Using CNNs for Quality Assessment of No-Reference and Full-Reference Compressed-Video Frames

1st Renato R. da Silva
School of Computer Science
Federal University of Uberlândia
Uberlândia, Brazil
renato.rsufu@gmail.com

2nd Luiz F. A. Brito
School of Computer Science
Federal University of Uberlândia
Uberlândia, Brazil
luiz.brito@ufu.br

3rd Marcelo K. Albertini
School of Computer Science
Federal University of Uberlândia
Uberlândia, Brazil
albertini@ufu.br

4th Marcelo Z. do Nascimento
School of Computer Science
Federal University of Uberlândia
Uberlândia, Brazil
marcelo.nascimento@ufu.br

5th André R. Backes
School of Computer Science
Federal University of Uberlândia
Uberlândia, Brazil
arbackes@yahoo.com.br

***Abstract*—For videos to be streamed, they have to be coded and sent to users as signals that are decoded back to be reproduced. This coding-decoding process may result in distortion that can bring differences in the quality perception of the content, consequently, influencing user experience. The approach proposed by Bosse et al. [1] suggests an Image Quality Assessment (IQA) method using an automated process. They use image datasets pre-labeled with quality scores to perform a Convolutional Neural Network (CNN) training. Then, based on the CNN models, they are able to perform predictions of image quality using both Full-Reference (FR) and No-Reference (NR) evaluation. In this paper, we explore these methods exposing the CNN quality prediction to images extracted from actual videos. Various quality compression levels were applied to them as well as two different video codecs. We also evaluated how their models perform while predicting human visual perception of quality in scenarios where there is no human pre-evaluation, observing its behavior along with metrics such as SSIM and PSNR. We observe that FR model is able to better infer human perception of quality for compressed videos. Differently, NR model does not show the same behaviour for most of the evaluated videos.**

***Index Terms*—Convolutional Neural Network, Digital Video Streaming, Quality Analysis.**

I. Introduction

The consume of digital contents is increasingly becoming part of our everyday lives, especially regarding digital video. Video streaming is widely transmitted nowadays not only for high definition television, but also for video chats, conferences and internet streaming [1].

For video contents to be transmitted to users, some important processes must be done. First, the video content has to be coded and transformed in signals that will be sent as packages to the final user [2]. Next, this signals are decoded and remounted as video content to be reproduced on the user side. The problem is that this coding-decoding process may result in distortions which may lead to differences in the quality perception. Therefore, the received video, when reproduced to the audience, may not show the exact quality as the original source file [2].

Depending on the context of video reproduction, quality assessment can be a crucial aspect. Usually, the quality of a video is related to the quality of the images or frames that compose it. The human perception of the quality of a visual content can be hard to quantify as it is a subjective matter and may vary from person to person. Thus, being able to assess quality is an important task but, definitely not a trivial one [3]. One of the ways to perform Image Quality Assessment (IQA) is to make a classification depending on the amount of information from the original reference image present in the distorted one [1]. When the access to the full reference image is available, the IQA approach is called Full-Reference (FR), and, when it is not available, the IQA is called No-Reference (NR) approach [1].

Some state-of-the-art techniques evaluate quality of images and videos purely based on human opinion. Basically, various samples of images with different levels and types of compression are shown to human subjects that, based on their visual perception, classify the samples with scores of quality [4].

Recently, other approaches have been proposed to perform IQA using automated methods. For example, in the work of Bosse et al. [1], the authors use datasets of images already classified according to their quality to train a Convolutional Neural Network (CNN). Classification was performed with traditional human review and used as training and validation samples to the CNN. Then, the CNN models were used to perform predictions of Image Quality Assessment.

Convolutional neural networks, in recent years, have shown great relevance among the traditional approaches related to

computer vision. This technique has been widely used due to the quality and amount of data available, and the computing power that has been growing significantly through the years. Furthermore, CNNs allow researchers to provide joint learning of resources and regression based on raw input data with very little manual interference needed [5].

These networks receive labeled samples as inputs. As these samples pass through the network layers by the epochs, features are extracted and the network learns, more generally, which features best represent each label [6]. Different types of layers can be used to build the network structure. Some of the most commonly used are the convolutional layer, the pooling layer and the fully connected layer. The convolutional layer is responsible for applying convolutions using activation filter masks responsible for extracting the features of the image samples. The use of this type of layer is the reason for the name "convolutional neural network". The filters are initially defined in a random way and have their values adjusted gradually at each iteration of the samples in the neural network [6]. The pooling layer is responsible for receiving samples and, based on some parameters, producing smaller samples which occupy less disk space. This fact is important since neural networks usually demand a large amount of input samples. Besides this advantage, this layer is intended to generate more robust features by reducing the sensitivity of the network to distortions. This way, a greater variety of images can be associated with the generated features, thus enhancing the classification [6]. Finally, the fully connected layer is responsible for performing regression and weight adjustments. The samples used as inputs to the neural network are initially divided into training and validation sets. Then, the validation set is compared with the training set in order to identify necessary weight adjustments for next iterations [6].

In their work, Bosse et al. [1] use TID2013 [7] and LIVE [8] datasets of images already classified according to their quality. The quality labels previously defined by human subjects are used as classes to train a CNN using 3000 epochs, 10 convolutional layers, 5 pooling layers, as well as 2 fully connected layers for regression. After the training process, the CNN models are used to perform predictions of Image Quality Assessment. Also, it is worth mentioning that, although the aim of their work was to propose methods for assessment of image quality in video streaming, all images used in the training process were single pictures and not extractions of compressed videos. Thus, we can consider that compression methods covered by the training process only exploit spatial redundancies to reduce the size of pictures. Another approach could also exploit temporal redundancies when considering a sequence of pictures as frames that compose the video. Besides, the tests provided by their original article only states the use of images as tests to the CNN, in order to obtain quality prediction and compare that result to the quality evaluation provided by the human subjects in the dataset.

In this paper, we explore the methods created by Bosse et al. [1] exposing the CNN quality prediction to frames extracted from real compressed videos. Two different video codecs using various quality compression levels were applied to them. We also evaluate how their methods perform while predicting human visual perception of quality in scenarios where there is no human pre-evaluation, observing its behaviour along with metrics such as SSIM and PSNR.

The remainder of this paper is organised as follows. In Section II we describe the process of acquisition, compression and extraction of frames used in the test process, as well as the basic application of these test samples in the CNN using models previously trained. Section III describes the experiments and we report the results achieved by this work. Conclusions and future work are presented in Section IV.

II. Methods

In this section we introduce our methods used to investigate whether the CNN models proposed in [1] infer correctly the perceptual quality of actual compressed videos. First, in Section II-A we present the raw videos used for compression. Then, in Sections II-B and II-C we describe the steps to compress and extract frames from the raw videos. Finally, in Section II-D we detail the process of evaluation using the inference models. A flow chart of all the steps can be observed in Figure 1. Also, the overall configurations of our methods for this paper is presented at Table I.

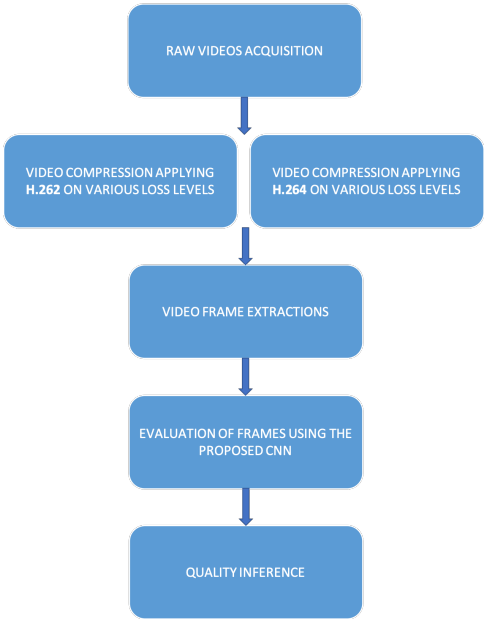


Fig. 1. Flow chart showing the main steps of this project.

A. Video Acquisition

First, we collected videos stored in raw formats that represent diverse scenarios. This diversity is important because different characteristics can introduce different challenges for video codecs and IQA models. For this paper, we obtained four 720p videos with 50 frames per second and no chroma sub sampling. We chose these videos because their quality and sub sampling level allowed us to do a more detailed evaluation.

As they have high quality we could explore more freely more levels of compressions. Also, these videos have no copyright restrictions and are available on https://media.xiph.org/video/derf/ [9].

In Figure 2, we present the preview for the four videos we used to compress and evaluate the CNN models proposed in [1]. Figure 2a, `ducks`, shows a frame extracted from the corresponding video that contains ducks swimming in a river. This scene has slow movements and the colours have low frequencies. The most challenging part for codecs to handle is the wave movements created by the ducks. Figure 2b, `house`, shows the landscape of a house surrounded by vegetation. This scene has elements with low frequency – the house – and high frequency – the vegetation. Trees have borders with irregular shapes, which will probably affect negatively the compression in these regions. Figure 2c, `park`, shows people running with distinct clothing in a park. The main characteristic of this scene is the fast movement of the objects. While camera follows people, trees appear and disappear in the foreground and background, creating a not straightforward time dependency among objects. Finally, Figure 2d, `town`, shows the aerial view of a town. This scene is very detailed and presents very high frequencies, a difficult scenario for video codecs.

B. Video Compression

We compressed the raw videos collected in the previous step using two video codecs provided by the FFmpeg software [10], namely, h.264 [11] and h.265 [12]. These two video codecs reduce the size of raw videos by exploiting spatial and temporal redundancies. First, they convert the image colour space to YCbCr and apply chroma sub sampling to reduce the size of each frame by half without perceptual degradation. This is due to our vision system that do not distinguish subtle changes of colours. Then, they apply prediction techniques to infer whole blocks of pixels by using data of other blocks previously processed. Some techniques predict blocks using only data contained in the same frame, this is called spatial prediction [13]. Frames, such as those containing blue skies, can have well defined patterns and algorithms can use knowledge gathered previously to predict next blocks. Spatial prediction techniques are also employed by image codecs such as JPEG [13] and JPEG2000 [14], which are used in the work of Bosse et al. [1]. Other techniques use data from other frames in order to predict next blocks, this is called temporal prediction [12]. For example, in a movie that contains a ball bouncing on the floor, blocks in the next frame can be predicted by observing the displacement of objects compared to their location in previous frames. Therefore, compression of videos can have different results when compared with compression of single images due to the addition of temporal prediction techniques.

Often, h.264 and h.265 codecs are used for lossy compression but can also be used for lossless compression. After the prediction step, these codecs use the Discrete Cosine Transform (DCT) to obtain coefficients in frequency domain and, then, they apply a quantization matrix to reduce data. The factor of this quantization matrix controls the compression behaviour. If the quantization factor is zero, then, all prediction errors are stored without reduction and, at decoding phase, they are used to reconstruct the video with no loss. If the quantization factor is greater than zero, then, it is a lossy compression and, the higher the quantization factor is, the smaller will be the size and the overall quality of the resulting video. Therefore, lossy compression increases the pixel-to-pixel error and can decrease perceptual quality when the quantization factor is high.

The FFmpeg software provides implementation of many video codecs and can control the bit rate of compressed videos using input parameters. In this paper, we chose h.264 and h.265 codecs because FFmpeg has a uniform parameter, the Constant Rate Factor `CRF`, that controls the compression level of these codecs. The `CRF` parameter for h.264 and h.265 varies from 0 to 51, where 0 means the compression is lossless and 51 means the compression has the highest loss. The default value for `CRF` is 23 and the documentation says that, in order to keep visually lossless quality, one should use `CRF` values near 17 or 18. In this paper, we vary `CRF` from 1 to 51 using h.264 and h.265 for every video described in Section II-A. Below is an example of the command line we used to compress the video `ducks.y4m` to `ducks_h264_1.mp4` using the codec h.264 with `CRF` equals to 1.

```
$ ffmpeg -i ducks.y4m \
  -vcodec libx264 -crf 1 ducks_h264_1.mp4
```

C. Frame Extraction

We used the FFmpeg software to extract 10 frames from each compressed video. First, we queried the duration of the videos using the `ffprobe` command, a program included in the FFmpeg installation. Below is an example of the command line we used to query the duration of the video `ducks_h264_1.mp4`.

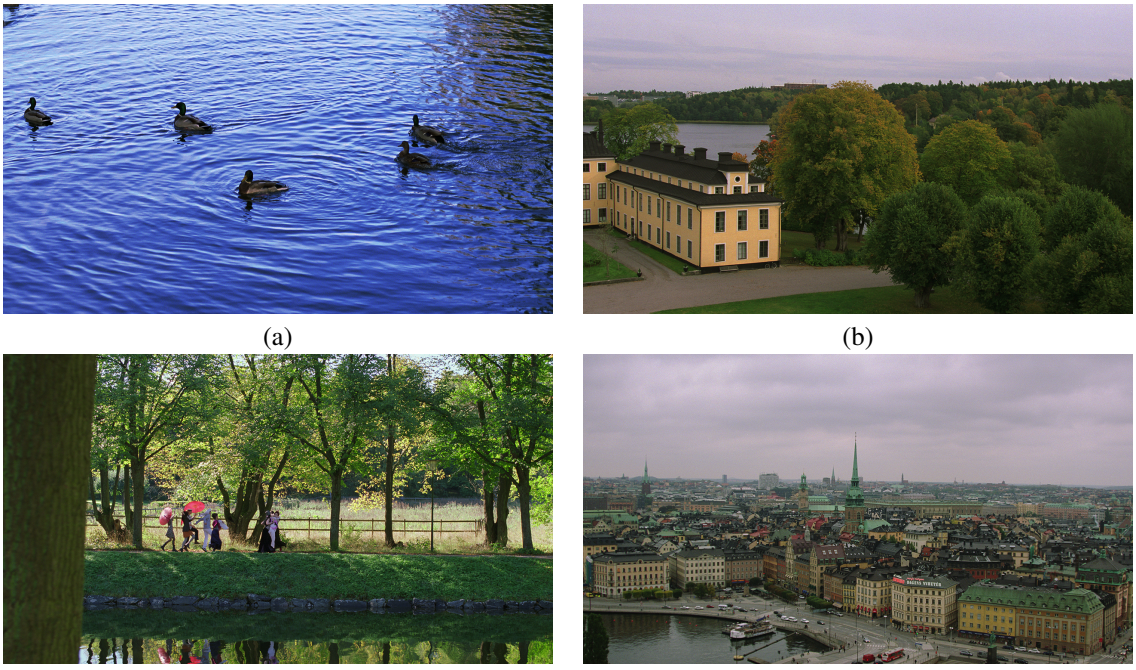
```
$ duration=$(ffprobe -i ducks_h264_1.mp4 \
    -show_entries format=duration \
    -v quiet -of csv="p=0")
```

Then, we generated 10 time stamps at random ranging from 0 to the duration of each video. Finally, we extracted the next frame after the generated time stamps in each compressed video. Note that the time stamps generated for a particular reference video were used to extract frames from all corresponding compressed videos. Below is an example of the command line we used to extract the first frame after the second 5 of the compressed video `ducks_h264_1.mp4` as the image `ducks_h264_1_5.bmp`.

```
$ ffmpeg -i ducks_h264_1.mp4 \
    -ss 5 -vframes 1 ducks_h264_1_5.bmp
```

D. Perceptual Quality Inference

After extraction, we evaluated the compressed video frames using the CNN models proposed by Bosse et al. in [1]. In their



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

Fig. 2. Preview of the videos obtained for compression, frame extraction, and model evaluation.

work, the authors built models from two datasets, TID2013 [7] and LIVE [8], using two different approaches, FR and NR, in which they compare CNNs with two different pooling layers applying standard mean or weighted mean. In this paper, we will explore only the FR and NR models built from TID2013 dataset using weighted mean variant of pooling layer.

The original authors have made their code and models available on https://github.com/dmaniry/deepIQA. In order to evaluate a compressed frame using the FR approach, the reference frame needs to be passed to their program. Differently, for NR approach, only the compressed frame needs to be passed as input. The output of all executions were stored in a Comma Separated Values (CSV) file to run the analysis described in Section III. Below is an example of the command line we used to execute the CNN model `fr_tid_weighted.model` to predict the perceptual quality of the compressed frame `ducks_h264_1_5.bmp` using the reference `ducks_reference_5.bmp` for the FR approach.

```
result=$(python evaluate.py \
--model fr_tid_weighted.model \
--top weighted \
ducks_h264_1_5.bmp ducks_reference_5.bmp)
```

Additionally, we also computed quality measurements of the compressed videos using the ImageMagick software [15]. This software is often used to automate image edition but it also provides the ability to compute quality measurements of compressed images given the reference picture. In this paper, we computed Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) for every compressed frame using the corresponding frame of the reference video. Below is an example of the command line we used to compute the PSNR of the compressed frame `ducks_h264_1_5.bmp` regarding its reference frame `ducks_reference_5.bmp`. The residuals are stored in the output image `residuals.png` and can be ignored.

TABLE I
EVALUATED PARAMETERS

Data	Value
Compression Codecs	H.264 and H.265
Quality Loss levels (CRF)	0, 1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51
Assessment approaches	FR and NR
Videos	4
Frames Extracted per Video	500
CNN Training model	TID2013 weighted
Total of Frame Samples	2000

```
psnr=$(magick compare -metric psnr \
ducks_h264_1_5.bmp ducks_reference_5.bmp \
residuals.png 2>&1)
```

All scripts for data acquisition, video compression, frame extraction and evaluation are public available in our repository on https://bitbucket.org/luizcoro/seminario-multimidia-2019/.

III. RESULTS

In this section, we present and discuss our results. We gathered outputs of the CNN models proposed by Bosse et al. in [1] along with the quality measures PSNR and SSIM. PSNR is based on pixel-to-pixel error of the compressed frames and SSIM is a measure that treats structural components differently in order to achieve a quality closer to our visual perception.

Therefore, the results of the CNN models should present more similarities with SSIM than PSNR.

Instead of presenting the results for each extracted frame, we opted to present in terms of mean and standard deviation. This aggregation also generalize the experiments and improve the readability of graphs. Additionally, we inverted the output v from the CNN models, to analyse quality level instead of quality loss level, also to be able to compare them with the other measures in this same behavior. It was accomplished by using the function $v_{inv} = 100 - v$. The 100 element in the function represents the highest possible output value v from the model and originally meant the highest quality loss, while 0 was the lowest quality loss. After the inversion, at the graphs, 0 represents the lowest quality level, while 100, the highest quality level.

In Figure 3, we present the results of the models FR and NR, and the measures PSNR and SSIM, varying values of `CRF`. We noted that the FR model describes more accurately the perceptual quality of the compressed videos than PSNR. In our experiments, PSNR had approximately a constant decreasing behaviour for all videos as the `CRF` increased and, therefore, does not corresponds well to our visual perception. Differently, the FR model presents a non-linear descending curve showing that the perceptual quality of compressed videos does not decrease at the same rate. This behaviour seems more natural to our visual perception since small degradations sometimes are not captured by our visual system. Furthermore, when the video begins to present distortions, as `CRF` values increases, our visual system begins to perceive the decreasing of quality more clearly, which agrees with the FR model.

Its important to notice that the FR value presents a larger standard deviation as `CRF` increases, especially for `park` and `house` videos. These videos present more details and movement, which affect negatively the compression. Differently, PNSR presents larger standard deviation for smaller values of `CRF`, thus corroborating its inability to correctly quantify the perceptual quality of the video.

Another important aspect is that the results of FR model have more similarities with SSIM than PSNR, even though they do not agree precisely. This is expected as the SSIM measure captures more characteristics of our visual system than PSNR does.

In contrast, the NR model did not exhibit an accurate description of our visual perception in most videos. This is due to the absence of information about reference frames, which compromises the ability of the method to infer the perceptual quality. For example, in the video `ducks`, the NR model obtained very low fluctuation as the `CRF` value increased. We believe that this behaviour happened because the frames were very similar. As we compress the video, it is expected its perceptual quality to diminish. Yet, it can be observed that the curves representing `park` and `town` videos, in Figure 3b, showed an unexpected oscillation (as also a large standard deviation), with exception of the video `town` compressed with h.265. However, in `house`, the NR model was very close to the FR model and described the perceptual quality of the compressed video more accurately. Such result may be due to the presence of reference in the training process. Therefore, there are cases in which the NR model can be used to infer the human perception of quality in compressed videos. Further investigation is needed to point the cases that the usage of NR model is appropriate.

According to FFmpeg documentation, `CRF` values around 17-18 were expected to generate compression without quality loss perceivable by our visual system. However, as our results show, this threshold appears to be around 25-26 when using the FR model, for the videos presented in this paper. Therefore, more aggressive compressions can be used, saving space and, consequently, improving transmission.

In this paper, we do not show the sequences of videos to compare with the results. We suggest running the scripts publicly available in our repository on https://bitbucket.org/luizcoro/seminario-multimidia-2019/ to have access and reproduce the compressed videos.

IV. Conclusions

In this work we evaluated the results of the methods created by Bosse et al. in [1], using various frames as test samples extracted from several compressed videos. For this work, we used four raw videos. We generated different quality levels of compression for each video. We also utilized h.264 and h.265 codec compression in order to explore the effects of the loss levels in the result of the automatic evaluation.

In terms of NR assessment of images, it can be noticed that the results is sometimes equivocated, as the methods suggest that visual perception alternates sometimes between lower and higher values, even though the quality in our tests only decreases. Also, as shown in Figure 3b, the algorithm predicted a kind of uniform level of quality despite the constant decreasing of compression quality. We believe that it is due to the fact that that video has very similar frames.

In contrast, results also demonstrated that the proposed methods of Image Quality Assessment using Deep CNN has a great effectiveness in most cases when using FR approach. Despite the CNN models have only been trained with single pictures, exploiting only spatial redundancies, FR method was able to infer perceptual quality on compressed video frames. Indicating that the approach covered in this paper can be considered a feasible solution for IQA of video frames specially in FR approach. As future work, we propose to investigate further the cases in which the usage of NR model is appropriate.

Acknowledgment

André R. Backes gratefully acknowledges the financial support of CNPq (National Council for Scientific and Technological Development, Brazil) (Grant #301715/2018-1). Marcelo Z. do Nascimento gratefully acknowledges the financial support of CNPq (National Council for Scientific and Technological Development, Brazil) (Grant #304848/2018-2). Luiz F. A. Brito gratefully acknowledges the financial support of FAPEMIG (Foundation to the Support of Research in Minas

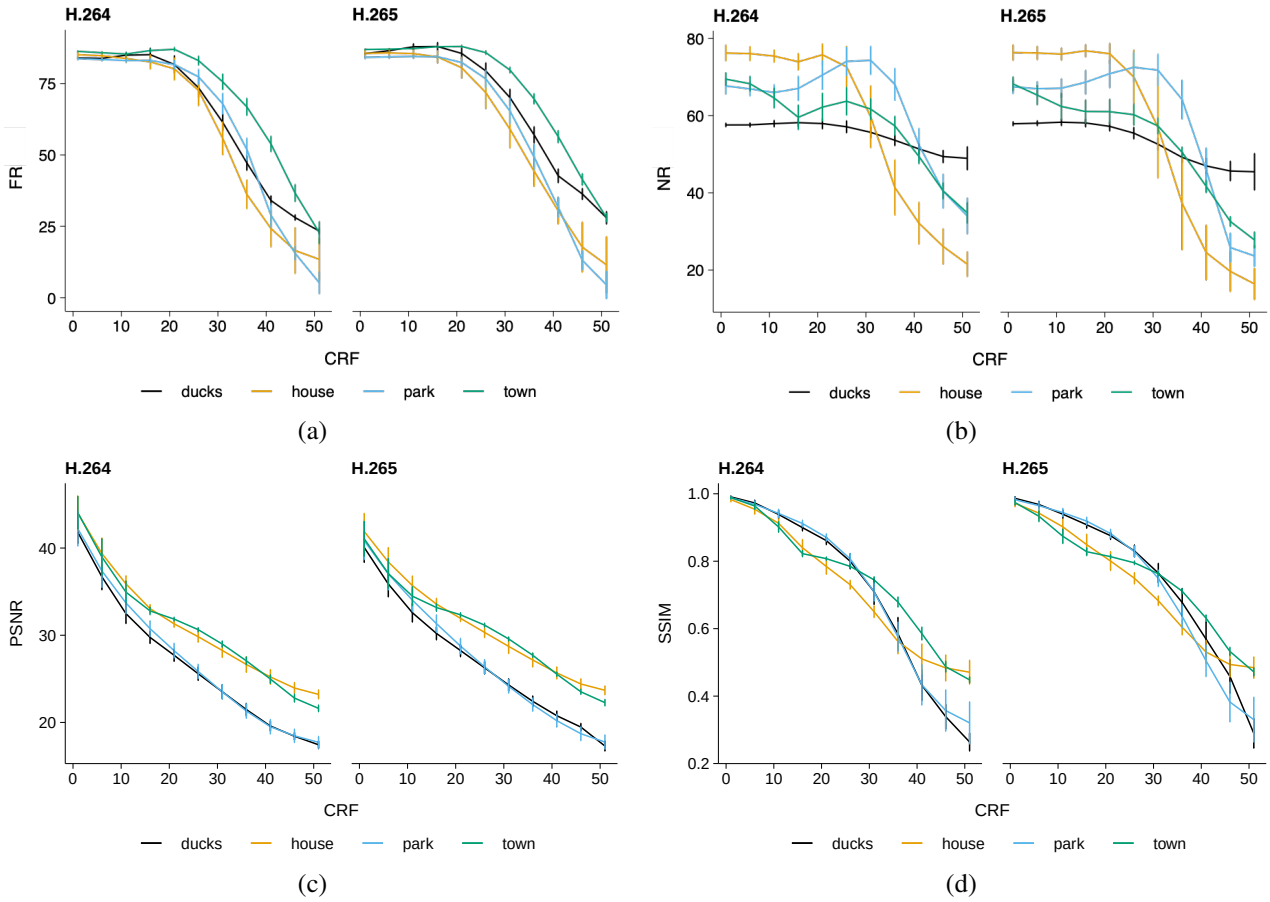


Fig. 3. Results for each video using h.264 and h.265 video codecs. In x axis, we vary the values of `CRF` while, in y axis, we present the results for the FR model (a), NR model (b), PSNR(c) and SSIM (d). Each line describes the different videos that we previously presented in Figure 2.

Gerais). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

REFERENCES

[1] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Trans. Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.

[2] Z.-N. Li, M. S. Drew, and J. Liu, *Fundamentals of Multimedia*, ser. Texts in Computer Science. Springer, 2014.

[3] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *ICASSP*. IEEE, 2002, pp. 3313–3316. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7874

[4] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J.-R. Ohm, and G. J. Sullivan, "Video quality evaluation methodology and verification testing of hevc compression performance," *IEEE Trans. Circuits Syst. Video Techn*, vol. 26, no. 1, pp. 76–90, 2016.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 10 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[6] L. Kang, P. Ye, Y. Li, and D. S. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *CVPR*. IEEE Computer Society, 2014, pp. 1733–1740. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6909096

[7] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. J. Kuo, "Image database tid2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, vol. 30, pp. 57 – 77, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596514001490

[8] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, Nov 2006.

[9] Xiph.Org Foundation, "Non-profit corporation dedicated to protecting the foundations of internet multimedia from control by private interests," https://www.xiph.org/, 1994–2019.

[10] FFmpeg Software, "Complete, cross-platform solution to record, convert and stream audio and video," https://ffmpeg.org/, 2000–2019.

[11] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Visually Lossless H.264 Compression of Natural Videos," *The Computer Journal*, vol. 56, no. 5, pp. 617–627, 07 2012. [Online]. Available: https://doi.org/10.1093/comjnl/bxs105

[12] V. Sze and M. Budagavi, "High throughput cabac entropy coding in hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1778–1791, Dec 2012.

[13] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[14] D. Taubman and M. Marcellin, *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice*. Springer Science & Business Media, 2012, vol. 642.

[15] ImageMagick Software, "Free software delivered as a ready-to-run binary distribution or as source code that you may use, copy, modify, and distribute in both open and proprietary applications," https://imagemagick.org/index.php, 1987–2019.