NO-REFERENCE VIDEO QUALITY ASSESSMENT VIA FEATURE LEARNING

Jingtao Xu⋆,† Peng Ye‡ Yong Liu† David Doermann⋆

⋆Language and Media Processing Laboratory, University of Maryland, College Park, USA
†Beijing Key Laboratory of Network System Architecture and Convergence Beijing University of Posts and Telecommunications, China
‡SONY US Research Center, USA

ABSTRACT
In this paper, we propose a novel “Opinion Free” (OF) No-Reference Video Quality Assessment (NR-VQA) algorithm based on frame-level unsupervised feature learning and hysteresis temporal pooling. The system consists of three components: feature extraction with max-min pooling, frame quality prediction and temporal pooling. Frame level features are first extracted by unsupervised feature learning and used to train a linear Support Vector Regressor (SVR) for predicting quality scores frame by frame. Frame-level quality scores are then combined by temporal pooling to obtain a single video quality score. We tested the proposed method on the LIVE video quality database and experimental results show that without training on human opinion scores the proposed method is comparable to state-of-the-art NR-VQA algorithms.

Index Terms— Video quality assessment, no-reference, feature learning, temporal pooling, human opinion

1. INTRODUCTION

With the popularization of the Fourth-Generation/Long Term Evolution (4G/LTE) wireless networks and the increasing use of digital cameras, smartphones, tablet computers and other mobile devices, there is a tremendous demand for the transmission of high quality images and videos. Guaranteeing the quality of experience (QoE) for end users is a critical problem. Subjective Quality Assessment (QA) methods can provide the most reliable QA results, but they are time-consuming and require huge human cost. As a result, reliable automated objective quality assessment methods are more desirable, especially for video. Based on the availability of reference videos, VQA methods can be divided into three categories, full-reference (FR), reduced-reference (RR) and no-reference (NR). Previous works have mainly focused on FR methods. For example, the industrial standard peak signal to noise ratio (PSNR) and structural similarity index (SSIM) [1] have been widely used. Recently, more complex models have been proposed, including MOVIE [2], STMAD [3] and STRRED [4]. However, the limit of bandwidth confines the transmission of original signal, so FR-VQA methods cannot be used in many practical applications. Therefore NR-VQA is the only feasible solution and perhaps the most important and challenging one.

Most previous NR-VQA methods focus on specific type of distortions, such as H.264/AVC encoding [5, 6] and transmission distortion [7, 8]. These methods require video stream information, distortion type and human opinion scores of videos which restrict their applications. For example, [8] yields a good result but still needs bit rate information from the video stream and human opinion scores for regression. Therefore it is important to study general purpose NR-VQA algorithm which does not have these constraints. However, relatively little effort has been devoted to NR-VQA problems. Konuk et al. [9] proposed a NR-VQA algorithm based on spatial and temporal information of the video. This method used bit rate and packet loss rate as features, which may not be available in many applications. Moreover, it does not provide a complete general purpose solution: four different evaluation functions are used for different types of distortions. Recently Saad et al. [10] proposed a NR-VQA method based on natural video statistics (NVS) in the DCT domain and incorporated the temporal information by using motion coherency which can indicate that the motion masks the visibility of distortion in moving scenes. All the above NR-VQA methods require human opinion scores for training. Mittal [11] presented an OF NR-VQA approach NVIE based on natural video statistics in the spatial domain. Video quality scores are calculated by combining inter subband correlations at different temporal scales.

Motivated by the success of unsupervised feature learning for NR-IQA in CORNIA [12], we propose an OF NR-VQA model – Video CORNIA. In the proposed model, frame-level features are extracted by unsupervised feature learning [12]. A linear SVR is used to learn a mapping from feature space to frame quality score. Then final video quality score is obtained by hysteresis temporal pooling.

Our contributions are three-fold. First, this is the first method that applies unsupervised feature learning in the VQA domain. Previous NR-VQA methods rely on hand-crafted
features based on complex image transformation and filtering techniques and can be very time consuming to compute. Second, a new max-min pooling method is proposed for spatial feature pooling which can decrease the feature memory space compared to the original CORNIA feature. Third, the performance of our NR-VQA model is comparable to state-of-the-art NR-VQA methods on the LIVE video quality database.

The remainder of this paper is organized as follows. Section 2 describes details of Video CORNIA. The experimental setup, results and discussions are presented in Section 3. The paper is concluded in Section 4.

2. THE VIDEO CORNIA MODEL

2.1. Spatial frame feature extraction

CORNIA [12] has achieved state-of-the-art performance using unsupervised feature learning for the NR-IQA problem. We extend it into VQA domain in this paper.

Given a video frame (only intensity component considered), the frame descriptor \(X(i,j)\) at location \((i,j)\) is extracted from \(B \times B\) patches \(I(i,j)\), where \((i,j)\) is sampled on a regular grid over each frame. Contrast normalization [12][14] is applied to each patch so:

\[
X(i,j) = \frac{I(i,j) - \mu}{\sigma + C} \tag{1}
\]

where \(\mu\) and \(\sigma\) is the local mean and standard deviation of each patch, and \(C\) is a constant which can prevent instabilities when the denominator tends to zero. For each frame, \(N\) normalized patches are extracted: \(X = [x_1, x_2, \ldots, x_N] \in R^{d \times N}\) \(d = B \times B\), where each column corresponds to a single patch.

A visual codebook \(D = [d_1, d_2, \ldots, d_K] \in R^{d \times K}\) is constructed by applying k-means clustering on a large number of unlabeled local descriptors [12]. Each column in \(D\) consists of a \(d\)-dimensional codeword which is normalized to unit length.

In the encoding step, we perform soft-assignment encoding using the normalized codebook. In particular, the encoding matrix \(\Omega\) consists of the inner product between each frame local descriptors and each codeword:

\[
\Omega = D^T \times X = \begin{pmatrix}
d_1 \cdot x_1 & d_1 \cdot x_2 & \ldots & d_1 \cdot x_N \\
d_2 \cdot x_1 & d_2 \cdot x_2 & \ldots & d_2 \cdot x_N \\
\vdots & \vdots & \ddots & \vdots \\
d_K \cdot x_1 & d_K \cdot x_2 & \ldots & d_K \cdot x_N
\end{pmatrix} \tag{2}
\]

where each column corresponds to one local descriptor \(x_i\) and each row corresponds to filter responses to one codeword \(d_k\).

The encoding matrix \(\Omega\) is then transformed into a feature vector via feature pooling. A max and min pooling method was used in original CORNIA to obtain the feature but this is not suitable for VQA. This is mainly because one video contains hundreds of frames requiring a large memory space for training. With the purpose of decreasing computation memory space, we utilize a maximum minus minimum pooling to keep both max and min information of the filter responses. For each row of \(\Omega\), the difference between maximal and minimal values are computed. This procedure could speed up the training part for the whole algorithm.

\[
Z(k) = \max(d_k \cdot x_1, \ldots, d_k \cdot x_N) - \min(d_k \cdot x_1, \ldots, d_k \cdot x_N) \tag{3}
\]

A \(K\)-dimension feature vector \([Z(1), \ldots, Z(K)]\) is computed for each frame. It is worth noting that the original CORNIA feature [12] consists of both max and min responses and the feature dimension is \(2K\). The proposed max-min pooling reduced the feature dimension of the CORNIA feature by half and improved the training and prediction speed. It can also capture the levels of various types of distortions well. As is shown in Fig.1, the histogram of filter responses distribution for one particular codeword varies with the levels of distortions, and therefore can be used to discriminate high and low quality frame.

2.2. Frame quality prediction

A linear SVR is used to learn a mapping from the frame spatial feature \(Z\) to quality score. Since standard VQA databases do not contain frame-level quality scores, we use a state-of-the-art FR measure – GMSD [15] to approximate human opinion scores for training. GMSD is very efficient to compute and has high linear correlation with human opinion scores (without applying nonlinear mapping).

2.3. Temporal pooling

Once frame-by-frame quality scores are obtained, we can combine them via temporal pooling to obtain the video quality score. Temporal information is important for VQA. Often times, temporal distortions such as ghosting, jitter and mosquito noise are more annoying than spatial distortions. In this paper, we apply the hysteresis pooling method [16], which is fast, to compute and can achieve better performance compared to simple average or percentile pooling.

Hysteresis pooling is motivated by the hysteresis effect observed in continuous subjective video quality assessment experiments by Seshadrinathan et al. [16]. These subjective tests show that the continuous time frame-level quality scores provided by human subjects followed a smooth trend along the timeline. Furthermore, human subjects’ reaction drops sharply with poor video quality but doesn’t increase immediately with the improvement of quality. Hysteresis pooling can significantly improve existing FR-VQA methods with average pooling. Therefore we use it in our NR-VQA work. Let \(f(t_i)\) represent the time varying scores for each frame at a specific time \(t_i\). The previous \(T\) seconds of information
is considered for memory effect. The maximal frame score (higher score indicates lower quality) among previous $T$ seconds frames is selected as the frame score $a(t_i)$. This reflects the severe distortion of the video give an obvious impact on human experience.

$$\begin{align*}
    a(t_i) &= f(t_i), \\
    a(t_i) &= \max[f(t)], \
    t = \{\max(t_i - T, 1), t_i\}, \
    t_i &> 1
\end{align*}$$

Then the next $T$ seconds quality scores are processed. When the video quality drops down and returns to a high quality level, human perception will not go back sharply. Toward this end we sort the latter $T$ seconds frame scores in descending order and use a Gaussian weighting function to combine them to a single value. Let $s = \{s_1, s_2, \ldots, s_K\}$ be the sorted $T$ frame scores, and $w = \{\omega_1, \omega_2, \ldots, \omega_K\}$ denote the descending half of Gaussian function that all elements sums to 1. The standard deviation of the Gaussian weighting function is set to $(2K - 1)/12$.

$$s = \text{sort}[f(t)], t = \{t_i, \min(t_i + T, T_{video})\}$$

$$b(t_i) = \sum_{i=1}^{K} s_i \cdot \omega_i, i = \{1, 2, \ldots, K\}$$

Here $T_{video}$ is the video length. In order to combine information from previous and latter frames, a simple linear combination is performed, and the final video quality is the average of these smoothed frame quality scores.

$$Q(t_i) = \alpha \times a(t_i) + (1 - \alpha) \times b(t_i)$$

$$Q_{video} = \frac{1}{T_{video}} \sum_{t_i=0}^{T_{video}} Q(t_i)$$

3. EXPERIMENTS

3.1. Experimental Setup

3.1.1. Database

The LIVE VQA database [17] is used to test the proposed model. The LIVE VQA database contains 10 reference videos and 150 distorted videos with 4 different distortions, including compression artifacts for MPEG-2 and H.264, transmission errors induced by IP networks and wireless networks. The frame resolution is $342 \times 768$. All video files have planar YUV 4:2:0 format and six videos contain 250 frames with 10 seconds long, 1 video contains 217 frames with 8.17 seconds long and 3 videos contain 500 frames with 10 seconds long. Each distorted video contains a DMOS represents the subjective quality of video.

3.1.2. Implementation details

The patch size $B$ is fixed to 5 in our experiment, and codebook size $K$ is 10000. For the hysteresis pooling, the previous and latter 2 seconds information is considered so $T$ is set to 2. The weight $\alpha$ is fixed to 0.2 to yield a better result which disclose the memory effect plays a minor role to current frame quality. Actually the results are similar when $\alpha$ is under 0.4.

Since the proposed model requires a training procedure, we divided the LIVE VQA database into content-independent training and testing sets: 80% content for training and rest 20% for testing. Due to 10 different reference videos included, only 45 iterations could be performed for training and testing. For comparison purposes, only 20% videos are tested by FR-VQA algorithms in each possible test to ensure a fair comparison to NR methods. First we evaluate Video CORNIA by applying it on each distortion type. Then we apply it to all four distorted videos. The median of the performance across all these iterations is reported. LIBLINEAR package [18] is utilized to implement SVR. In our experiments, we perform $\epsilon$-SVR with a linear kernel because of the high dimension of feature.
calculate before nonlinear fitting): calculating LCC, a nonlinear fitting processing \[19\] is applied algorithm should have both higher LCC and SROCC. Before evaluate the performance of VQA algorithms. A better VQA and Pearson’s linear correlation coefficient (LCC) are used to Spearman’s rank ordered correlation coefficient (SROCC) 3.1.3. Evaluation

Spearman’s rank ordered correlation coefficient (SROCC) and Pearson’s linear correlation coefficient (LCC) are used to evaluate the performance of VQA algorithms. A better VQA algorithm should have both higher LCC and SROCC. Before calculating LCC, a nonlinear fitting processing \[19\] is applied on the quality scores to map to subjective DMOS (SROCC is calculated before nonlinear fitting):

\[
Q = a_1 \left( \frac{1}{2} - \frac{1}{1 + e^{a_2(Q_{video} - a_3)}} \right) + a_4 Q_{video} + a_5
\]  

where \(a_1, a_2, a_3, a_4\) and \(a_5\) are parameters determined by the nonlinear regression procedure.

3.2. Results and discussion

First, we compared video CORNIA with two Full-Reference measures, SSIM [1] and MOVIE [2], one Reduced-Reference measure, STRRED [4] and two No-Reference measures, V-BLIINDS [10] and NVIE [11]. Experimental results are shown in Table 1 and Table 2. For SROCC, our method is competitive with the state-of-the-art FR-VQA and RR-VQA models on MPEG-2 and H.264 distortion. Video CORNIA outperforms NVIE and is comparable to V-BLIINDS. It is worth noting that our model does not use human opinion scores for training, while V-BLIINDS used human opinion scores. For wireless and IP distortion, performance of all algorithms are poor. This is mainly due to the special characteristics of these packet loss distortions. They may lead to local distortions in video frames which could only affect relatively small regions. As for compression distortions, frame distortions are more uniform than packet loss ones.

Second, we compare the proposed max-min pooling method with the max and min pooling method in CORNIA. In particular, we compute the correlation between each frame-level feature and GMSD for each frame and the highest SROCC values are reported in Table 3. We can see that max-min pooling yields more effective features than the max and min pooling method, while only using a half of the memory space. We test the average running time of training procedure of all four distortions videos for these two kinds of pooling methods. The training time with max-min pooling is 65% shorter than that with max and min pooling.

Third, to demonstrate the effectiveness of the hysteresis pooling, we compare it with the average and the percentile temporal pooling. Percentile pooling emphasizes that the severe spatial distortion give greater influence on human experience. We sort all the frame scores in descending order and average the highest 5% scores as the video quality value. Results are presented in Table 4. With different temporal pooling methods, the overall performance is fairly good and hysteresis pooling yield the best result among three methods. However, for wireless distortion the percentile pooling makes a better performance. It’s mainly because this kind of distortion is usually non-uniform in both spatial and temporal domain which could be better expressed by percentile pooling.

4. CONCLUSION

In this paper, we proposed a human opinion free NR-VQA model which extends the previous NR-IQA algorithm CORNIA into VQA. Frame level features are extracted with max-min pooling method. Then the mapping between feature space and frame quality is learned by using SVR. Finally, with hysteresis temporal pooling, the overall video quality can be obtained. Without using human opinion scores for training, our model correlates well with human opinion scores and is comparable to state-of-the-art NR-VQA algorithms.

---

### Table 1: Median SROCC on 45 times test, 80% of content for training, 20% for testing

<table>
<thead>
<tr>
<th>Method</th>
<th>Wireless</th>
<th>IP</th>
<th>H.264</th>
<th>MPEG-2</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIM</td>
<td>0.679</td>
<td>0.717</td>
<td>0.853</td>
<td>0.810</td>
<td>0.632</td>
</tr>
<tr>
<td>MOVIE</td>
<td>0.910</td>
<td>0.845</td>
<td>0.959</td>
<td>0.925</td>
<td>0.866</td>
</tr>
<tr>
<td>STRRED</td>
<td>0.799</td>
<td>0.805</td>
<td>0.891</td>
<td>0.900</td>
<td>0.799</td>
</tr>
<tr>
<td>V-BLIINDS</td>
<td>0.734</td>
<td>0.689</td>
<td>0.829</td>
<td>0.857</td>
<td>0.752</td>
</tr>
<tr>
<td>NVIE</td>
<td>0.731</td>
<td>0.939</td>
<td>0.940</td>
<td>0.935</td>
<td>0.693</td>
</tr>
<tr>
<td>V-CORNIA</td>
<td>0.674</td>
<td>0.810</td>
<td>0.925</td>
<td>0.934</td>
<td>0.768</td>
</tr>
</tbody>
</table>

### Table 2: Median LCC on 45 times test, 80% of content for training, 20% for testing

<table>
<thead>
<tr>
<th>Method</th>
<th>Wireless</th>
<th>IP</th>
<th>H.264</th>
<th>MPEG-2</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIM</td>
<td>0.714</td>
<td>0.600</td>
<td>0.810</td>
<td>0.786</td>
<td>0.654</td>
</tr>
<tr>
<td>MOVIE</td>
<td>0.800</td>
<td>0.788</td>
<td>0.905</td>
<td>0.929</td>
<td>0.807</td>
</tr>
<tr>
<td>STRRED</td>
<td>0.771</td>
<td>0.771</td>
<td>0.885</td>
<td>0.809</td>
<td>0.826</td>
</tr>
<tr>
<td>V-BLIINDS</td>
<td>0.619</td>
<td>0.657</td>
<td>0.857</td>
<td>0.881</td>
<td>0.737</td>
</tr>
<tr>
<td>NVIE</td>
<td>0.381</td>
<td>0.714</td>
<td>0.881</td>
<td>0.635</td>
<td>0.629</td>
</tr>
<tr>
<td>V-CORNIA</td>
<td>0.595</td>
<td>0.714</td>
<td>0.857</td>
<td>0.929</td>
<td>0.740</td>
</tr>
</tbody>
</table>

### Table 3: Top SROCC between each one of the feature values and GMSD score

<table>
<thead>
<tr>
<th>Pooling method</th>
<th>Wireless</th>
<th>IP</th>
<th>H.264</th>
<th>MPEG-2</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>max-min</td>
<td>0.674</td>
<td>0.714</td>
<td>0.810</td>
<td>0.657</td>
<td>0.762</td>
</tr>
<tr>
<td>SROCC*</td>
<td>0.619</td>
<td>0.714</td>
<td>0.810</td>
<td>0.657</td>
<td>0.654</td>
</tr>
<tr>
<td>LCC*</td>
<td>0.801</td>
<td>0.857</td>
<td>0.855</td>
<td>0.875</td>
<td>0.677</td>
</tr>
<tr>
<td>SROCC(\times)</td>
<td>0.667</td>
<td>0.660</td>
<td>0.667</td>
<td>0.857</td>
<td>0.646</td>
</tr>
<tr>
<td>LCC(\times)</td>
<td>0.814</td>
<td>0.762</td>
<td>0.838</td>
<td>0.894</td>
<td>0.663</td>
</tr>
<tr>
<td>SROCC(\times)</td>
<td>0.595</td>
<td>0.714</td>
<td>0.857</td>
<td>0.929</td>
<td>0.740</td>
</tr>
<tr>
<td>LCC(\times)</td>
<td>0.674</td>
<td>0.810</td>
<td>0.925</td>
<td>0.934</td>
<td>0.768</td>
</tr>
</tbody>
</table>

### Table 4: Performance with different temporal pooling methods. Bold face indicate the best SROCC and LCC. 1 is average pooling, 2 is percentile pooling and 3 is hysteresis pooling

<table>
<thead>
<tr>
<th>Pooling method</th>
<th>Wireless</th>
<th>IP</th>
<th>H.264</th>
<th>MPEG-2</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>max-min</td>
<td>0.674</td>
<td>0.714</td>
<td>0.810</td>
<td>0.657</td>
<td>0.762</td>
</tr>
<tr>
<td>SROCC*</td>
<td>0.619</td>
<td>0.714</td>
<td>0.810</td>
<td>0.657</td>
<td>0.654</td>
</tr>
<tr>
<td>LCC*</td>
<td>0.801</td>
<td>0.857</td>
<td>0.855</td>
<td>0.875</td>
<td>0.677</td>
</tr>
<tr>
<td>SROCC(\times)</td>
<td>0.667</td>
<td>0.660</td>
<td>0.667</td>
<td>0.857</td>
<td>0.646</td>
</tr>
<tr>
<td>LCC(\times)</td>
<td>0.814</td>
<td>0.762</td>
<td>0.838</td>
<td>0.894</td>
<td>0.663</td>
</tr>
<tr>
<td>SROCC(\times)</td>
<td>0.595</td>
<td>0.714</td>
<td>0.857</td>
<td>0.929</td>
<td>0.740</td>
</tr>
<tr>
<td>LCC(\times)</td>
<td>0.674</td>
<td>0.810</td>
<td>0.925</td>
<td>0.934</td>
<td>0.768</td>
</tr>
</tbody>
</table>
5. REFERENCES


