2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) | 979-8-3503-3667-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/WiMob58348.2023.10187827

Deep learning assisted quality ranking for list decoding of videos subject to transmission errors

Alexis Guichemerre† and Stéphane Coulombe†
†*Dept. of Software and IT Engineering*
École de technologie supérieure
Montreal, Canada
alexis.guichemerre.1@ens.etsmtl.ca,
stephane.coulombe@etsmtl.ca

Anthony Trioux*, François-Xavier Coudoux*, Patrick Corlay*
**UPHF, CNRS, Univ. Lille,*
UMR 8520 - IEMN
Valenciennes, France
{anthony.trioux, francois-xavier.coudoux, patrick.corlay}@uphf.fr

***Abstract*—In this paper, we propose a new deep learning-based quality ranking framework to assist video list decoding methods in the context of unreliable video transmissions. The objective is to identify an intact image (corrected video frame) among a list of candidate images generated by a list decoding method, where all candidates, except for the intact image are corrupted. The framework comprises a deep learning-based no-reference image quality assessment (NR-IQA) for non-uniform video distortions (NUD) system to rank the candidate images according to their quality, which allows identifying the best one. To show the validity of our proposed framework, we develop an NR-IQA system relying on a proven patch-based convolutional neural network (CNN) architecture, which we adapt to better account for the non-uniform distortions observed in the candidate images, e.g., H.265 transmission errors during wireless communications. Specifically, we modify the patch size on which our CNN for non-uniform distortions (CNN-NUD) operates to capture a larger and more meaningful spatial context. Moreover, we develop a new training database using images resulting from various bit modifications in the received video packets, to simulate the list decoding process, and train the system using a full reference IQA (FR-IQA) method. Experiments on intra frames of videos encoded using H.265 show the ability of this system to identify an intact image among a set of five candidate images with an average accuracy of 96.6%, whereas traditional NR-IQA metrics or the initially trained CNN system offer poor accuracy ranging between 15.7% and 33.6%, respectively.**

***Index Terms*—Video Quality, Non-uniform Distortions, List Decoding, Convolutional Neural Network (CNN), H.265, transmission errors, wireless communications**

I. Introduction

The last few years have been marked by a growth in the number of intelligent systems. Most of them are equipped with video sensors to allow their interaction with the environment. As the video stream coming from these sensors is usually transmitted to a core computer in order to be processed and analyzed, video compression is inevitable prior to transmission due to the huge amount of data that has to be transmitted. Video communications mainly rely on wireless systems, and as such inevitably suffer from errors in the transmitted bitstream. Traditional approaches to fix such errors consist in using forward error correction (FEC) codes [1]–[4], error correction [5]–[8] or error concealment [9]–[12].

Although challenging, error correction avoids the overhead introduced by FEC, and unlike error concealment, can recover the intact (originally transmitted) information. Video list decoding is an error correction approach in which several altered versions of a received video packet are generated by flipping specific bits [5]–[8]. Each represents a meaningful attempt to correct the received packet, and is known as a *candidate*. Among the generated candidates, only one corresponds to the intact (corrected) packet. The list decoding method must then select, using a ranking criterion, the best candidate, ideally corresponding to the intact video packet. For instance, in [5] and [6], the authors use soft bits, which are real values representative of the probability of having received an actual 0 or 1 (e.g., log-likelihood ratios), to select the most likely video slice candidate. More precisely, they sum the absolute soft bit information of the inverted bits and the most likely video slice candidate is the one with the lowest sum (0 if no bits were inverted). One problem with this approach is that soft bit values are rarely available above the physical layer of the protocol stack.

In order to increase the error correction capability without relying on soft bit information, recent works have proposed to use the Cyclic Redundancy Check (CRC) to guide correction. In this context, the CRC is not considered as an overhead since it is ubiquitously present in communications for validating packet integrity, and is not simply added to support error correction. Two methods have recently been proposed for packet correction. The first is an arithmetic approach [13], [14], while the other is based on a lookup table [15]. As demonstrated in [16], such methods offer great potential in Bluetooth Low Energy (BLE) and the Internet of Things (IoT) environments. However, one drawback with this approach is that it may generate a large number of candidates as N, the number of assumed errors in each packet, grows. To tackle this problem, the authors propose using additional validation steps such as checksum and decoding validations. While this allows to drastically reduce the number of possible candidates, several candidates may nonetheless still remain. For instance, in [16] the authors present experimental results where H.264 Baseline encoded packets restricted to 1500 bytes protected with CRC-24 lead to 34,705 valid candidates when a maximum of N=3 errors is considered. This number is reduced to an average of 117.1 when checksum validation (CV) at the UDP level

is applied and to 3.4 when additional video decoding checks (VDC) are applied (e.g the video decoder can reconstruct without crashing or encountering invalid information). When several candidates remain after CV+VDC, the recovered candidate is arbitrarily chosen as the first decodable one [13]. This could lead to an improper decision, *i.e.,* a video containing visual artifacts. To circumvent this, the decision regarding the right candidate may be made by focusing on the visual aspect of each candidate in the list. Such a visual consideration is presented in [17], where the most likely candidate is decided using a pixel-domain alignment metric.

This idea can be extended by using reliable visual quality metrics (VQMs) for evaluations. Such metrics are divided into three categories: full reference image quality assessment (FR-IQA), reduced reference IQA (RR-IQA) and no-reference IQA (NR-IQA) metrics. The FR-IQA metrics such as the Structural Similarity Index (SSIM) [18], the Peak Signal-to-Noise Ratio (PSNR), the Visual Information Fidelity (VIF), or even the recent Video Multimethod Assessment Fusion (VMAF) metric proposed by Netflix, are not usable in practice, since it is necessary to have access to the original image to compute them.

To overcome this problem with the FR-IQA metrics, NR-IQA metrics such as the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [19], the Perception based Image Quality Evaluator (PIQE) [20], the Naturalness Image Quality Evaluator (NIQE) [21] and recent learning based methods can be used [22]–[24]. However, in this paper, we show that these metrics are not reliable for assessing the visual quality of images degraded due to non-uniform distortions resulting from inverted bits within received H.265 video packets. The learning-based metrics need at the very least to be retrained on an image database comprising non-uniform distortions as observed in video damaged during wireless transmissions and candidate images generated by list decoding. And, depending on the learning-based architecture, more extensive modifications may be required. As these popular metrics are not well adapted to the studied problem, in this paper, we propose a new NR-IQA metric relying on a convolutional neural network (CNN) as part of our deep learning-based framework to support video list decoding. The latter is evaluated considering H.265 wireless communications where videos are subject to transmission errors. We focus on intra-coded images due to their importance in video communications [25].

Our contributions are as follows:

- We propose a new deep learning-based *quality ranking* framework to assist video list decoding methods in the context of unreliable video transmissions.
- We created and are making available two new video databases extracted from the *Sports-1M* YouTube dataset comprising 1 million videos at various resolutions [26]. The first one, used as training database, contains intact and corrupted images (with non-uniform distortions) created from 2590 extracted videos. The second one, used as testing database, is similar to the first one but contains intact and corrupted images generated from another set of 113 videos. The code to recreate these two databases is made available.
- We show from simulations that for intra-coded H.265 frames subject to transmission errors, the proposed approach is highly reliable for identifying the image with the highest quality among a list of candidates with an average accuracy of 96.6% while traditional NR-IQA metrics perform poorly with an accuracy ranging between 15.7% and 33.6%.
- We show that the proposed database plays a key role in the good performance obtained since the initial CNN architecture [27] trained on the LIVE/TID2008 database has accuracy of 32.1% while its accuracy reaches 95.2% when trained on the proposed database. Nevertheless, our improved method not only reaches a higher accuracy (96.6%) but also a better visual quality when a wrong decision is made.

Although H.265 [28] has been selected for this work, the same methodology can be applied to other video compression standards such as H.264 and Versatile Video Coding (VVC) [29]. We have selected H.265 because it is among the standards that offer the best compression efficiency at a reasonable computational complexity. We focus on Intra frames since errors on them have a more negative impact on the visual quality than on Inter frames. Their size is also much larger, making them more prone to errors.

The goal of this work is not to propose a new deep learning architecture or to show that the proposed system outperforms other possible architectures, but rather, to demonstrate that the proposed approach and architecture have the capability to solve the problem of identifying the intact (corrected) image among several images exhibiting to non-uniform distortions due to transmission errors (bit inversions).

The paper is organized as follows: the proposed method is described in Section II. Simulation results are presented and analyzed in Section III. Finally, conclusion and future works are given in Section IV.

II. Proposed Visual Quality Ranking Framework

In this section, we present our deep learning-based quality ranking framework to assist video list decoding methods. We introduce a convolutional neural network for non-uniform video distortions (CNN-NUD) to rank images obtained from a list decoding method. Note that other deep learning-based architectures could also be selected for this purpose. As mentioned, our goal is not to show that this specific CNN-NUD system outperforms other possible architectures, but rather, to demonstrate that the proposed approach and architecture have the capability to solve the problem at hand. We also show the system modifications and other considerations related to the training, for example, that are required to solve it.

The proposed framework is illustrated in Fig. 1. The system is first trained with a dataset containing *intact* and *corrupted* decoded images. The dataset is obtained by first encoding several images using H.265. Then, we simulate the list decoding

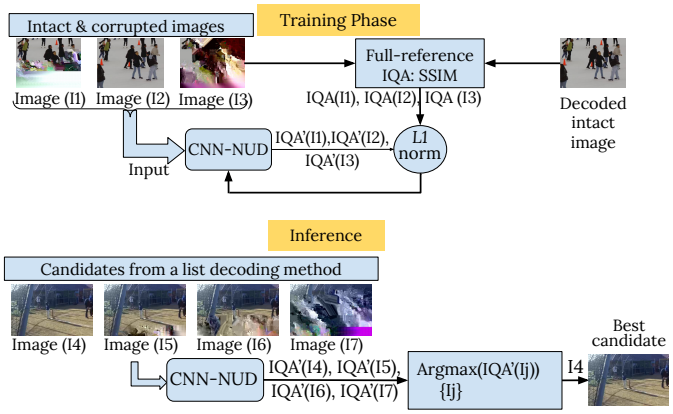


Fig. 1: Global CNN-NUD framework (training on top, inference at the bottom) for the ranking of candidate images generated by a list decoding method, with illustrative examples.

process where bits of each received video packet are inverted at various positions in an attempt to correct transmission errors. Thus, for each encoded video, we obtain a list of video candidates comprising the intact video along with several corrupted versions of it. The loss function, corresponding to the *L1* norm, uses the predicted IQA score, denoted IQA', and a full-reference metric score calculated between each decoded candidate and the decoded intact image. During the inference phase, each candidate image resulting from video list decoding passes through the trained CNN-NUD, and the one with the highest predicted score is selected as the *best* one. We maintain that our system *ranks* images since we are interested in the accurate selection of the intact image rather than having quality scores accurately matching human assessments. Developing such a system is necessary as existing NR-IQA metrics perform poorly in identifying the intact image, as will be apparent in light of our experimental results (see Table II, where their accuracy is less than 42% when there are only four candidates). In the remainder of this section, we present the various aspects of the proposed CNN-NUD.

A. No-reference CNN-based visual quality estimation

The image quality ranking system we develop relies on a proven CNN architecture. We reuse the one presented in [27], which was developed to perform IQA of corrupted images with uniform distortions (e.g., white noise, blur). In this paper, we modify the system according to our problem, and in particular, to support non-uniform video distortions observed in candidates. The system is composed of 5 layers, as illustrated in Fig. 2 and assigns the same visual quality score to each patch of an image.

It takes as input non-overlapping grayscale patches (blocks of pixels) of size 32×32. A comparative study performed by the authors on the size of the patches (ranging from 8 to 48), showed that the performance measured using the Spearman's Rank Order Correlation Coefficient (SROCC) increased slightly, from 0.946 to 0.959, with increasing patch size. These

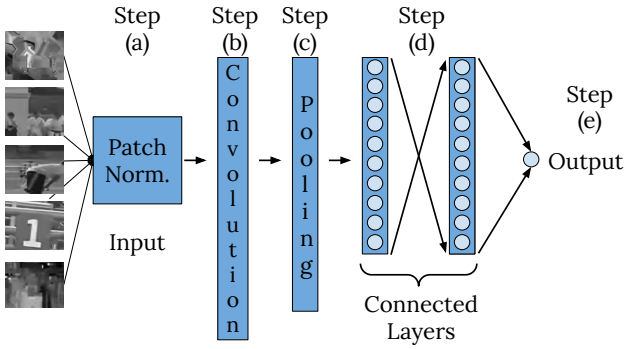


Fig. 2: Architecture of the initial CNN [27].

input patches are normalized as follows (step a):

$$p_{j,m} = \frac{p'_{j,m} - \mu_{j,m}}{\sigma_{j,m} + C}, \ 1 \leq m < K_j, \tag{1}$$

where K_j is the number of patches in an image I_j, $p_{j,m}$ is the normalized intensity of the m-th patch $p'_{j,m}$, $\mu_{j,m}$ its mean, $\sigma_{j,m}$ its standard deviation and C is a constant fixed at 1 to ensure numerical stability even when $\sigma_{j,m}$ is close to zero.

The result is fed to a convolution layer of size 7×7 with 50 filters (step b). Each feature from the convolution is sent to a pooling layer with a max pooling (step c). Its output is fed to two connected layers with 400 neurons each (step d). The last layer consists of a linear regression (step e) to obtain the patch's predicted visual score. The image's visual score is obtained by averaging the one of all patches forming the image.

In [27], the network is initially trained with two databases: LIVE [30] and TID2008 [31]. The loss function used is the $L1$ norm defined as follows:

$$L = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{K_j} \sum_{m=1}^{K_j} \| f(p_{j,m}; \mathbf{w}) - y_{j,m} \|_{L1}, \tag{2}$$

where M corresponds to the number of images used for training the network, $p_{j,m}$ and $y_{j,m}$ correspond to the m-th input patch of image I_j (the j-th training image) and its Difference Mean Opinion Score (DMOS) value, respectively, and $f(p_{j,m}; \mathbf{w})$ represents the predicted patch quality value from the network with weight $\mathbf{w}$. Given the uniform degradation of the images used for training, the value of $y_{j,m}$ is kept constant for each patch of the image and corresponds to the visual score of the whole image (*i.e.,* DMOS). As we will see in the next subsection, this approach, although competitive in terms of best candidate selection accuracy, makes wrong decisions leading to strong image quality degradations in our study case. Therefore, changes are required to consider the non-uniform nature of the distortions observed in our context of interest.

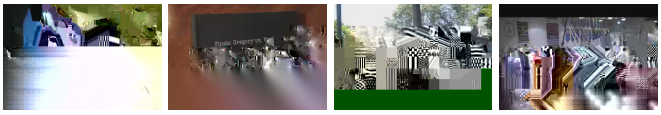
In the inference phase, our system estimates the quality of an image I, comprising K patches, as:

$$\text{IQA'}(I) = \frac{1}{K} \sum_{m=1}^{K} f(p_m; \mathbf{w}), \tag{3}$$

where p_m is the m-th image patch.

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

Fig. 3: Visual illustrations from the LIVE database [30] containing image distortions: (a) JPEG compression, (b) JPEG 2000 compression, (c) fast fading, (d) gaussian blur.



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

Fig. 4: Visual illustrations from the proposed database with non-uniform video distortions: (a)-(d) examples of 4 decoded H.265 video packets containing intra frames where a bit is inverted at different locations.

B. Modification of the CNN to support non-uniform video distortions

As mentioned earlier, the initial NR-IQA architecture is trained with uniform image distortions as shown in Fig. 3a-3d. However, distortions observed in the context of list decoding, where one or several bits of the received video packet are inverted at various positions in an attempt to correct transmission errors, are clearly different and non-uniform, as shown in Fig 4a-4d.

Therefore, the first modification concerns the metric used in the training process. In the initial architecture, a single DMOS value was used for all the patches coming from the same image since distortions applied were uniform within the image. In our case, this simplification does not hold. Indeed, as this work considers NUD, a different score needs to be assigned to intact and visually-corrupted patches during the training. Having access to the original and erroneous images in the training process, we propose assigning to each patch a specific visual quality value by using a FR-IQA metric such as the SSIM. SSIM was selected here due its ability to measure changes in the structures (edges, textures) of images. The metric is calculated between the patch coming from a candidate and the intact one (at the same spatial position).

Furthermore, when dealing with corrupted packets, it can be seen that the reconstructed image might contains some 32×32 erroneous patches exhibiting uniform colors (all black, gray, etc.). The consequence of these distortions is that the convolution operation, whose objective is to extract features, cannot obtain any discriminative feature after the pooling layer. Indeed, patches containing uniform colors can either be original/initial patches coming from the image (e.g., flat areas) or erroneous patches resulting from a bit error. To avoid such ambiguity leading to misclassification, the second modification we propose is to increase the patch size from 32×32 to 64×64. This allows capturing a larger and more meaningful spatial context within which discontinuities introduced by bit errors are more likely to occur. This is shown in Fig. 5.

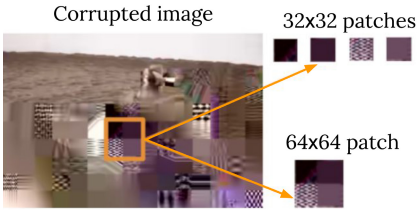


Fig. 5: Example of distortion for a patch size of 32×32 versus 64×64 for an H.265 packet subject to bit alteration.

Using this patch size modification, the convolution layer of size 7×7 is able to capture information resulting from the block boundaries. In order to train our system, we create a new database containing intact and corrupted images from 2590 videos using the SPORT-1M database [32], which consists of 1 million YouTube videos at various resolutions (e.g., 176×144, 320×240, 640×480, 1280×720). Downloaded videos are encoded via the H.265 Test Model (HM) encoder [33], with the following parameters: QP=37, one image per packet. Each corrupted image is a decoded packet after a bit was inverted at a random position to simulate a candidate generated by the list decoding process. The code for generating the database can be found at the following link: https://github.com/AlexisGuichemerreCode/CNN_Ranking_ListDecoding. In a first approach, we consider a high quantization parameter (e.g., QP=37) to train the proposed CNN. For high QP values, a single bit error has the potential to create more significant visual distortion. Indeed, videos encoded at a higher QP contain more critical information such as the prediction modes, while those encoded at a lower QP contain more residual information.

Finally, we modified the learning rate parameter to 10^{-5}. Indeed, the value of 10^{-1}, which was used to train the initial architecture in [27], gave a negative R^2 score when retraining it with our new database. The network was trained using 50 epochs as in [27].

C. Selection of the best candidate

Our neural network allows assigning an image quality value to each patch of the image. The average of the patch quality values within an image I_j is calculated, assigned to the whole image, and is denoted IQA'(I_j). This metric then allows ranking each candidate of a set from the most degraded image to the least. The system selects the optimal image, I^c_{opt}, among P candidates as:

$$I^c_{opt} = \underset{\{I^c_j, 1 \leq j \leq P\}}{\operatorname{argmax}} \text{ IQA'}(I^c_j), \tag{4}$$

where I^c_j is the j-th candidate image.

III. Simulations and Results

In this section, we assess and compare the performance of various NR-IQA metrics, including the proposed system,

in terms of R^2 score for the training phase, accuracy to discriminate the best candidate from the list, and average visual quality for both the selected and the wrongly-classified candidates.

A. Training performance of the CNN-NUD system

We first evaluate the performance of the system during the training phase using the R^2 score [34] computed as:

$$R^2 = 1 - \frac{\sum_{j=1}^{M}\sum_{m=1}^{K_j}(y_{j,m} - \hat{y}_{j,m})}{\sum_{j=1}^{M}\sum_{m=1}^{K_j}(y_{j,m} - \overline{y})}, \quad (5)$$

where $y_{j,m}$ is the SSIM score associated with the m-th patch sample of image I_j (with $1 \leq m \leq K_j$), $\hat{y}_{j,m}$ the SSIM score predicted by the CNN associated with that same patch, and $\overline{y}$ the mean SSIM score of the total database comprising M images. A zero value for R^2 indicates no correlation whereas a score close to 1 indicates a perfect correlation. Note that the training was performed using the 73545 patches comprising the proposed database.

The R^2 score for each tested CNN-based system is shown in Table I. The three systems are: CNN retrained (CNN from [27] trained with our proposed database), the proposed CNN-NUD with 32×32 patches and with 64×64 patches. In the following subsections, we will show that although higher R^2 values would have been hoped for, the systems are still able to identify the best candidate with high accuracy. Even so, the systems greatly differ when we analyze the quality of the candidates they select, especially upon wrong decisions.

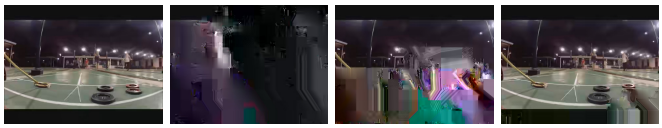
TABLE I: R^2 score for each CNN-based system.

System	R^2
CNN [27] retrained	0.07
Proposed CNN-NUD 32×32	0.25
Proposed CNN-NUD 64×64	0.48

B. Candidate selection performance

In this subsection, we evaluate and analyze the candidate selection performance of various no-reference IQA metrics: BRISQUE, NIQE, PIQE, original CNN proposed in [27] (which is trained with the LIVE [30] and TID2008 [31]), original CNN retrained on our dataset, our proposed CNN-NUD 32×32 and CNN-NUD 64×64 trained on our dataset. As we clearly do not have access to the intact video during the inference phase, only NR-IQA methods can be used (for instance, PSNR and SSIM cannot be used).

Testing database: To study the performance of the proposed system with respect to the studied problem, we create a second database containing 113 videos of various resolutions selected randomly from VGA to 720p. For simplicity, each video is then rescaled at a resolution of 448×360 and encoded using H.265 (HM encoder) with the following parameters: QP=37, 1 packet/frame. A bit is then inverted at a random position in the packet containing the intra (I) frame to simulate the candidate generation process of list decoding. For each video, four possible candidates are generated, namely: the original image without any error (Fig 6a), a candidate with one inverted bit located at about 30% of the packet length (Fig 6b), another one with an inverted bit located at about 40% (Fig 6c) and a last one with an inverted bit at about 70% (Fig 6d) of the packet length. This simulates, for example, the generation of candidates after performing CRC-based error correction from [14]. Each candidate frame is decoded with FFmpeg [35] and without any concealment. The main reasons for creating such candidates are that the number of decodable candidates after the CRC-based error correction with checksum/decoder validation (CV+VDC) is significantly low and inverted bits appear randomly. Note that an inverted bit located at the beginning of the packet normally produces a stronger degradation within the image.



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

Fig. 6: Example of 4 different levels of distortion for an H.265 packet, comprising an intra frame, where a single bit is inverted. Each example corresponds to an inverted bit occurring at a different position, in percentage, within the packet (0% is the first bit and 100% the last): (a) original image, (b) inverted bit at about 30%, (c) inverted bit at about 40%, (d) inverted bit at about 70%.

Accuracy analysis with 4 candidates: Considering this testing database composed of 113 videos and 4 candidates for each video, simulations reveal that the three CNN-based methods (CNN retrained, and the proposed CNN-NUD 32×32 and 64×64) allow correctly selecting the intact video among the other candidates with an average accuracy for different QP of 95.7%, 96.6% and 98.2%, respectively. This is shown in Table II (columns with 4 candidates), which presents a performance comparison between various state-of-the-art no-reference IQA metrics and our CNN-NUD solutions. The accuracy represents the percentage of good classification, *i.e.,* percentage of the time the system can identify the intact video among the candidates. Note that since the original image is clearly not available at the decoder, only no-reference image quality metrics are required to be included in the performance evaluation.

As observed, none of the tested NR-IQA metrics is able to reliably rank the candidates and adequately select the best one from the list. When considering four candidates in the list, the best results obtained for such metrics are very low (*i.e.,* 36.3% at most, obtained for QP=37). The original CNN proposed in [27], which is trained with the LIVE [30] and TID2008 [31], increases the accuracy to up to 42.4%, obtained for QP=37. This poor result is not surprising since the CNN is trained with databases containing visual distortions relating to JPEG 2000 compression, JPEG compression, white gaussian noise, gaussian blur and fast fading (see Fig. 3a-

TABLE II: Performance comparison of various NR metrics for four and five candidates: BRISQUE, NIQE, PIQE, CNN LIVE (CNN from [27] which is trained with the LIVE [30] and TID2008 [31] databases), CNN retrained (CNN from [27] trained with our proposed database) and the proposed CNN-NUD 32×32 and 64×64. *Avg. SSIM choice* represents the average systems' SSIM on the test data while *Avg. SSIM wrong* represents the average SSIM of wrongly-classified images.

	NR-IQA metrics	Accuracy (%)		Avg. SSIM choice		Avg. SSIM wrong	
		4 cand.	5 cand.	4 cand.	5 cand.	4 cand.	5 cand.
QP=22	BRISQUE [19]	15.9	13.3	0.667	0.635	0.604	0.566
	NIQE [21]	15.1	11.5	0.685	0.661	0.629	0.601
	PIQE [20]	34.5	29.2	0.789	0.725	0.678	0.580
	CNN LIVE/TID2008 [27]	26.5	22.1	0.682	0.686	0.567	0.596
	CNN [27] retrained	95.5	95.5	0.975	0.975	0.453	0.453
	Proposed CNN-NUD 32×32	96.4	96.4	0.979	0.979	0.409	0.409
	Proposed CNN-NUD 64×64	**98.2**	**97.3**	**0.993**	**0.993**	**0.622**	**0.735**
QP=27	BRISQUE [19]	20.4	18.6	0.702	0.703	0.626	0.628
	NIQE [21]	15.1	12.4	0.730	0.723	0.682	0.674
	PIQE [20]	38.9	35.4	0.733	0.744	0.563	0.581
	CNN LIVE/TID2008 [27]	34.5	33.6	0.737	0.740	0.599	0.608
	CNN [27] retrained	95.5	94.6	0.982	0.982	0.607	0.671
	Proposed CNN-NUD 32×32	94.6	93.8	0.980	0.980	0.625	0.678
	Proposed CNN-NUD 64×64	**98.2**	**97.3**	**0.996**	**0.997**	**0.813**	**0.873**
QP=32	BRISQUE [19]	28.3	21.2	0.718	0.752	0.606	0.654
	NIQE [21]	18.6	14.2	0.702	0.728	0.634	0.666
	PIQE [20]	38.3	35.4	0.711	0.716	0.531	0.540
	CNN LIVE/TID2008 [27]	33.6	32.7	0.726	0.726	0.588	0.592
	CNN [27] retrained	97.3	96.4	0.988	0.988	0.555	0.662
	Proposed CNN-NUD 32×32	**99.1**	**96.5**	0.992	0.991	0.189	0.757
	Proposed CNN-NUD 64×64	97.3	96.4	**0.992**	**0.992**	**0.730**	**0.789**
QP=37	BRISQUE [19]	35.4	35.4	0.756	0.756	0.623	0.623
	NIQE [21]	24.8	24.8	0.729	0.729	0.639	0.639
	PIQE [20]	36.3	34.5	0.744	0.744	0.622	0.598
	CNN LIVE/TID2008 [27]	42.4	39.8	0.762	0.765	0.580	0.610
	CNN [27] retrained	94.6	94.6	0.981	0.981	0.650	0.655
	Proposed CNN-NUD 32×32	96.4	95.5	0.989	0.993	0.702	0.834
	Proposed CNN-NUD 64×64	**99.1**	**95.5**	**0.999**	**0.999**	**0.996**	**0.980**
Average	BRISQUE [19]	25.0	22.1	-	-	-	-
	NIQE [21]	18.4	15.7	-	-	-	-
	PIQE [20]	37.0	33.6	-	-	-	-
	CNN LIVE/TID2008 [27]	34.3	32.1	-	-	-	-
	CNN [27] retrained	95.7	95.2	-	-	-	-
	Proposed CNN-NUD 32×32	96.6	95.5	-	-	-	-
	Proposed CNN-NUD 64×64	**98.2**	**96.6**	-	-	-	-

3d). Note that the fast fading effects are studied in the case of JPEG 2000 with error resilience features enabled, which induce visual artifacts completely different from the ones resulting from bit errors in an H.265 compressed video stream. When training that CNN with our proposed database, which comprise distortions generated by bit errors such as those observed in wireless transmissions, the accuracy increases to 94.6% at QP=37.

Applying the system changes described in section II allows the proposed CNN-NUD to reach an accuracy of 96.4% and 99.1%, at QP=37, for the 32×32 and 64×64 systems, respectively.

Effectively, by increasing the size of the input parameter to capture more local information, applying a reference metric for each patch rather than for the whole image during the training (*i.e.,* assigning the SSIM locally patch by patch), developing a training database composed of realistically distorted candidates and training the system under these conditions, we succeeded in developing a system capable of reliably selecting the best candidate (*i.e.,* the intact one) from a list issued from a list decoding approach. Even if the retrained CNN performs adequately in terms of accuracy, we are able to further increase it. Indeed the average accuracy, considering all QPs, for the retrained CNN is 95.7%, while the proposed CNN-NUD systems offer an average accuracy of 96.6% and 98.2% for the 32×32 and the 64×64 patch sizes, respectively.

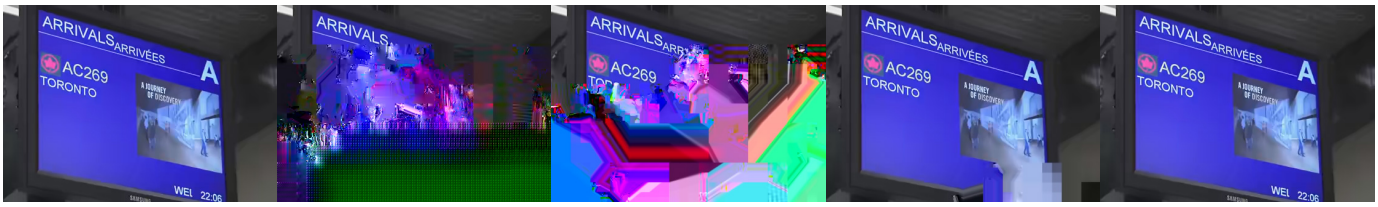
Fig 7 provides a visual example of the candidate selection performed by each method. We can observe that CNN LIVE/TID2008 and CNN retrained make very poor choices. While our CNN-NUD 32×32 makes a better choice, our CNN-NUD 64×64 manages to select the intact version.

Accuracy analysis with 5 candidates: Moreover, as observed in Fig. 6, the test was performed with three candidate videos with major degradation as well as the intact one. When considering an inverted bit located at the end of the packet (position at about 90% of the packet length), a smaller degradation is observed on the intra image as shown in Fig 8. In the following, we also assess the performance of the system by adding a fifth candidate with an inverted bit located at about 90% of the packet length. As observed in Fig. 8, such a location for a wrong bit causes a smaller degradation over the frame (red square) and corresponds to an even more challenging case for the system. In such a case, and as shown in Table II (columns with 5 candidates), the proposed CNN-NUD and the CNN retrained systems still offer great accuracy (95.2% for CNN retrained, 95.5% for 32×32 and 96.6% for 64×64), whereas the other methods still fail in isolating the best candidate (the average accuracy ranges between [15.7%–33.6%] for non CNN-based methods and is 32.1% for CNN LIVE/TID2008). Of course, such challenging error images, in which only a tiny part of the image is distorted, might lead the network to fail to detect them as impairments. Still, in such a case, although the selected candidate is not the best (*i.e.,* the one with no/less distortion), we still have significantly higher SSIM scores than with the other methods based on NR-IQA or on the original CNN.

Finally, in addition to performing tests over challenging situations, we also assessed the performance of the system for different resolutions (*i.e.,* considering larger images), even though the results are not shown due to lack of space. Specifically, 56 videos of size 704×448 were used. The proposed model returned an accuracy of 98.2% for QP=37. Such a good accuracy is expected, since the degradation within the images are similar to those observed at lower resolutions.

Quality analysis: To evaluate the quality of the selected candidates, we compute the SSIM between the intact image and the selected one. In Table II, we show the average SSIM for each metric for four QP values for the case of 4 and 5 candidates. We do not compute the average over all QPs since we do not find it relevant. We can observe that the average SSIM computed over the images selected as best candidate is significantly higher for the proposed systems and the retrained CNN compared to the other methods, for both 4 and 5 candidates. Indeed, regardless of the considered QP, the average SSIMs are no less than 0.975 for the retrained CNN, 0.979 and 0.992 for the proposed system with 32×32 and 64×64 patches, respectively, while for the other methods they do not exceed 0.765, in the best case.

Moreover, when making a wrong decision, the proposed system also outperforms the others, as demonstrated by the higher average SSIM values for wrongly-classified images. Thus, our system not only allows to increase the identification



(a) Intact (b) CNN LIVE/TID2008 [27] (c) CNN [27] retrained (d) Our CNN-NUD 32×32 (e) Our CNN-NUD 64×64

Fig. 7: Example of visual comparisons (QP=32) when a wrong selection has been made for each network except Proposed CNN-NUD 64×64. (a) Intact image. (b)-(e) Selected candidates by: (b) CNN LIVE/TID2008 [27], (c) CNN [27] retrained, (d) Proposed CNN-NUD 32×32, (e) Proposed CNN-NUD 64×64 (which selected the intact candidate).

(a) (b)

Fig. 8: (a) Intact intra image after H.265 decompression (QP=37). (b) A candidate image on which a bit is inverted at the end of the packet (random position at about 90%).

of the best candidate, which increases the average SSIM of selected candidates, but also allows to reduce the distortions due to the selection of a wrong candidate. For instance, for QP=37 and considering 5 candidates, the average SSIM of wrongly-classified images is 0.834 and 0.980 for the proposed CNN-NUD system with 32×32 and 64×64 patches, respectively, compared to mean SSIM values in the interval [0.598–0.639] for non CNN-based methods and of 0.655 for CNN retrained. The superiority of the proposed CNN-NUD system with 64×64 patches is even more evident at other QP values.

The average SSIM of wrongly-classified images is an important metric to compare methods. Indeed, while we wish to increase accuracy, we may never reach 100% and selecting a candidate as close as the intact version becomes essential. Effectively, two methods with a similar accuracy may exhibit very different average SSIMs for wrongly-classified images. It is thus important for a system to select the image with the least distortions when making a wrong decision, especially because any degradation will propagate to the subsequent images in the context of video compression. In practice, the intact candidate may not always be among the available candidates and selecting the one with the best quality is crucial.

Finally, it is important to note that even if the proposed CNN-NUD system with 64×64 patches was trained only with images encoded at QP=37, its performance is nevertheless very good at other QPs. In future work, we will study the impact of training with videos compressed with multiple QP values.

C. Simulation of simplified list decoding process

In this subsection, we simulate a simplified list decoding process with the previous video test database. We suppose that one error is located at a random bit location in each packet due to unreliable transmission and that the list decoding process generates candidates where a single bit is inverted in each. Therefore, the candidates list will include the intact (corrected) video packet along with candidates having two erroneous bits: the original one due to transmission and a wrongly inverted one by list decoding. In the context of multiple errors, it should be easier for the methods to identify the intact image among the other candidates since with two inverted bits instead of one, wrong candidates are likely to experience higher visual distortion.

In Table III, we show the results when 4 candidates are generated. Results of BRISQUE, NIQE and PIQE are not displayed as they perform poorly (performance similar to Table II). We clearly see the important role of our new training database in increasing the accuracy of methods. Also, we observe that the proposed CNN-NUD methods allow to further reduce the number of wrong decisions (from 4.25 to 2.25 on average). Such reduction is of paramount importance as wrong decisions even with relatively good quality will negatively impact the next images due to error propagation.

TABLE III: Accuracy and number of wrong decisions (113 tests) comparison of various NR metrics for four candidates for a simplified list decoding process with one error: CNN LIVE (CNN from [27]), CNN retrained with our proposed database) and the proposed CNN-NUD 32×32 and 64×64.

	NR-IQA metrics	**Accuracy (%)**	**Nb wrong classifications**
QP=22	CNN LIVE/TID2008 [27]	29.0	80
	CNN [27] retrained	97.3	3
	Proposed CNN-NUD 32×32	**98.2**	**2**
	Proposed CNN-NUD 64×64	**98.2**	**2**
QP=27	CNN LIVE/TID2008 [27]	30.4	79
	CNN [27] retrained	93.8	7
	Proposed CNN-NUD 32×32	**97.3**	**3**
	Proposed CNN-NUD 64×64	**97.3**	**3**
QP=32	CNN LIVE/TID2008 [27]	36.5	72
	CNN [27] retrained	98.2	2
	Proposed CNN-NUD 32×32	98.2	2
	Proposed CNN-NUD 64×64	**99.1**	**1**
QP=37	CNN LIVE/TID2008 [27]	36.5	72
	CNN [27] retrained	95.5	5
	Proposed CNN-NUD 32×32	**98.2**	**2**
	Proposed CNN-NUD 64×64	97.3	3
Average	CNN LIVE/TID2008 [27]	33.1	75.8
	CNN [27] retrained	96.2	4.25
	Proposed CNN-NUD 32×32	**97.9**	**2.25**
	Proposed CNN-NUD 64×64	**97.9**	**2.25**

IV. Conclusion and Perspectives

In this paper, we proposed a new deep learning-based quality ranking framework, comprising dedicated training/testing databases, to assist video list decoding methods in the context of unreliable video transmissions. We showed the merits of this framework using a CNN architecture modified to perform visual quality assessment for non-uniform video distortions. We focused on the case of intra-coded H.265 frames subject to bit errors as in wireless transmissions. On a first experiment, the proposed approach was shown to be highly reliable for identifying the image with the highest quality among a list of candidates with an average accuracy of 96.6% while traditional NR-IQA metrics perform poorly with an accuracy ranging between 15.7% and 33.6%. Furthermore, our method, applied to a simplified list decoding approach, reached an average accuracy of 97.9% while the original CNN without retraining reached only 33.1%. For all tested QP values, compared to the original CNN architecture, our CNN-NUD system is more robust in terms of identifying the intact image candidate while limiting the visual distortion when making a wrong decision, even in challenging cases. Future works will include the modification of the system to support inter-coded frames and to account for chroma channels as the observed artifacts are color-dependent. Moreover, we will study the performance of our system with the recent VVC standard [29].

References

[1] H. Tian and Y. Wu, "An application layer forward error correction method of wireless network broadcasting communication for smart video surveillance system," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–9, 2019.

[2] J. Westerlund, "Forward error correction in real-time video streaming applications," Master's thesis, Umeå University, Sweden, 2015.

[3] Y. Huo, C. Hellge, T. Wiegand, and L. Hanzo, "A tutorial and review on inter-layer FEC coded layered video streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1166–1207, 2015.

[4] S. Fonnes, "Reducing packet loss in real-time wireless multicast video streams with forward error correction," Master's thesis, University of Oslo, Norway, 2018.

[5] D. Levine, W. E. Lynch, and T. Le-Ngoc, "Iterative joint source-channel decoding of H.264 compressed video," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1517–1520.

[6] N. Q. Nguyen, W. E. Lynch, and T. Le-Ngoc, "Iterative joint source-channel decoding for H.264 video transmission using virtual checking method at source decoder," in *Proc. IEEE 23rd Can. Conf. Electr. Comput. Eng.*, 2010, pp. 1–4.

[7] F. Golaghazadeh, S. Coulombe, F.-X. Coudoux, and P. Corlay, "Low complexity H.264 list decoder for enhanced quality real-time video over IP," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2017, pp. 1–6.

[8] F. Golaghazadeh, S. Coulombe, F.-X. Coudoux, and P. Corlay, "Checksum-filtered list decoding applied to H.264 and H.265 video error correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1993–2006, 2017.

[9] M. Kazemi, M. Ghanbari, and S. Shirmohammadi, "A review of temporal video error concealment techniques and their suitability for HEVC and VVC," *Multimedia Tools and Appl.*, vol. 80, pp. 1–46, 03 2021.

[10] H. Byongsu, J. Jonghyon, and R. Cholsu, "An improved multi-directional interpolation for spatial error concealment," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2587–2598, 2019.

[11] A. Sankisa, A. Punjabi, and A. K. Katsaggelos, "Video error concealment using deep neural networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 380–384.

[12] B. Chung and C. Yim, "Bi-sequential video error concealment method using adaptive homography-based registration," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 30, no. 6, pp. 1535–1549, 2020.

[13] V. Boussard, F. Golaghazadeh, S. Coulombe, F. X. Coudoux, and P. Corlay, "Robust H.264 video decoding using CRC-based single error correction and non-desynchronizing bits validation," in *IEEE International Conference on Image Processing*, Oct 2020, pp. 1098–1102.

[14] V. Boussard, S. Coulombe, F.-X. Coudoux, and P. Corlay, "Enhanced CRC-based correction of multiple errors with candidate validation," in *Signal Processing: Image Communication*, vol. 99, 2021, p. 116475.

[15] V. Boussard, S. Coulombe, F.-X. Coudoux, and P. Corlay, "CRC-based correction of multiple errors using an optimized lookup table," *IEEE Access*, vol. 10, pp. 23 931–23 947, 2022.

[16] V. Boussard, S. Coulombe, F.-X. Coudoux, P. Corlay, and A. Trioux, "CRC-based multi-error correction of H.265 encoded videos in wireless communications," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, 2021, pp. 1–5.

[17] R. A. Farrugia and C. J. Debono, "Robust decoder-based error control strategy for recovery of H.264/AVC video content," vol. 5, no. 13. *IET*, 2011, pp. 1928–1938.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[19] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *Trans. Img. Proc.*, vol. 21, no. 12, p. 4695–4708, dec 2012.

[20] V. N, P. D, M. C. Bh, S. S. Channappayya, and S. S. Medasani, "Blind image quality evaluation using perception based features," in *2015 Twenty First National Conf. on Communications*, 2015, pp. 1–6.

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

[22] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Metaiqa: Deep meta-learning for no-reference image quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 143–14 152.

[23] S. A. Golestaneh, S. Dadsetan, and K. M. Kitani, "No-reference image quality assessment via transformers, relative ranking, and self-consistency," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1220–1230.

[24] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Generalizable no-reference image quality assessment via deep meta-learning," *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 32, no. 3, pp. 1048–1060, 2022.

[25] W. Tan, B. Shen, A. J. Patti, and G. Cheung, "Temporal propagation analysis for small errors in a single-frame in H.264 video," in *IEEE International Conference on Image Processing, ICIP 2008, October 12-15, 2008, San Diego, California, USA*. IEEE, 2008, pp. 2864–2867.

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[27] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1733–1740, 2014.

[28] ITU-T H.265 and ISO/IEC JTC 1/SC 29/WG 11, "ITU-T recommendation H.265: High Efficiency Video Coding," 2013.

[29] B. Bross, J. Chen, S. Liu, and Y. Wang, "ITU-T and ISO/IEC JVET-S2001 - versatile video coding (draft 10)," 2020.

[30] "Live image quality assessment database." [Online]. Available: http://live.ece.utexas.edu/research/quality/subjective.htm

[31] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008 - a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, pp. 30–45, 01 2009.

[32] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[33] K. Suhring, "H.265/HEVC HM reference software 16.20," 2019, [Available] https://hevc.hhi.fraunhofer.de/.

[34] A. Di Bucchianico, "Coefficient of determination (R^2)," *Encyclopedia of Statistics in Quality and Reliability*, vol. 1, 2008.

[35] FFmpeg codec documentation. [Online]. Available: https://www.ffmpeg.org/ffmpeg-codecs.html\#Video-Decoders