition methods include two steps: feature extraction and similarity computation.

DFV and DFI models are used as the feature extractors in the DFV-SVQA and DFI-SVQA methods proposed in this paper, as shown in Fig. 1. In this section, we present an overall description of how these DFV and DFI models (feature extractors) are built.

Traditional face recognition methods, e.g., EigenFace [23], FV-Face [24] and High-Dim Face [25], commonly involve the extraction of handcrafted features and the training of a non-linear classifier. These methods are time-consuming and cannot achieve very good performance. Recently, deep learning methods, especially convolutional neural networks (CNNs), have enabled great advances in face recognition. Many CNN-based methods, such as DeepID3 [26], Facenet [27], BaiduFace [28], Center-Loss Face [29] and SphereFace [30], outperform human on the LFW [31] benchmark. CNNs can automatically learn to extract features for face recognition that are more discriminative and powerful than handcrafted features.

Since the purposes of the DFV and DFI models are to extract discriminative features for the DFV and DFI tasks, respectively, we can adapt state-of-the-art face recognition models to build such models. Center-Loss Face (CLF) [29] and SphereFace (SF) [30] are two state-of-the-art face recognition methods. Both CLF and SF are implemented based on CNNs.
The pretrained models\(^3\) have been made publicly available, and both achieve an accuracy of over 99% on the LFW benchmark with the CAISA-WebFace [32] database as the only training data.

We consider two approaches to building DFV and DFI models: a fine-tuning approach and a non-fine-tuning approach. In the non-fine-tuning approach, a pretrained model that has been trained on a conventional face image dataset is used directly as the feature extractor. In the fine-tuning approach, the MFD dataset is used to further fine-tune such a pretrained model. There are two reasons for using the fine-tuning approach: (1) The number of images in the MFD dataset is not sufficient to train a CNN model from scratch. Fine-tuning a pretrained model is a better choice. (2) Conventional face images are quite different from the faces in surveillance videos because of the low face resolutions and widely varying lighting conditions that are typically encountered in surveillance videos. Therefore, we must fine-tune a model that has been trained on conventional face images to make it more appropriate for application to surveillance videos.

The two approaches are more concretely described below.

1) **Non-fine-tuning**: The pretrained CLF and SF models are directly used as the feature extractors for DFV-SVQA and DFI-SVQA. Notably, the original input size for the pretrained CLF and SF models is 112 × 96, so we need to resize the face images to 112 × 96 via ‘BICUBIC’ interpolation before feeding them into the feature extractors.

2) **Fine-tuning**: To make the CNNs more appropriate for application to the face images in the MFD dataset, we apply the following modifications to the network architectures: (1) we adjust the input size from 112 × 96 to 64 × 64, (2) we remove some of the pooling layers from the original networks, and (3) we decrease the number of units in the last fully connected layer from 512 to 128. The first two modifications are related to the reduced size of the input images (64 × 64 instead of 112 × 96). The last modification is applied mainly to extract more compact features to reduce the computational cost of the subsequent similarity comparisons. Detailed information on the modified network architectures is shown in TABLE III. Based on the modified networks, we fine-tune the models with the MFD dataset. For the details of the fine-tuning of the DFV and DFI models, please refer to Sec. III-C and IV-B.

### III. DISTORTED-FACE-VERIFICATION-BASED SVQA (DFV-SVQA)

In this section, we first describe how to apply the DFV task for SVQA and then introduce the subjective labeling procedure used to build a dataset for training the DFV models and choosing the similarity threshold. With the labeled dataset obtained in this way, we fine-tune two existing face recognition models, and finally, we chose the best model based on a performance comparison.

#### A. Method Description

The DFV task is simple and intuitive. Given a compressed face image, we compare it against the corresponding reference

\(^3\)Center-Loss Face: https://github.com/iywen/caffe-face  
SphereFace: https://github.com/wylwu/sphereface

The number of images in the MFD dataset is not sufficient to train a CNN model from scratch. Fine-tuning a pretrained model is a better choice. (2) Conventional face images are quite different from the faces in surveillance videos because of the low face resolutions and widely varying lighting conditions that are typically encountered in surveillance videos. Therefore, we must fine-tune a model that has been trained on conventional face images to make it more appropriate for application to surveillance videos.

**TABLE III**: Modified network architectures for model fine-tuning. The first row shows the dimensions of the input layer. The second row shows the dimensions of the convolution and pooling layers, where \([k \times k, m] \times N\) denotes \(N\) cascaded convolution layers with \(m\) filters with dimensions of \(k \times k\), a double-row bracket denotes a cascaded residual block [33], and \(2 \times 2 \downarrow\) denotes a \(2 \times 2\) pooling layer. The third row shows the number of units in the FC layer, and the fourth row describes the loss layer.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Center-Loss Face</th>
<th>SphereFace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Layer</td>
<td>64 × 64</td>
<td>64 × 64</td>
</tr>
<tr>
<td>Convolutions &amp; Pooling Layers</td>
<td>([3 \times 3, 32] \times 1)</td>
<td>([3 \times 3, 64] \times 1)</td>
</tr>
<tr>
<td></td>
<td>([3 \times 2] \times 2)</td>
<td>([3 \times 3, 64] \times 1)</td>
</tr>
<tr>
<td></td>
<td>([3 \times 3, 128] \times 4)</td>
<td>([3 \times 3, 128] \times 4)</td>
</tr>
<tr>
<td></td>
<td>([2 \times 2] \downarrow)</td>
<td>([3 \times 3, 128] \times 4)</td>
</tr>
<tr>
<td></td>
<td>([3 \times 3, 512] \times 6)</td>
<td>([3 \times 3, 512] \times 6)</td>
</tr>
<tr>
<td>FC Layer</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Loss Layer</td>
<td>Softmax + Center Loss</td>
<td>Angular-Softmax [30]</td>
</tr>
</tbody>
</table>

(undistorted) image and judge whether it can still be recognized. The recognition rate decreases as the compression level increases. Two sample face image pairs are shown in Fig. 4.

![Sample face image pairs to illustrate the DFV task [22]](image)

Since the DFV task deals specifically with face images and the final goal of SVQA is to assess the quality of a video sequence, we will show how the quality of a compressed surveillance video can be assessed based on the DFV task. Given a compressed video (DV) and its RV, we first detect faces from the RV and crop the corresponding faces from both the RV and the DV. Then, we extract the features of all face image pairs via the DFV model (feature extractor) and compute the similarity of each pair. If the similarity of a face image pair is higher than a predefined threshold, that pair is assigned a label of 1 (recognized); otherwise, its label is 0 (unrecognized). The threshold is selected such that the trained model achieves the best possible performance on the validation set (for our experiments, the validation set was split out from the training data at a proportion of 20%). Let \(N_r\) denote the total number of face image pairs, and let \(N_u\) denote the number of face image pairs with a label of 1; then, we can define the \(VR\) metric, which can be regarded as the DFV rate, as the quality measure for the DV as follows:

\[
VR = \frac{N_r}{N_u}. 
\]

(1)
The DFV-SVQA pipeline is shown in the bottom subfigure in Fig. 5.

For this pipeline, in order to output the quality score, it’s required that the training of the DFV model (Red arrow in Fig. 5) and selection of the similarity threshold (Blue arrow in Fig. 5). As stated in the introduction, we seek an objective quality assessment method that is consistent with human performance. Therefore, subjective labels are required to obtain these two components. Subjective labels indicate whether constructed face image pairs such as those shown in Fig. 4 can be verified by human. The details of the collection of subjective labels for the MFD dataset will be described in the next subsection. Based on the labeled MFD dataset, we trained our DFV models and chose an appropriate similarity threshold.

B. Subjective Labeling

In subjective labeling, subjects are asked to assign labels to constructed face image pairs indicating whether they can be recognized as the same person. Since the face image pairs in our study were constructed only within a given FG, we took an easier approach; instead of asking subjects to label each face image pair individually, we showed all 18 face images in the same FG at once and asked subjects to label the critical distorted face at which the distorted faces switched from “recognizable” to “unrecognizable” with respect to the reference face.

Fig. 6 shows the interface for subjective labeling, in which the 18 faces in the current FG are arranged in ascending order of their compression levels, with the reference face corresponding to level 0. The subjects were asked to compare each distorted face, from left to right, to the reference face and to label two faces: the rightmost face that could be confidently recognized, whose corresponding compression level was then recorded as \( L_1 \) (Level 1), and the leftmost face that could not be confidently recognized, whose corresponding compression level was then recorded as \( L_2 \) (Level 2). The reason for labeling two critical faces rather than one is that we found it difficult for subjects to select only one critical face with a high confidence in practice. In other words, there is a gray area in which humans have difficulty judging whether a given distorted face is recognizable, and it is difficult for subjects to identify the critical distortion level that separates recognizable from nonrecognizable faces. We recorded the labeled critical levels \( (L_1, L_2) \) for each FG.

We recruited 7 subjects, 5 males and 2 females, with ages ranging from 20 to 25. All subjects had good corrected eyesight and had a research background in image processing. All tests were conducted using the same MATLAB-based interface shown in Fig. 6, and each subject was shown all selected FGs generated with all three video codecs.

After the subjective labeling process, we performed outlier elimination on the labeling results, which is necessary and crucial for subjective experiments. For each FG, we took the following steps to eliminate outliers in our experiments.

1) First, we deleted both the maximum and minimum values of \( L_1 \) and \( L_2 \) among the 7 subjects’ labels. This practice can eliminate outliers to some degree and has been used in a previous study [34].
2) Next, we computed the variances of both \( L_1 \) and \( L_2 \) among the remaining labels.
3) If the sum of the variances of \( L_1 \) and \( L_2 \) was greater than 5, we deleted this FG and its two related FGs, which shared the same reference face but were compressed with the other two video codecs. This was done to guarantee fair comparisons among the three codecs in the subsequent experiments.

Finally, to obtain the final labels, we averaged the remaining 5 subjects’ labels for each FG, rounding down for \( L_1 \) and up for \( L_2 \). The numbers of remaining FGs belonging to the different RVs after outlier elimination are denoted by “FG-oe” in TABLE II.

C. DFV Model Fine-tuning

As described in Sec. II-B, DFV models were built based on the CLF and SF methods using two approaches: non-fine-tuning and fine-tuning. In total, four models were thus obtained: non-fine-tuned Center-Loss Face (CLF-NFT), fine-tuned Center-Loss Face (CLF-FT), non-fine-tuned SphereFace (SF-NFT) and fine-tuned SphereFace (SF-FT). We tested the performances of the different models and selected the best-performing one for use in the DFV-SVQA method.

Here, we describe the fine-tuning of the pretrained models. The training of face recognition models based on deep learning is usually treated as a multiclass classification problem, which means that each identity is treated as one category and different face images belonging to the same identity should be classified into the same category. This practice makes the trained model robust to intraintentity variance and sensitive to interidentity variance. For the DFV task, we defined the categories as follows. For each FG in the MFD dataset, we defined the subset of faces with compression levels no higher than \( L_1 \) as \( S_1 \) and the subset of faces with compression levels no lower than \( L_2 \) as \( S_2 \). Then, we treated \( S_1 \) and \( S_2 \) in each FG as two independent categories. Thus, the total number of categories...
was twice the number of FGs. This practice made the fine-tuned models more robust to the variances within $S_1$ or $S_2$ and more sensitive to the variance between $S_1$ and $S_2$. We split the training data into two sets: 80% for training and 20% for validation. The validation set was used for similarity threshold selection during training.

During testing, it is usually necessary to construct positive pairs and negative pairs for the face verification task. In accordance with the definition of the DFV task, we combined the reference face with each corresponding compressed face with a compression level no higher than $L_1$ in each FG to form positive pairs, and we combined the reference face with each corresponding compressed face with a compression level no lower than $L_2$ in each FG to form negative pairs. Then, based on these face image pairs, we compared the performances of the different models.

We used leave-one-video-out validation to test the generalization ability of the trained models. In each round, we used 5 of the 6 videos as training data and the remaining one as testing data. Then, we averaged the testing performances from all six rounds to obtain the final performance of the fine-tuned DFV model (CLF-FT or SF-FT). For the non-fine-tuned models, we directly tested the performances of the pretrained models (CLF-NFT and SF-NFT) on the 6 videos and took the average to obtain their final performances.

### D. DFV Model Comparison and Selection

Here, we compare the performances of all of the DFV models obtained under different conditions. The metric we adopt is the prediction accuracy, which is defined as follows:

$$P = \frac{N_{\text{correct}}}{N_{\text{test}}}$$  \hspace{1cm} (2)

where $P$ is the prediction accuracy, $N_{\text{correct}}$ denotes the number of correctly verified pairs, and $N_{\text{test}}$ denotes the total number of test pairs. A higher $P$ value indicates that a model’s performance on the DFV task is more consistent with human perception.

The FGs in the MFD dataset were generated with three different video codecs. To explore whether different video codecs affected the performances of the fine-tuned models, we tested two methods of fine-tuning the DFV models: one-codec fine-tuning and multicodec fine-tuning. For all DFV models, we tested their individual performances on the data compressed with each of the three video codecs. For example, we fine-tuned a DFV model based on the data compressed with H264 and tested that model’s performance on not only the data compressed with H264 but also the data compressed with the other two video codecs. This strategy was applied for two reasons. First, it allowed us to compare the quality of videos compressed with different codecs using the same objective model to obtain fair comparison results. Second, it allowed us to see whether a model fine-tuned on one specific video codec could be generalized to other video codecs. We show the test results in TABLE IV. The following observations can be drawn. First, compared with the non-fine-tuned models, the fine-tuned models mostly achieve better performance, which can be explained by their adaptation to the face images found in surveillance videos. Second, all of the one-codec fine-tuned models show similar results, while the multicodec fine-tuned models perform slightly better. A good generalization ability is observed in cross-codec testing for all DFV models. Third, a comparison between the CLF and SF methods clearly reveals that SF slightly outperforms CLF on most, but not all, test cases.

In summary, for the DFV task, the fine-tuned models are better choices than the non-fine-tuned models, and the fine-tuned SF model performs better than the fine-tuned CLF model. Therefore, we selected the multicodec fine-tuned SF model, which is abbreviated as SF-FT(3c) for simplicity, for use in the DFV-SVQA method. We will present further experimental results of the DFV-SVQA method based on the SF-FT(3c) model in Sec. V. It should be noticed that due to the small size of our constructed dataset, overfitting may have occurred for the fine-tuned models. The leave-one-video-out validation strategy can avoid overfitting to some degree, but the variety and size of the dataset are of key importance for generating high-generalization fine-tuned models. Therefore, to increase the model generation capability, a large dataset with high variety is desired. In this paper, models trained with our constructed dataset are considered only as examples. Users can collect additional video data that are suitable for their specific application scenarios with which to train the DFV model (and the same holds true for the DFI model).

### IV. DISTORTED-FACE-IDENTIFICATION-BASED SVQA (DFI-SVQA)

Compared with the face verification task, face identification is more difficult and more similar to real-world application scenarios. Based on the conventional face identification task, we propose the DFI task. In this section, we first describe how to assess the quality of compressed surveillance videos based on the proposed DFI task. We call this method DFI-SVQA. Then, we introduce how we fine-tuned our DFI models and selected the best model based on a performance comparison.
### A. Method Description

The conventional face identification task can be described as follows. First, it is necessary to construct an identity gallery that includes a single standard face image for every identity. Second, a given test face image is compared to each identity in the gallery, and its predicted identity is chosen as the one with the highest similarity, which is also called the rank-1 identity. If the rank-1 identity for the test face image is the same as its ground-truth identity, it is called a rank-1 hit. The DFI task is the same as the conventional face identification task except that the identity gallery is constructed from the RV, whereas the test data are taken from the DV. These two tasks are illustrated at the bottom of Fig. 1.

For SVQA, we are interested in how much semantic information is lost due to video compression. In the full-reference SVQA scenario, if we let \( R_{raf} \) denote the recognition rate of reference faces in the conventional face identification task and \( R_{dis} \) denote the recognition rate of compressed faces in the DFI task, then the drop from \( R_{raf} \) to \( R_{dis} \) reflects the quality of the compressed video (DV). Thus, the proposed DFI-SVQA method is formulated as follows. First, we construct an identity gallery from the RV. Then, we conduct separate face identification tests with the RV and DV. We calculate the rank-1 hit rates for both videos. The \( IR \) (identification rate) metric, which is the quality measure for the DV, is defined as follows:

\[
IR = 1 - \frac{R_{raf} - R_{dis}}{R_{raf}} = \frac{R_{dis}}{R_{raf}}
\]

\( R_{raf} \) and \( R_{dis} \) are the rank-1 hit rates for the RV and DV, respectively, and are calculated as follows:

\[
R_{raf} = \frac{N^{1}_{raf}}{N_{raf}}, \quad R_{dis} = \frac{N^{1}_{dis}}{N_{dis}}
\]

where \( N_{raf} \) and \( N_{dis} \) denote the numbers of test faces in the RV and DV, respectively, with \( N_{raf} \) usually being equal to \( N_{dis} \), and \( N^{1}_{raf} \) and \( N^{1}_{dis} \) denote the numbers of test faces that are rank-1 hits.

\( IR \) is the quality metric proposed based on the DFI-SVQA method, with a higher \( IR \) value indicating a higher video quality. This metric reflects the degree of quality degradation of a compressed surveillance video compared with its RV. If a video is only slightly distorted, its \( IR \) value will be close to 1. In contrast, if a video is severely distorted, its \( IR \) value will be low.

### B. DFI Model Fine-tuning

We constructed four DFI models under different conditions, namely, CLF-NFT, CLF-FT, SF-NFT and SF-FT, which are analogous to the DFV models described in Sec. III-C.

In this section, we mainly describe the fine-tuning of the pretrained models to obtain fine-tuned DFI models. The problem is again treated as a multiclass classification problem, where each identity is treated as one category. In contrast to the fine-tuning of the DFV models, we defined the categories as follows. For each identity in the MFD dataset, we assigned all the corresponding reference (undistorted) faces to one category, without considering the distorted faces. Then, we used the data in all of these categories to fine-tune the pretrained CLF and SF models in order to better adapt them to the typical data found in surveillance videos. The different DFI models (constructed under the four different conditions) are compared in the next subsection.

### Table IV: Prediction accuracies of the different DFV models

<table>
<thead>
<tr>
<th>NFT / FT</th>
<th>Model</th>
<th>Prediction Accuracy (( P ))</th>
<th>Test (H264)</th>
<th>Test (H265)</th>
<th>Test (AVS2)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFT</td>
<td>CLF</td>
<td>0.971</td>
<td>0.980</td>
<td>0.961</td>
<td>0.971</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SF</td>
<td>0.984</td>
<td>0.982</td>
<td>0.973</td>
<td>0.980</td>
<td></td>
</tr>
<tr>
<td>FT (H264)</td>
<td>CLF</td>
<td>0.990</td>
<td>0.971</td>
<td>0.972</td>
<td>0.978</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SF</td>
<td>0.991</td>
<td>0.983</td>
<td>0.984</td>
<td>0.986</td>
<td></td>
</tr>
<tr>
<td>FT (H265)</td>
<td>CLF</td>
<td>0.991</td>
<td>0.977</td>
<td>0.979</td>
<td>0.982</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SF</td>
<td>0.984</td>
<td>0.992</td>
<td>0.976</td>
<td>0.984</td>
<td></td>
</tr>
<tr>
<td>FT (AVS2)</td>
<td>CLF</td>
<td>0.994</td>
<td>0.979</td>
<td>0.980</td>
<td>0.984</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SF</td>
<td>0.987</td>
<td>0.984</td>
<td>0.984</td>
<td>0.985</td>
<td></td>
</tr>
<tr>
<td>FT (3 codecs)</td>
<td>CLF</td>
<td>0.992</td>
<td>0.987</td>
<td>0.982</td>
<td>0.987</td>
<td>0.987</td>
</tr>
<tr>
<td></td>
<td>SF</td>
<td>0.996</td>
<td>0.988</td>
<td>0.987</td>
<td>0.990</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7: Workflow of the DFI-SVQA method.
We again applied leave-one-video-out validation to obtain general performance results. In each round, we treated the identities in 5 of the 6 RVs as the training data and the remaining identities as the test data. We averaged the testing results from the 6 rounds to obtain the final performance of each fine-tuned model (CLF-FT and SF-FT). For the non-fine-tuned models (CLF-NFT and SF-NFT), we directly tested the performances of the pretrained models on the 6 RVs and averaged the results to obtain their final performances.

C. DFI Model Comparison and Selection

TABLE V: Performance results for the different DFI models.

<table>
<thead>
<tr>
<th></th>
<th>CLF-NFT</th>
<th>CLF-FT</th>
<th>SF-NFT</th>
<th>SF-FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>0.917</td>
<td>0.932</td>
<td>0.959</td>
<td>0.937</td>
</tr>
</tbody>
</table>

In this section, we compare the performances of the DFI models trained under different conditions in order to select the best-performing one for SVQA. The metric we adopt is the rank-1 hit rate ($R_1$), which is defined as follows:

$$R_1 = \frac{N_{hit}}{N_{test}}$$

where $N_{test}$ denotes the number of test faces and $N_{hit}$ denotes the number of correctly hit test faces.

TABLE V shows the hit rates of the different models. Regarding the non-fine-tuned models, SF-NFT outperforms CLF-NFT; however, with regard to the fine-tuned models, CLF-FT outperforms CLF-NFT, whereas SF-FT performs worse than SF-NFT. This behavior can be explained by the small number of training samples (2000 ~ 3000 face images) used for model fine-tuning, which caused the fine-tuned model (SF-FT) to overfit the training data and perform poorly on the test data. By comparison, 18 times as many training samples were used to fine-tune the DFV models because of the inclusion of distorted face images. Consequently, overfitting did not occur in the fine-tuned DFV models.

Finally, we chose the SF-NFT model for use in the DFI-SVQA method because it achieves the highest rank-1 hit rate among all four models.

V. EXPERIMENTS WITH DFV-SVQA AND DFI-SVQA

In this section, we report experiments performed with the DFV-SVQA and DFI-SVQA methods. We used the DFV model selected in Sec. III-D and the DFI model selected in Sec. IV-C as the feature extractors in the DFV-SVQA and DFI-SVQA methods, respectively. All experiments reported in this section were conducted using the surveillance video dataset constructed as described in Sec. II-A. The manually constructed face image dataset was used only to fine-tune the DFV and DFI models and was not applied in these tests. Instead, we used an automatic procedure to construct a face image set based on a given set of surveillance videos for testing; this procedure is described in this section.

Although a face image set constructed via a manual procedure (as described in Sec. II-A) can more accurately capture the face information contained in surveillance videos, the manpower required for its construction greatly limits the practical applicability of the proposed methods. The proposed automatic procedure for face image set construction can thus improve the practicality of our proposed SVQA methods and guarantee their applicability in real-world scenarios.

A. Automatic Face Detection Procedure

The automatic procedure for face image set construction based on a test set of surveillance videos consists of four steps: face detection, identity clustering, frontal face filtering and identity clarification.

1. **Face Detection** Here, we conduct face detection with MTCNNs [35]. However, many other face detection methods [27], [36]–[39] could also be used. We attempt to adjust the threshold to avoid false positive detection results. The minimum size of a detected face is set to 32 x 32.

2. **Identity Clustering** Based on the detected faces, we next apply the TC_ODAL [40] algorithm for identity clustering to associate faces belonging to the same identity in adjacent frames. Two types of errors inevitably arise in this step: 1) faces that belong to a single identity may be clustered into 2 or more clusters, or 2) one cluster may include faces belonging to different identities. These errors are referred to as type I errors and type II errors, respectively.

3. **Frontal Face Filtering** Then, we filter out non-frontal faces from the clustered identities. We detect 5 landmark points for each face, as shown in Fig. 8. Then, we measure the distance between the two eyes (denoted by $W$) and the distance between the eyes and the mouth (denoted by $H$). We simply select all faces that satisfy $0.7 < W/H < 1.2$ as frontal faces. These two thresholds are adopted based on the statistics of manually detected faces.

4. **Identity Clarification** In the case of type I errors, faces belonging to the same identity may be clustered into multiple clusters with a small number of faces. Therefore, we filter out identities associated with fewer than 5 faces to address type I errors. In the case of type II errors, a large proportion of the faces in an affected cluster will usually belong to one identity. Therefore, for identities associated with more than 10 faces, we apply the k-means clustering algorithm with $k=2$ to each such identity and retain only the larger cluster as belonging to this identity.

![Fig. 8: Example faces with landmarks. (a) Non-rotated face ($W/H = 1$). (b) Horizontally rotated face with an angle of less than 45 degrees ($W/H = 0.716$). (c) Vertically rotated face with an angle of less than 45 degrees ($W/H = 1.07$). (d) Horizontally rotated face with an angle of greater than 45 degrees, which is filtered out ($W/H = 0.55$).](image-url)
If the number of faces remaining is still larger than 10, we randomly select 10 faces for this identity. This strategy not only mitigates type II errors in the clustered identities obtained in step (2) but also achieves a balance among the numbers of faces associated with different identities.

All steps above are applied to each RV. The detection information obtained from the RV is then used to crop faces from its corresponding DV(s). For the DFI-SVQA method in particular, it is necessary to construct a face image gallery and probe face images. For each tested RV, we construct a face image gallery by randomly selecting one face image associated with each identity, and the remaining faces associated with each identity are used as probe face images. By contrast, for the DFV-SVQA method, it is not necessary to know the identities of the detected faces. Thus, we apply only the above steps (1) and (3) to collect face images from the RV. Compared with that for the DFI-SVQA method, the procedure for constructing the face image set for the DFV-SVQA method is simpler and less time-consuming. The time efficiency of these two methods will be discussed in Sec. VI-C.

B. Experimental Analysis

In this section, we report experiments conducted with the DFV-SVQA and DFI-SVQA methods. All experiments were performed using face image sets automatically constructed from the surveillance video dataset. Our analysis focuses on how the four considered factors, i.e., the video codec, compression level, face resolution and light intensity, affect the quality of the compressed surveillance videos.

1) Codec and Compression Level: The constructed surveillance video dataset contains 6 RVs and their corresponding DVs. The DVs have been compressed with three codecs, namely, H264, H265, and AVS2, at 17 different compression levels for each codec. In this section, we explore how different codecs and compression levels affect the quality of compressed surveillance videos.

Using the DFV-SVQA and DFI-SVQA methods, we first predicted the $VR$ and $IR$ scores for all DVs. Then, for each codec, we averaged the $VR$ and $IR$ scores of the DVs with respect to all 6 RVs at each compression level. Thus, we obtained the average $VR$ and $IR$ scores for each of the 17 compression levels for each codec. These $VR$ and $IR$ scores are plotted with respect to the compression level in Fig. 9. The x axis in Fig. 9 represents the bit rate, where a lower bit rate corresponds to a higher compression level, and the y axes in Fig. 9 (a) and (b) represent the $VR$ and $IR$ scores, respectively.

The $VR$ and $IR$ scores decrease with decreasing bit rate within a certain range ($< 3$ Mb/s) for all three codecs. However, at high bit rates, the $VR$ and $IR$ scores saturate. This shows that compressing a surveillance video with a sufficiently low bit rate will significantly affect the recognizability of faces in the compressed video, whereas a sufficiently high bit rate has only a slight effect. In addition, we compare how different codecs affect the $VR$ and $IR$ scores of compressed videos. We find that at a given compression level, H265 and AVS2 outperform H264, achieving higher $VR$ and $IR$ scores. This means that H265 and AVS2 preserve more face information in the compressed videos, which is beneficial for the face recognition task. Moreover, AVS2 achieves slightly higher $VR$ and $IR$ scores than H265 does, although the difference is not significant. Therefore, in accordance with the above observations, the following priority order for the different codecs is suggested: AVS2 is superior to H265, and H265 is superior to H264, when considered from the QoR perspective.

2) Face Resolution: Next, we explore how different face resolutions affect the quality of compressed surveillance videos. Face resolution is a key factor in the face recognition task, especially for surveillance videos. Because the resolutions of face images in surveillance videos are usually small...
and inhomogeneous, determining how the face resolution affects the quality of surveillance videos will be helpful for guiding the selection of suitable zoom factors or distances for the deployment of surveillance cameras.

For each codec and compression level, we calculated the \( VR \) and \( IR \) scores of the DVs with respect to the 6 RVs at each face resolution. Then, we averaged the \( VR \) and \( IR \) scores across the 6 DVs to obtain the average \( VR \) and \( IR \) scores for each face resolution. As a specific example, we here present our analysis based on the DVs compressed with H264 at a compression level of 2 Mb/s. The average \( VR \) and \( IR \) scores are plotted with respect to the face resolution in Fig. 10. Notably, because of the unbalanced numbers of face images with different resolutions, we applied a bucket-splitting strategy to calculate the \( VR \) and \( IR \) scores. The edge sizes of the face images were mostly within the range of 40–95 pixels. We first split this range into consecutive buckets with a width of 3 pixels. Then, we assigned the face images to the corresponding buckets in accordance with their resolutions. Finally, we calculated the \( VR \) and \( IR \) scores for each bucket.

Fig. 10 (a) shows that the \( VR \) score increases with increasing face resolution. Despite the fluctuations in the \( IR \) score observed in Fig. 10 (b), the \( IR \) score also shows an overall increasing trend as the face resolution increases. Therefore, the results show that we can compress surveillance videos that contain high-resolution faces at a high bit rate while maintaining a good video quality (high \( VR \) or \( IR \) scores).

3) Light Intensity: We estimated the light intensities of the 6 RVs as described in Sec. II-A4 and ranked them from RV1 to RV6 in ascending order of their light intensities. We calculated the \( VR \) and \( IR \) scores of the DVs with respect to the 6 RVs to investigate how the light intensity affects the quality of compressed surveillance videos.

The \( VR \) and \( IR \) scores are plotted with respect to the bit rate for each of the 6 RVs in Fig. 11. There seem to be no strong correlations between the light intensity and the \( VR \) and \( IR \) metrics. In Fig. 11 (a) and (b), the \( BitRate-VR \) and \( BitRate-IR \) curves for RV2 to RV5 intersect with each other and show no clear ordering, indicating that the light intensities of these videos do not significantly affect the \( VR \) or \( IR \) scores of the corresponding DVs. However, both the \( BitRate-VR \) and \( BitRate-IR \) curves of RV1 and RV6 are markedly lower and higher, respectively, than the curves of the other RVs. This may be explained by the extreme light intensities of RV1 and RV6. Thus, it may be concluded that in a moderate range of light intensities, e.g., RV2 to RV5, variations in light intensity do not significantly affect the \( VR \) and \( IR \) metrics. However, extreme light intensities, e.g., RV1 and RV6, do have a significant effect on the \( VR \) and \( IR \) metrics.

In summary, the four considered factors exert different effects on the quality of compressed surveillance videos. These findings help us to understand how surveillance video quality is related to multiple factors in practice. Importantly, the effects of the different factors are reliably reflected by the \( VR \) and \( IR \) metrics. Thus, instead of grappling with multiple confounding factors, we can directly use these metrics in practical applications. Notably, other factors that have not been considered in our experiments, e.g., the head pose, may also affect the quality measures of surveillance videos. We will explore the effects of additional factors in future work.

VI. FURTHER DISCUSSIONS OF DFV-SVQA AND DFI-SVQA

In this section, we first compare the DFV-SVQA and DFI-SVQA methods. Then, we further compare the proposed methods with several conventional VQA methods, e.g., PSNR, SSIM, and PVM. In particular, as a practical consideration, we compare the time efficiencies of various full-reference VQA methods. Finally, we suggest application scenarios for the DFV-SVQA and DFI-SVQA methods to help readers understand their real-world applicability.

A. Comparisons Between DFV-SVQA and DFI-SVQA

We first explore the relationship between the DFV-SVQA and DFI-SVQA methods to help us understand the differences between these two proposed SVQA methods, which imply their suitability for different practical application scenarios. The \( VR-IR \) curves for each of the 6 RVs are plotted for each of the three video codecs in Fig. 12. For most of the RVs, the \( VR-IR \) curves are nonlinear, whereas the curves for RV1 are nearly diagonal. The degree of nonlinearity appears to increase from RV1 to RV6.

Here, we attempt to explain this phenomenon from a mathematical viewpoint. Let us define a function \( f \) with the form \( IR = f(\VR) \) to represent a single curve shown in Fig. 12, and let us define the derivative of \( f \) as \( g(\VR) = \partial f(\VR)/\partial \VR \). Then, we can clarify the relationship between \( VR \) and \( IR \) in terms of the value of the derivative \( g(\VR) \). If \( g(\VR) > 1 \), we know that \( IR \) varies faster than \( VR \); if \( g(\VR) = 1 \), then \( IR \) varies at the same rate as \( VR \); if \( g(\VR) < 1 \), then \( IR \) varies more slowly than \( VR \). For these nonlinear curves, in
Fig. 12: Comparisons between the $VR$ and $IR$ metrics with respect to different RVs for three video codecs. (Best viewed in color.)

Fig. 13: Compare the DFV-SVQA and DFI-SVQA to PSNR, SSIM, PVM and VMAF. We present the comparison results for the DVs compressed with the H264 video codec as an example. (Best viewed in color.)

the range of small $VR$, $IR$ varies faster than $VR$, whereas in the range of large $VR$, $IR$ varies more slowly than $VR$. Since $VR$ monotonically increases as the bit rate increases, as shown in Fig. 9, we can conclude that at low bit rates, $IR$ varies faster than $VR$, and at high bit rates, $IR$ varies more slowly than $VR$.

In addition, it can be observed from Fig. 12 that a stronger light intensity is associated with a more nonlinear curve. Based on the above conclusion, it can be inferred that a stronger light intensity will result in a larger difference between the rates of variation of $IR$ with respect to $VR$ (or the values of $g$) that correspond to low and high bit rates.

B. Comparisons with Conventional VQA Methods

Both of the proposed SVQA methods assess video quality from the QoR perspective. Here, we compare our proposed methods with conventional QoE-based full-reference VQA methods and analyze the differences between these two kinds of VQA methods. We selected four VQA methods for comparison: PSNR, SSIM, PVM and VMAF. PSNR and SSIM are two widely used methods for VQA (and image quality assessment, or IQA), and PVM and VMAF are two methods proposed in recent years. The latter usually offer more accurate quality predictions than PSNR and SSIM but with a lower time efficiency. This low time efficiency may prevent their widespread use in practice. We will compare the time efficiencies of the different methods in Sec. VI-C.

Based on the DVs compressed with H264 (as an example), let us compare the DFV-SVQA and DFI-SVQA methods with the four selected VQA methods. First, we calculated the quality metrics of the different methods, i.e., the $VR$, $IR$, PSNR, SSIM, PVM, and VMAF scores, for the DVs with respect to the 6 RVs at each compression level. Then, we averaged the predicted quality metrics across all 6 DVs to obtain the average quality metrics for each compression level. We then plotted the quality metrics of the two methods to be compared as points in a 2D coordinate space and drew curves through the points corresponding to the different compression levels. The resulting curves are shown in Fig. 13. In each subfigure of Fig. 13, we compare both $VR$ and $IR$, which are represented by red and blue curves, respectively, with a conventional quality metric. In each case, the two curves are not identical (the blue curve is above the red curve); this phenomenon is caused by the intrinsic differences between the DFV-SVQA and DFI-SVQA methods, as analyzed in Sec. VI-A.

In Fig. 13, it is obvious that relative to $VR$ and $IR$, the quality metrics of the four conventional VQA methods saturate in certain ranges. This saturation means that the $VR$ or $IR$ value does not change as the conventional quality metric varies. For example, in Fig. 13 (b), when the SSIM score is lower than 0.6 or higher than 0.9, the corresponding $VR$ and $IR$ values do not change significantly, whereas when the SSIM score is in the range of 0.6 to 0.9, $VR$ and $IR$ vary accordingly. This behavior indicates that only a narrow range of SSIM scores are effective for SVQA, whereas outside that range, the variations in the SSIM score are not useful for assessing the quality of surveillance videos. Specifically, a distorted face with SSIM<0.6 will never be recognized, whereas a distorted face with SSIM>0.9 can always be recognized.

Such saturation also occurs for the other three VQA methods, especially for PSNR and PVM. However, VMAF shows only a narrow saturation range (for VMAF scores lower than 10 or higher than 90); thus, $VR$ and $IR$ vary with the VMAF.
include the following three steps: face detection and alignment.

The time efficiency of a VQA method determines whether it can be widely used in practice. For the task of high-resolution VQA in particular, the time efficiency is especially critical.

We compare our proposed DFV-SVQA and DFI-SVQA methods with the same four full-reference VQA methods considered previously (PSNR, SSIM [41], PVM [2], and VMAF [3]) in Fig. VI. The results for SSIM, PVM and VMAF are based on the source codes released by their authors. PSNR was implemented based on the MATLAB platform. Our methods were implemented based on the MATLAB platform and the Caffe [42] framework. To ensure fair comparisons, we tested the time efficiencies on our constructed surveillance video dataset. These surveillance videos have a resolution of 1080p. In TABLE VI, we list the average time spent per frame for each of the different methods. PSNR and SSIM are both methods with high efficiency. When the input/output (I/O) time is included, the speed of PSNR is almost 20 fps, and the speed of SSIM is approximately 5 fps. The time efficiencies of the VMAF, DFV-SVQA and DFI-SVQA methods are slightly lower, with speeds of approximately 2~5 fps. Compared with the other methods, PVM has the lowest time efficiency, with an average time spent per frame of approximately 16 s. This low time efficiency prevents the PVM method from being used in many practical applications.

Table VI: Time efficiency comparisons of the different VQA methods.

<table>
<thead>
<tr>
<th>VQA Method</th>
<th>Without I/O</th>
<th>With I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>0.0126 s</td>
<td>0.0613 s</td>
</tr>
<tr>
<td>SSIM [1]</td>
<td>0.165 s</td>
<td>0.212 s</td>
</tr>
<tr>
<td>PVM [2]</td>
<td>15.38 s</td>
<td>15.43 s</td>
</tr>
<tr>
<td>VMAF [3]</td>
<td>0.411 s</td>
<td>0.455 s</td>
</tr>
<tr>
<td>DFV-SVQA</td>
<td>0.256 s (0.102+0.138+0.016)</td>
<td>0.309 s (0.155+0.138+0.016)</td>
</tr>
<tr>
<td>DFI-SVQA</td>
<td>0.598 s (0.473+0.081+0.044)</td>
<td>0.651 s (0.526+0.081+0.044)</td>
</tr>
</tbody>
</table>

D. Application Scenarios for DFV-SVQA and DFI-SVQA

In this section, we present some possible application scenarios for the DFV-SVQA and DFI-SVQA methods.
Generally, the DFV-SVQA method is more flexible than the DFI-SVQA method due to the difficulty of the identity clustering step in DFI-SVQA. DFV-SVQA can be applied for online SVQA tasks; in other words, we can integrate this method into surveillance cameras to assess the quality of compressed streaming videos and then adjust the compression level in real time in accordance with the quality score (VR). This is an efficient approach for surveillance video compression since it can help to optimize the compression level in response to changes in important factors in the captured scenes, e.g., the light intensities at different times of day. Although the tested speed of DFV-SVQA (∼4 fps) as reported in TABLE VI is not sufficient for real-time applications, there are several potential methods available to speed up its operation, e.g., sampling frames for SVQA or further optimizing the network architectures of the face detection model and feature extractor.

By contrast, DFI-SVQA would be difficult to use for online SVQA tasks because the identity clustering and gallery construction procedures require the entire captured video sequence. Instead, the DFI-SVQA method is more suitable for offline SVQA tasks. In such a scenario, DFI-SVQA can be used to compare and test different video codecs or to compare the performances of different surveillance cameras in order to help users select the best-performing video codecs and surveillance cameras for their applications. Of course, the DFV-SVQA method can also be used for offline SVQA tasks.

We have presented two possible application scenarios for the DFV-SVQA and DFI-SVQA methods. Users can choose either of these methods depending on their needs.

VII. CONCLUSION

In this paper, we have proposed two full-reference SVQA methods from the QoR perspective. We first proposed two new tasks, namely, the DFV and DFI tasks, based on which we further proposed the DFV-SVQA and DFI-SVQA methods and their corresponding quality metrics, VR and IR. The core components of the DFV-SVQA and DFI-SVQA methods are their feature extractors (the DFV model and the DFI model, respectively). We used the Center-Loss Face and SphereFace methods as the bases for constructing such feature extractors and fine-tuned them with face images extracted from real surveillance videos. To improve the practical applicability of our methods, we also proposed an automatic procedure for constructing a set of face images to allow our methods to automatically assess the quality of surveillance videos without the need for human intervention.

We constructed a real-world surveillance video dataset while considering four quality-influencing factors: the video codec, compression level, face resolution and light intensity. Our experimental results show that all of these factors have an effect on the quality of compressed surveillance videos and that these effects are reliably reflected by the VR and IR metrics. In addition, we compared the proposed methods with QoE-based VQA methods in terms of their performance and time efficiency. Our proposed methods achieve more effective SVQA while maintaining an acceptable time efficiency.

Nevertheless, our proposed SVQA methods have some limitations. First, they are full-reference VQA methods, meaning that they require reference videos. Second, they consider frontal faces only. Third, they cannot currently be used to process high-resolution surveillance videos in real time, and their efficiency depends on the availability of high-performance GPU cards.

ACKNOWLEDGMENT

This work was partially supported by National Basic Research Program of China (973 Program) under contract 2015CB351803 and the Natural Science Foundation of China under contracts 61572042, 61390514, 61527804. We also acknowledge the high-performance computing platform of Peking University for providing computational resources.

REFERENCES


