Pattern Recognition Letters 000 (2017) 1–10

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# TWM: A framework for creating highly compressible videos targeted to computer vision tasks

Fernanda A. Andaló[a,*], Otávio A.B. Penatti[b], Vanessa Testoni[b]

[a] Institute of Computing, University of Campinas (Unicamp), Av. Albert Einstein, 1251, Campinas, SP 13083-852, Brazil
[b] SAMSUNG Research Institute, Av. Cambacica 1200, Campinas, SP 13097-160, Brazil

ARTICLE INFO

*Article history:*
Available online xxx

*Keywords:*
Frame composition
Object detection and tracking
Video compression
Computer vision applications

ABSTRACT

We present a simple yet effective framework – *Transmitting What Matters* (TWM) – to generate highly compressible videos containing only relevant information targeted to specific computer vision tasks, such as faces for the task of face expression recognition, license plates for the task of optical character recognition, among others. TWM takes advantage of the final desired computer vision task to compose video frames only with the necessary data. The video frames are compressed and can be stored or transmitted to powerful servers where extensive and time-consuming tasks are performed. Experiments explore the trade-offs between distortion and bitrate for a wide range of compression levels, and the impact generated by compression artifacts on the accuracy of the desired vision task. We show that, for two computer vision tasks implemented by different methods, it is possible to dramatically reduce the amount of required data to be stored or transmitted, without compromising accuracy. With $PSNR_{YUV}$ quality of over 41 dB, the bitrate was reduced up to four times, while a detection task was affected by only ∼1 pixel and a classification task by 1∼2 percentage points.

## 1. Introduction

Vision-based systems are becoming more popular nowadays, especially because of the increasing power of devices and the new capabilities for information storage. Such systems tend to become even more popular in the near future with the huge availability of data for training machine learning methods. Not only the online availability of images and videos is increasing, yet also their resolutions are getting higher year after year.

According to Cisco [12], the Internet traffic will reach 1 petabits per second (Pbps) in the busiest hour by 2018 and video data will be responsible for approximately 79% of it. This is the equivalent of 335 million people streaming high-definition (HD) video continuously. In such conditions, it is of crucial importance to develop solutions to reduce the amount of video data to be transferred through the network. Having less data also reduces the impact of storage requirements in any system. Even in scenarios where there are no concerns about infrastructure and bandwidth limitations, the transmission and storage of entire videos in ultra-high definitions (UHD), such as 4K and 8K, are challenging.

When concerning high-resolution images and videos for computer vision tasks, simply reducing their spatial or temporal resolution would be a straightforward solution, however it is not always an option since low resolution data make most computer vision techniques much less precise.

Tasks like face recognition, for instance, require faces to be represented with a minimum resolution in order to perform the recognition effectively. As an example, consider a classroom scenario in which students' faces are recorded and transmitted to an external server which performs face recognition. Faces recorded at 5–10 m away from the camera with high resolution of 1080p (1920 × 1080) contain around 65–30 pixels horizontally, i.e., critically close to the lowest resolution required by current face identification applications [3]. Therefore, at this minimum required resolution, an already compressed video of a class would require gigabytes of storage space. Taking into account that multiple classrooms may be recorded daily and simultaneously, the school would need to store and transmit an incredible amount of information. Naturally, this huge amount of generated video information is not only a problem in the school scenario, but also in other systems that require high-resolution videos, like surveillance systems, sports, etc.

In this paper, we propose an interesting alternative framework, named *Transmitting What Matters* (TWM), that saves storage and still keeps enough resolution, by composing videos with only

[*] Corresponding author.
*E-mail address:* feandalo@ic.unicamp.br (F.A. Andaló).

http://dx.doi.org/10.1016/j.patrec.2017.08.025
0167-8655/© 2017 Elsevier B.V. All rights reserved.

relevant data for a considered computer vision task. The solution allows high compression while keeping enough resolution for the final tasks. Besides, there is a double gain, one related to the content generation and another to the optimized compression.

We evaluate the proposed framework both in terms of compression rate, comparing how much we can reduce the videos to be transmitted and stored, and also in terms of the impact of the proposed method in the accuracy of the final computer vision task, analyzing the compression trade-offs. This is another important contribution of this work, since here the computer vision tasks are performed considering not raw uncompressed video datasets, but compressed videos with all the intrinsic artifacts caused by compression techniques, such as blocking and blur. Since the majority of online available images and videos is compressed, it is fundamental to analyze the effects of compression on the accuracy of computer vision tasks that will be mostly performed on compressed data.

For evaluation and illustration purposes, we consider the school scenario in which videos of a classroom are recorded and the students' faces are detected. Two computer vision tasks will be performed: detection of facial landmarks and gender classification. For both tasks, we show that we can obtain up to four times of reduction in the amount of data to be transferred and stored. At the same time, we show that despite the high compression in the data, both tasks can still be effectively performed in the decoded video.

In our previous work [1], we evaluated the framework for just one task implemented by a single method. In this work, we reinforce the method robustness and flexibility by adding an additional task and different implementation methods, besides thoroughly evaluating the framework.

It is important to emphasize that the school scenario and the two mentioned computer vision tasks were selected for evaluation and illustration purposes. There are many other scenarios where the proposed framework is useful. For instance, in traffic surveillance, hours of videos must be recorded with a constant background (highway) so that cars' license plates can be later detected and their letters and numbers recognized.

The remainder of the paper is the following. Section 2 presents related work. In Section 3, we detail the proposed framework. Section 4 presents the experiments and obtained results. Finally, Section 5 concludes the paper and shows opportunities for future work.

## 2. Related work

Current solutions to compress videos and reduce bandwidth usage do not address the entire process of optimized creation and compression depending on the desired final task.

In the literature, there are basically two related solutions: *Tiled streaming* [20,24,25] and *Region-of-Interest (RoI) video encoding* [10,13,19,30,31,36,41]. Other methods focus on video codecs [15] or protocols [4] to perform adaptive transmission.

*Tiled streaming* methods encode a video sequence by dividing frames into a grid of independent tiles which are scalably encoded and stored. This content can then be streamed with a spatial or quality resolution compatible with the available bandwidth. A lower resolution version of the sequence can be initially transmitted until a user selects a region of interest, by zooming-in. After that, only the additional bits for representing the tiles covering the selected RoI in higher resolution are transferred. Therefore, these methods potentially reduce bandwidth consumption, although they do not reduce storage requirements due to the higher size of the scalably coded file.

In *RoI video encoding* methods, foreground-background identification is conducted, allowing background regions to be more compressed. Even though the background is highly compressed, these methods still need to encode it, since the whole frames have to be reconstructed in the typical application scenarios. *RoI video encoding* is also included in perceptual video coding [26], which is a rising research area that includes perceptual properties of the human visual system in the coding models and implementations.

In [10], the authors present a RoI encoding system for video conference. The encoded video stream, which contains the RoIs in a good quality and the background in a bad quality, is transmitted to all receiving clients which decode it, crop out and render the RoIs. Differently from Bulla et al. [10], our framework group the RoIs, or objects, in the frames before transmitting the video, which permits savings in data transmission.

In [41], the authors propose a RoI-based HEVC coding for conversational videos. They encode face regions with adjustable quantization parameters depending on the importance of the region within the face, by generating a pixel-wise weight map. The result is a better visual quality in face regions for conversational videos. The method in [41] does not compose a new video containing only the RoIs, however it could be adapted and used at one of the modules of our framework, specifically at the video encoding step. In such case, we would optimize the quality of face regions, harming even less the final computer vision task, if it is related to face analysis.

TWM is not a new codec or protocol, but rather a scheme that enables current codecs to produce more effective results. We propose a framework to transmit only what matters, saving storage, bandwidth, and still keeping enough resolution for performing complex computer vision tasks at the receiver side. In this sense, to the best of our knowledge, our framework is not comparable with any other in the literature.

## 3. Transmitting what matters

In this section, we detail the proposed framework – *Transmitting What Matters* (TWM) – which generates compressed videos containing only the objects of interest.[1] Our explanation and examples are based on the school scenario, in which the final computer vision task is related to the facial analysis of students recorded in a classroom. However, the framework is general enough to work in many other scenarios, like face recognition in surveillance systems, visual analysis of athletes in sports systems, crop identification or plague analysis in agricultural systems, license plate recognition in traffic surveillance systems, and others.

The pipeline illustrated in Fig. 1 summarizes the framework.

The system receives as input a digital video as well as parameters that inform the category of objects of interest and the desired spatial resolution for these objects. Based on the provided data, **for each frame** of the input video, TWM:

(1) detects and extracts the objects of interest, considering the informed category (faces, for example);
(2) adjusts the spatial resolution of the extracted objects according to the resolution parameter;
(3) composes a final frame with the extracted and adjusted objects grouped spatially in a grid.

Finally, the new frames are joined and encoded with a state-of-the-art video codec which benefits from the visual similarities and local correlations, both spatially in each frame and temporally across several frames. These visual similarities considerably improve the effectiveness of the video codec, consequently increasing the compression capacity.

[1] TWM is patented by Samsung – Grant US9699476B2, Publication date 2017-07-04.

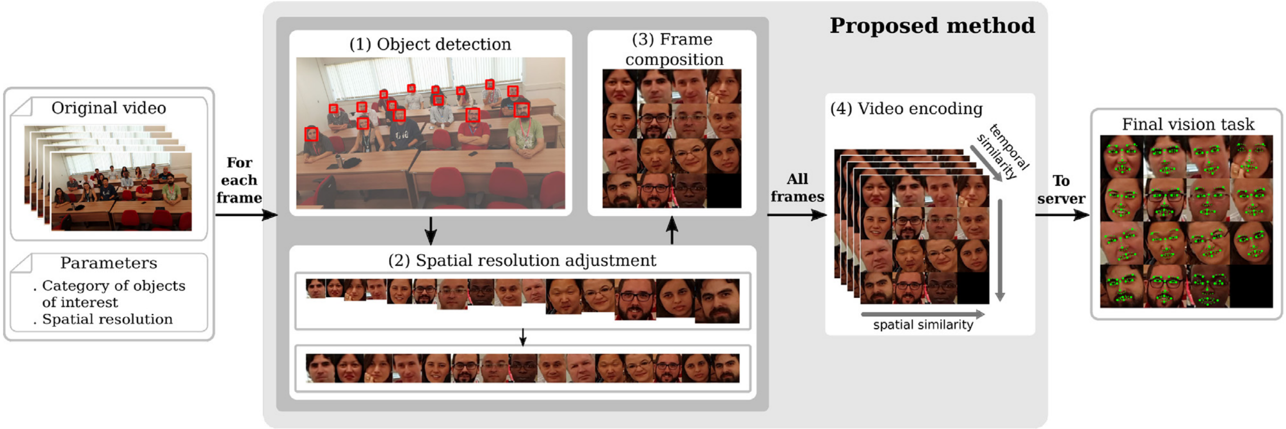


**Fig. 1.** Proposed framework: creating compressed videos taking into account a target computer vision task. For each frame of the input video, steps (1) Object detection, (2) Spatial resolution adjustment, and (3) Frame composition are performed. All created frames are encoded in a compressed video at step (4) Video encoding. Finally, the compressed video can be transferred to a server where the final computer vision task takes place.

After the final video encoding step, the generated compressed video can be transmitted to a server where the computer vision task will take place. If there is more than one category of objects (for instance, faces and hands), the whole process is repeated for each category, therefore generating several compressed videos.

In the following subsections, we detail each step of the proposed framework.

### 3.1. Object detection

The object detection step receives as input the original video and parameters specifying the category of the objects of interest (e.g., face, car, license plate, etc.). Object detection is then performed considering the category informed as parameter.

There are many possibilities to implement this step, since, in the literature, many precise object detectors work with a large number of object categories. Such detectors include Deformable Part Models [17,18], Faster R-CNN [35], YOLO [34], SSD [29], among others. One could also consider the use of object detectors trained specifically for the specified object category, which could be possibly more precise than generic detectors.

It is important to emphasize that due to the framework flexibility, any object detector could be selected based on the scenario requirements and restrictions (higher accuracy, lower complexity, etc.). The framework itself does not interfere with the performance of the object detector, since it is applied directly on the original video and would be necessary even if our framework was not being used.

In our example, where the objects of interest are faces, we used the OpenCV [7] implementation of the Viola and Jones face detector [38] improved by Lienhart and Maydt [28]. It is a classic method which employs a cascade of boosted trained classifiers to search for the object of interest, at different sizes, in an image.

At this step, objects can also be tracked across frames to help the frame composition step. Depending on the category, a myriad of tracking algorithms can be used [2,8].

We employed a simple tracking procedure, by matching detected objects using the absolute difference norm between objects in the previous frame and the actual frame. More specifically, consider a list $L_j = \{f_{1,j}, \dots, f_{n_j,j}\}$, where $f_{i,\,j}$ is the face at position $i$ in the grid of frame $j$, and $n_j$ is the number of detected faces in frame $j$. Similarly, the list of detected faces in frame $j+1$ is $L_{j+1}$. The face $f_{i,j+1}$ which will be placed at position $i$ in the grid of frame $j+1$ is the one in $L_{j+1}$ that minimizes the absolute difference norm to $f_{i,\,j}$:

$$f_{i,j+1} = \text{argmin}_{f \in L_{j+1}} \sqrt{\sum_p (f(p) - f_{i,j}(p))^2}, \qquad (1)$$

where $p$ corresponds to a pixel position within a face.

### 3.2. Spatial resolution adjustment

This step receives as input the objects of interest represented as image tiles already cropped from the original frame and the spatial resolution informed as parameter.

All the image tiles are adjusted in order to be represented according to the spatial resolution parameter. The target spatial resolution needs to be selected according to the final desired vision task, because different tasks often require a specific minimum image resolution for accurateness.

To adjust the spatial resolution of the detected objects, we used cubic interpolation for the up-sampling process, and area interpolation for the down-sampling process.

### 3.3. Frame composition

For each input video frame, the tiles with the detected objects (already spatially adjusted by the previous step) are organized in a grid. The grid can have different forms, like a single row or a single column, for instance. However, for better exploiting the video codec capabilities of taking advantage of both horizontally and vertically local correlations, the grid should be configured as a rectangle or square of variable size according to the number of detected objects.

The size of the grid can be configured to fit the number of expected objects (for example, according to the maximum capacity of a classroom). Even if the grid is sparsely filled, the estimation and motion compensation techniques inherent to the compression algorithms would efficiently detect and compress the empty regions without using additional bits.

By using the tracking information generated by the object detection step, TWM can place the same object at the same grid position at all frames, in order to obtain even higher compression rates due to the inter frame prediction techniques employed by current codecs. However, tracking is an optional procedure, which is useful for scenarios with repeating, steady objects. For scenarios with fast moving objects, such as traffic surveillance, the detected objects (e.g. license plates) change on a frame-by-frame basis. Therefore,

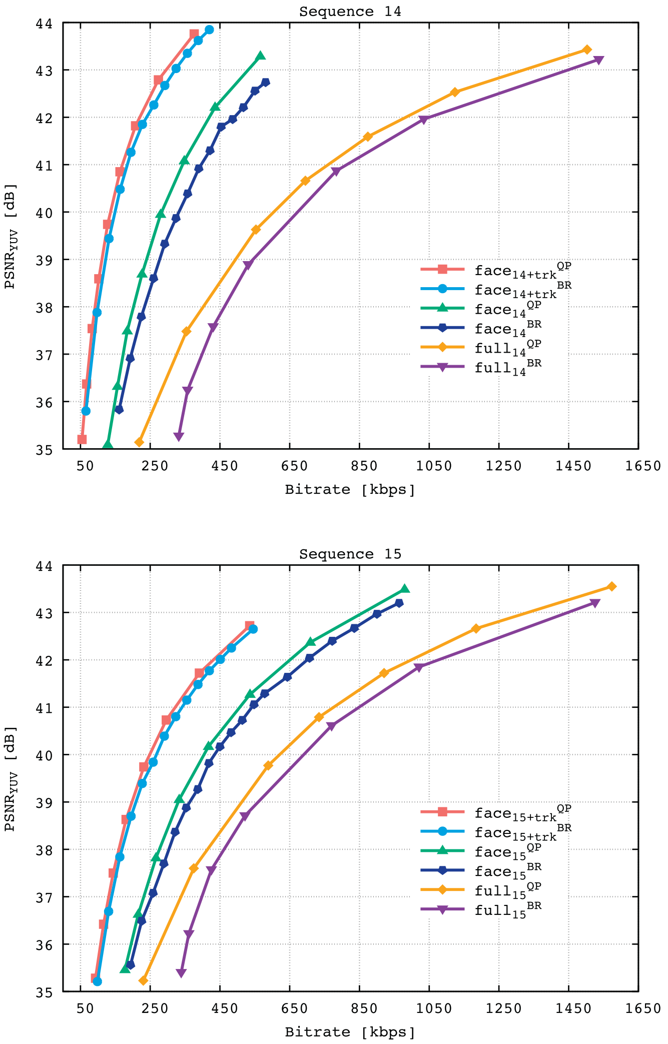


**Fig. 2.** Bitrate [kbps] × $PSNR_{YUV}$ [dB] curves for two video sequences. The best result, shown in red as $face^{QP}_{i+trk}$, is reached when fixed QP and tracking are both employed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in this case, applying a tracking procedure may not be so beneficial.

### 3.4. Video encoding

Finally, in the video encoding step, the generated frames are joined and encoded with a video codec which benefits from the visual similarities and local correlations, both spatially in each frame and temporally across several frames.

Current codecs, such as the popular H.264/MPEG-4 Advanced Video Coding (AVC) [21] or the current state-of-the-art High Efficiency Video Coding (HEVC) [9], employ very efficient intrapicture and interpicture prediction methods as well as advanced motion estimation and compensation techniques [37,39]. Therefore, they are able to exploit all the high spatial and temporal correlations introduced in the generated videos, where the frames contain similar objects placed in similar positions.

On top of that, codecs enable several configuration parameters that can be optimized according to the final main pursued compression result. If it is more important to ensure that a fixed number of frames, named a group of pictures (GOP), is encoded with approximately the same bitrate (despite of the actual content of each GOP), the codec parameters can be chosen to target a final specific bitrate through its inherent rate control mode. On the other hand, if the final application requires an approximately constant quality for all decoded frames, then the codec parameters can be set to reach a fixed quality at the cost of a final variable bitrate. The main codec parameter responsible for each compression result is the quantization parameter (QP).

In the proposed framework, it is key to ensure that all decoded frames achieve a minimum quality required to the satisfactory performance of the final computer vision task. Therefore, each object (or pixel area, since codecs do not perform object recognition) will be encoded with a QP chosen by the codec which employs rate-distortion optimized quantization (RDOQ) techniques, such as in [23]. Consequently, on average, a higher compression level, or a higher QP, will be applied to previously down-sampled high-resolution objects; and a lower compression level, or a lower QP, will be applied to previously up-sampled objects that may already be blurry and cannot afford to be highly quantized.

It is important to note that a system that does not employ our framework would still have to encode the original high-resolution video. The amount of raw 4K UHD video generated in the considered scenario, and in other similar scenarios, easily reaches the terabyte range for only one 30min recorded video. Encoding such 4K UHD videos with current state-of-the-art video codecs requires a considerable amount of time [6], even when optimized implementations of the codecs are used.

Therefore, by reducing the total amount of data to be encoded and by creating a much more compressible final video content, we not only save bandwidth and storage, but also dramatically reduce the video encoding processing time.

## 4. Experiments and results

The experiments, which were conducted on a dataset of captured video sequences (Section 4.1), verified how much video compression can be obtained with TWM (Section 4.2), and how the introduced compression artifacts affect the final computer vision tasks (Section 4.3).

### 4.1. Video sequences

A good approach to evaluate TWM, with all its modules, is by considering high resolution videos with frames containing the object of interest appearing multiple times along the frames, in different resolutions and positions. Such videos would require compression before being transmitted or stored, while also keeping enough resolution to avoid harming the final task. Since we could not find a standard dataset with such characteristics, we captured 15 video sequences in 4K UHD resolution (3840 × 2160 pixels) at 30 fps. All sequences are progressively scanned and use the YUV 4:2:0 color format with 8 bit per color sample. Each video sequence has 420 frames and was acquired in an environment simulating a classroom, with the camera in front of the room recording the students.

The notation presented in Table 1 is used to designate each video sequence considered in the experiments. Each one of the 15 original high-resolution video sequences is identified by $full_i$, where $i$ is the video sequence number. The notation $full_i^{BR}$ identifies each video sequence $full_i$ compressed with fixed bitrate (BR). Similarly, $full_i^{QP}$ identifies each video sequence $full_i$ compressed with fixed QP.

By applying the initial three steps of our framework and prior to the video encoding step, the raw videos containing only the faces of the students are identified by $face_i$. After the video encoding step, the corresponding compressed videos are identified by $face_i^{BR}$ when compressed using fixed bitrate, or by $face_i^{QP}$ when using fixed QP.

**Table 1**
Description of the video sequences analyzed in the experiments and respective notation. The *face* sequences are generated using TWM.

| Frame sample | Notation | Description |
|---|---|---|
| | $full^{BR}$ | 4K original video compressed by HEVC with fixed bitrate |
| | $full^{QP}$ | 4K original video compressed by HEVC with fixed quantization parameter |
| | $face^{BR}$ | Sequence containing only faces, compressed by HEVC with fixed bitrate |
| | $face^{QP}$ | Sequence containing only faces, compressed by HEVC with fixed quantization parameter |

When composing the $face_i$ videos, TWM can benefit from the tracking algorithm. These sequences are identified by the “+trk” string after the video sequence number.

### 4.2. Video encoding results

We selected the HEVC as the most suitable video codec because it is shown to be especially effective for low bitrates and high-resolution video content [32]. The official HEVC HM 16.4 test model software [22] was used. For the HM encoder parameter definition, we employed the test conditions and software reference configurations officially recommended by the Joint Collaborative Team on Video Coding [5].

Video encoding results are shown through distortion × bitrate curves and visual quality comparisons. Since the codecs performance and achieved bitrates depend on the content which varies per video sequence, it is not possible to report video encoding results as an average for the dataset. Therefore, we selected two video sequences from our dataset, numbered as 14 and 15, for reporting results. These two sequences correspond to different moments in the class after students changed places.

We show the rate-distortion curves of the combined luminance (Y) and chrominance (U and V) components in Fig. 2. The combined Peak Signal-to-Noise Ratio is computed as

$$PSNR_{YUV} = (6PSNR_Y + PSNR_U + PSNR_V)/8. \qquad (2)$$

The first interesting result shown in Fig. 2 regards the encoding performance for the original 4K $full_{14}$ and $full_{15}$ sequences (the results obtained for these two sequences are our baseline, since, as mentioned in Section 2, to the best of our knowledge, there is no other method available in the literature to be used for comparison). Since the content of our dataset video sequences is highly compressible, with a static classroom background and seated students watching the class with a limited range of motion, HEVC impressively reaches less than 1 Mbps while keeping more than 40 dB. As reported in [14], HEVC reaches around 3 Mbps when encoding 4K sequences with similar quality.

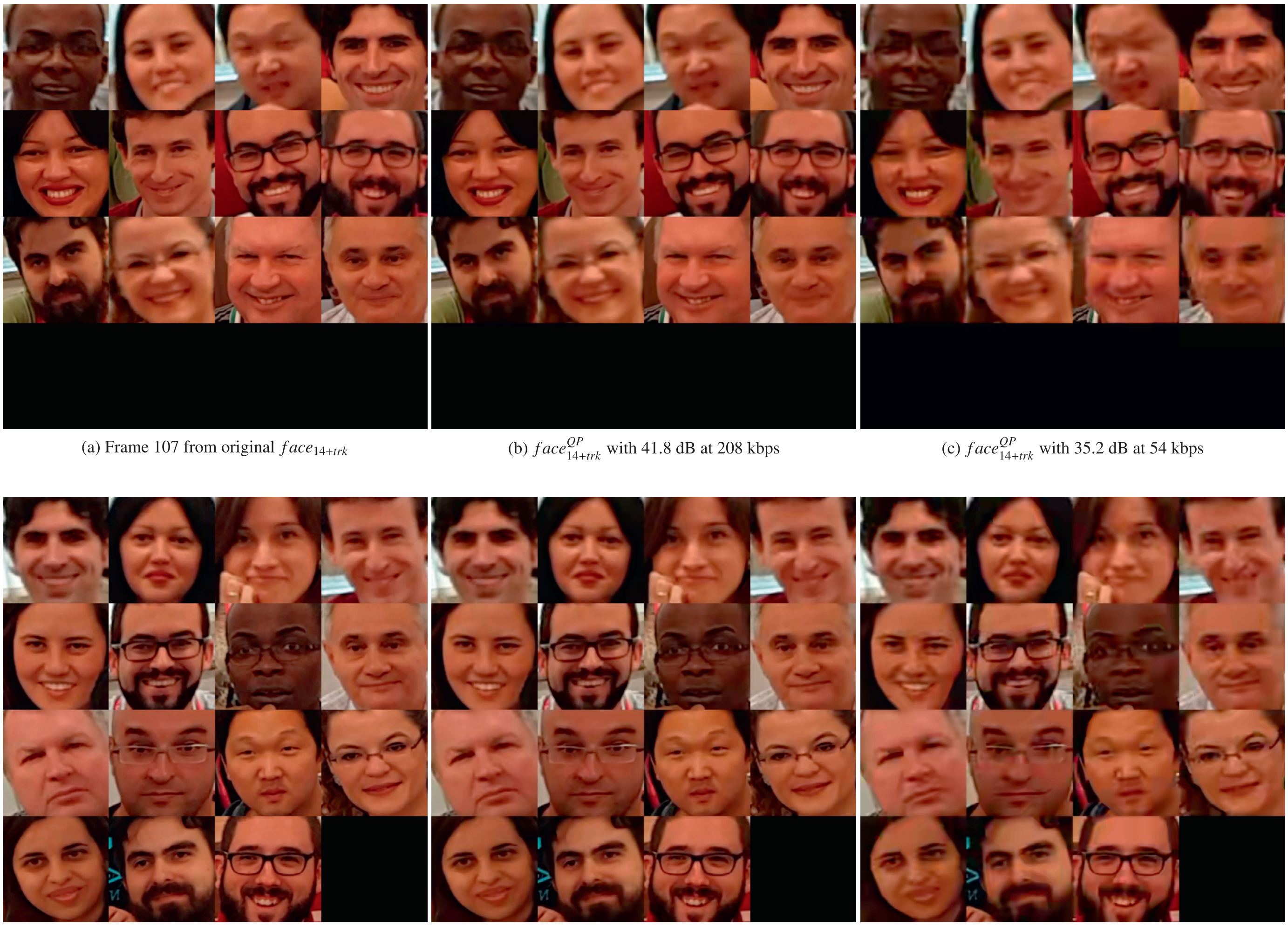
Also due to the nature of our video sequences, the encoding performance with fixed QP (sequences $full^{QP}_{14}$ and $full^{QP}_{15}$) is always better than the encoding performance with fixed bitrate (sequences $full^{BR}_{14}$ and $full^{BR}_{15}$). The gains obtained with fixed QP are more perceptible at the challenging low bitrate scenario (around 300 kbps), where the quality difference between the two modes achieves around 2 dB for similar bitrates.

Another worth mentioning point regards the significant gains obtained with the tracking algorithm in the encoding of the face videos. These gains were expected because, even though faces in general are already similar, placing the same faces in similar positions throughout the video adds even more similarities and temporal correlations which are properly exploited by HEVC. In Fig. 2, for similar bitrates, the quality gains obtained with tracking for $face_{15+trk}$ sequence are up to 3.5 dB, and for $face_{14+trk}$ sequence the gains are up to 5.5 dB.

The differences between the two sequences are due to the fact that $face_{15}$ sequence has, on average, more information than $face_{14}$ sequence, since more students are present in the former. This is also the reason why, for similar $PSNR_{YUV}$ qualities, the bitrates achieved for $face_{15}$ sequence are always higher than the bitrates for $face_{14}$ sequence. Even though these sequences are not 4K and are comprised by only faces, they still contain a considerable amount of data to be compressed since each face is represented as a 128 × 128 square and the whole frames, with at most 15 faces, have a resolution of 512 × 512 pixels.

In Section 4.3 it will be shown that $PSNR_{YUV}$ qualities above 41 dB are sufficient to effectively perform the selected computer vision tasks. As can be seen in Fig. 2, the gains obtained with our framework are increasingly higher in the quality range above 40 dB. For sequence 14 at 42 dB, for instance, the bitrate can be reduced 4 times, from around 1 Mbps for $full^{QP}_{14}$ to 250 kbps for $face^{QP}_{14+trk}$, when compared to the baseline where the full video is encoded. At the same quality of 42 dB for sequence 15, the bitrate can be reduced 2.2 times, from around 1 Mbps for $full^{QP}_{15}$ to 450 kbps for $face^{QP}_{15+trk}$.

A visual quality comparison is presented in Fig. 3. Frames 107 and 33 extracted from sequences $face_{14+trk}$ and $face_{15+trk}$



(a) Frame 107 from original $face_{14+trk}$ (b) $face^{QP}_{14+trk}$ with 41.8 dB at 208 kbps (c) $face^{QP}_{14+trk}$ with 35.2 dB at 54 kbps

(d) Frame 33 from original $face_{15+trk}$ (e) $face^{QP}_{15+trk}$ with 41.7 dB at 391 kbps (f) $face^{QP}_{15+trk}$ with 35.3 dB at 93 kbps

**Fig. 3.** Visual quality comparison results for sequences $face_{14+trk}$ and $face_{15+trk}$, encoded with different $PSNR_{YUV}$ qualities.

are shown, respectively, in Fig. 3(a) and in Fig. 3(d). Results in Figs. 3(b), (c), (e) and (f) show both frames encoded in different qualities with corresponding compression artifacts. We can see that frames at 41 dB have almost no compression artifacts and are visually very similar to the uncompressed versions, although they are encoded with much less bits. It is not expected all faces to show the same quality, due to the effect of the spatial resolution adjustment step. The students in the back of the classroom were captured in small resolutions and their faces had to be up-sampled, which generated the blurry effect. The faces of the students in front of the classroom (closer to the camera) were captured in high resolutions and, even after being down-sampled, were still preserving enough details. Artifacts created by compression can be easily observed in Figs. 5 and 6.

We opted for not reporting processing times because the framework allows the user to freely choose the codec and its implementation, such as the previously mentioned H.264/MPEG-4 AVC or the HEVC. For the HEVC, for instance, one could choose the official HEVC HM test model software, which was used in this work, or the x265 [33], an open-source computationally efficient HEVC implementation. The processing times required for performing the other framework steps (object detection, spatial resolution adjustment and frame composition) also vary depending on the chosen techniques for implementing each step.

### 4.3. Impact on computer vision tasks

To analyze the impact of the proposed framework on tasks performed after the decoding of the generated videos, we chose two tasks: face landmark detection, which refers to the detection of keypoints in faces that can be further used, for instance, to aid the analysis of facial expressions [11]; and gender classification, which refers to the automatic inference of a person's gender from a face image.

The impact on these two tasks is analyzed separately in the next sections. In the following experiments, we are considering video sequences compressed with fixed QP and using the tracking algorithm, which were the configurations in which TWM achieved the best results as shown in Section 4.2.

#### 4.3.1. Landmark detection

In order to detect facial landmarks, we employed two different methods: Intraface [40] and Face++ [42].

The Intraface method [40] builds on drawbacks of 2nd order descent methods, specifically the fact that the function to be optimized may not be differentiable or the Hessian may be large and not positive definite. The authors propose a Supervised Descent Method (SDM) for minimizing a Non-linear Least Squares (NLS) function. In the training phase, SDM learns a set of generic descent directions that minimizes the mean of NLS functions sampled at

Please cite this article as: F.A. Andaló et al., TWM: A framework for creating highly compressible videos targeted to computer vision tasks, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.08.025

different locations; and in the testing phase, SDM minimizes the NLS objective considering the learned descent directions. To locate the landmarks, the NLS function $f(x)$ to be minimized is a non-linear function, where $x$ represents the landmark locations which will be aligned to a known template consisting of 66 landmarks. Here we only use 49 landmarks for evaluation (all landmarks except for the jaw points).

The Face++ method for detecting landmarks [42] utilizes a coarse-to-fine convolutional network cascade with four levels. Each network level is trained to locally refine a subset of landmarks generated by previous levels, in addiction to predicting explicit geometric constrains. Deep Convolutions Neural Networks (DCNN) are used as building blocks of this system. To run the experiments related to Face++, we used the available online API,[2] which returns 5 detected landmarks (the center of each eye, far left and far right points in the mouth, and nose tip).

We compared the displacement of facial landmarks detected by the Intraface and Face++ method in two versions of the videos, considering $i = 1, \dots, 15$:

- after the three initial steps of TWM, but prior to the encoding step ($face_{i+trk}$);
- after the encoding step with fixed QP ($face^{QP}_{i+trk}$) and different compression levels.

The mean displacement error is computed as the sum of per-landmark displacement ($L_1$-norm) for the same face between correspondent frames of two different sequences, divided by the total number of facial landmarks. Then, it is normalized by the number of faces in the frames and by the number of total frames in the considered sequence.

For the Intraface method, we also computed mean displacement errors between features detected on the original high-resolution $full_i$ videos (before TWM) and the $face_i$ sequences (before encoding). The errors are negligible, being less than 1 pixel and standard deviation of 0.1 pixel. This small difference is due to round-off errors caused by the up and down-sampling processes, which consider real numbers as scale factors.

Fig. 4 presents $PSNR_{YUV}$ × mean displacement error × bitrate curves for sequences 14 and 15. They show mean displacement errors, and standard deviations, between facial landmarks detected in the raw sequences $face_{i+trk}$ and in the respective compressed sequences $face^{QP}_{i+trk}$, considering different compression levels.

For $PSNR_{YUV}$ values greater than 41 dB, the displacement error for the considered task is lower than 1 pixel, and the detection displacement for both sequences yields almost the same behavior. However, for lower reconstruction qualities, errors for sequence 14 are higher than for sequence 15, including higher standard deviations, because of the different content of the sequences. This is also corroborated by the previous results presented in Fig. 2 and by the achieved bitrates for the same $PSNR_{YUV}$ quality in Fig. 4, which are lower for sequence 14 than for sequence 15.

Note that although Face++ method yields lower mean displacement errors when compared to Intraface, the standard deviation is higher. Intraface method is based on an optimization process that takes a template into consideration, imposing stronger geometric constraints among the landmarks. This can explain the lower standard deviation. On the other hand, Face++ uses a more robust approach – cascade of DCNNs – to refine landmark locations, although it imposes weaker geometric constraints. This is why the mean displacement error is lower, at the same time that a few landmarks cannot be rightly localized due to compression artifacts, resulting in a higher standard deviation.

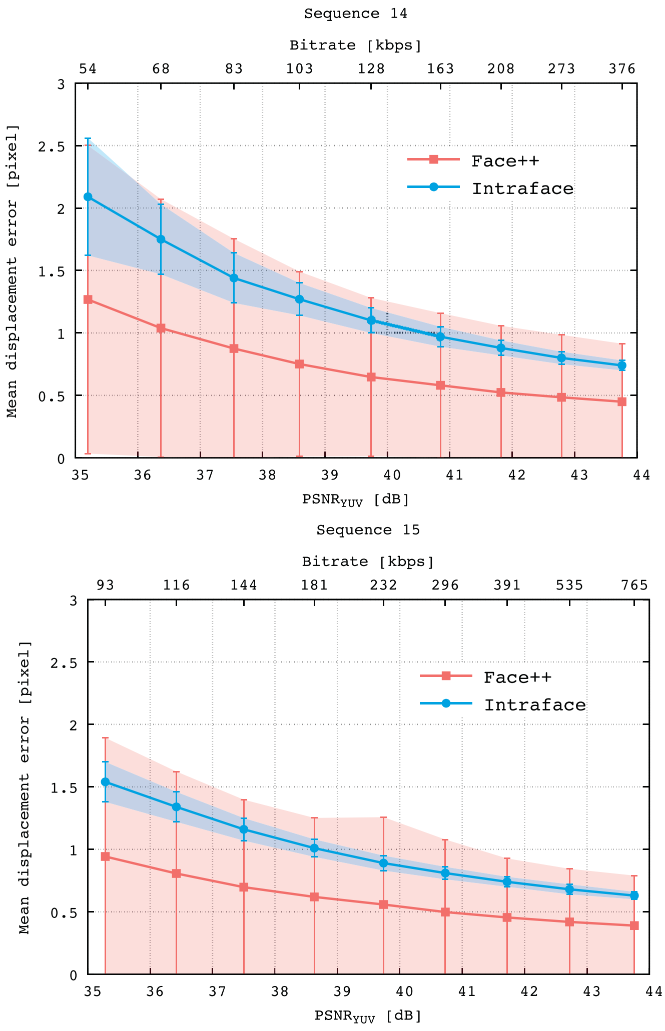
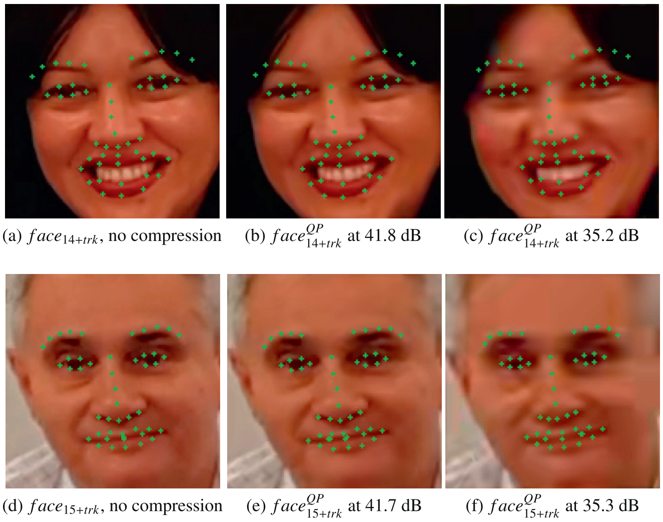


**Fig. 4.** Facial landmark displacement results ($PSNR_{YUV}$ [dB] × mean displacement error [pixel] × bitrate [kbps]). Errors were computed between raw sequence $face_{i+trk}$ and compressed sequence $face^{QP}_{i+trk}$, considering several compression levels. Error bars indicate the standard deviation of the computed error. Above quality 41 dB, errors are below 1 pixel.

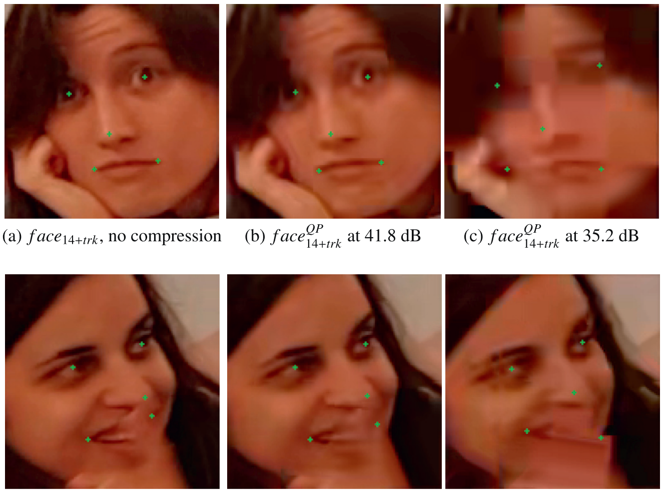

(a) $face_{14+trk}$, no compression (b) $face^{QP}_{14+trk}$ at 41.8 dB (c) $face^{QP}_{14+trk}$ at 35.2 dB

(d) $face_{15+trk}$, no compression (e) $face^{QP}_{15+trk}$ at 41.7 dB (f) $face^{QP}_{15+trk}$ at 35.3 dB

**Fig. 5.** Landmarks detected by the Intraface method, in different compression levels (the more to the right, the more compression). For faces (c) and (e), compression artifacts affect the detection more than the mean displacement error for their respective sequence. Face (c) was affected by 2.57 px and face (e) by 0.78 px. Errors are more visible in the eyes and nose areas.

[2] https://www.faceplusplus.com/

Please cite this article as: F.A. Andaló et al., TWM: A framework for creating highly compressible videos targeted to computer vision tasks, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.08.025



(a) $face_{14+trk}$, no compression (b) $face^{QP}_{14+trk}$ at 41.8 dB (c) $face^{QP}_{14+trk}$ at 35.2 dB

(d) $face_{15+trk}$, no compression (e) $face^{QP}_{15+trk}$ at 41.7 dB (f) $face^{QP}_{15+trk}$ at 35.3 dB

**Fig. 6.** Landmarks detected by the Face++ method, in different compression levels. Faces (c) and (f) were the most affected by the compression artifacts in their respective sequence. Face (c) was affected by 5.97 px; and face (f) by 3.41 px.

Fig. 5 shows examples of facial landmarks detected by the Intraface method, in which compression artifacts affected the detection more than the mean displacement error. Fig. 6 shows examples of facial landmarks detected by the Face++ method which were most affected by the compression artifacts. Larger displacements can be observed in features detected in frames with lower $PSNR_{YUV}$, and an almost perfect detection at qualities higher than 41 dB.

### 4.3.2. Gender classification

In order to perform gender classification, we employed two different methods: a Convolution Neural Network (CNN) specifically trained to classify face images in respect to gender, which we name here as GenderNet [27]; and Face++ method [16].

GenderNet [27] is a CNN comprised of three convolutional layers and two fully-connected layers. This shallow architecture is chosen to reduce the risk of overfitting and motivated by the small number of classes (binary classification). GenderNet receives RGB face images as input. Each convolutional layer is followed by a rectified linear operator (ReLU) and a max pooling layer. The two first fully-connected layers are followed by a ReLU and a dropout layer. The output of the last fully-connected layer is fed to a softmax layer that assigns a probability for each class.

The Face++ method for gender classification [16] uses a deep learning framework, called Deep Compactness Learning, which concomitantly optimizes the compactness and discriminative ability of face representation. The method uses this representation as input and builds a Look-Up-Table for classification. Due to the id-preserving property of the representation, images belonging to the same bin share common attributes. When the number of training sample is large enough, classifying becomes as simple as a table look-up. To run the experiments related to Face++, we considered the same online API previously used for the landmark detection task.

We compared the results for a compressed video $face^{QP}_{i+trk}$ to the results for the corresponding raw video sequence $face_{i+trk}$, regarding accuracy and difference between the returned probability distributions.

Accuracy is computed considering the classes assigned by the methods for the raw video sequence $face_{i+trk}$ as labels (true values). The difference between the probability distributions returned by the methods for a raw and compressed video sequences is com-

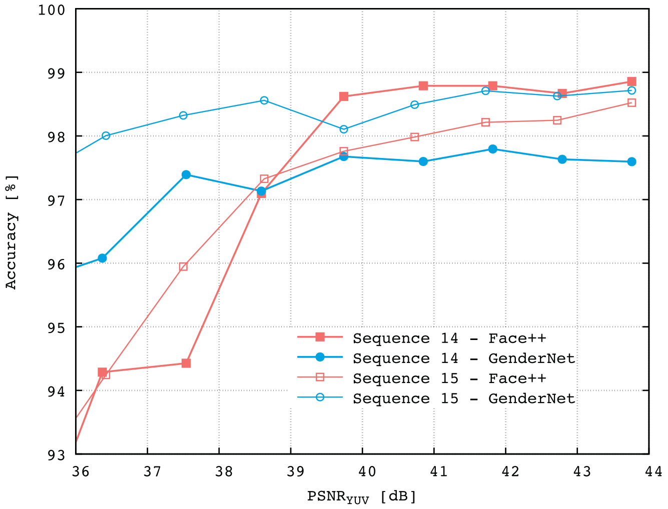


**Fig. 7.** Gender classification results ($PSNR_{YUV}$ [dB] × accuracy). Accuracy was computed between raw sequence $face_{i+trk}$ (labels) and compressed sequence $face^{QP}_{i+trk}$, considering several compression levels. Above quality 41 dB, accuracy is near 98 ∼ 99%.

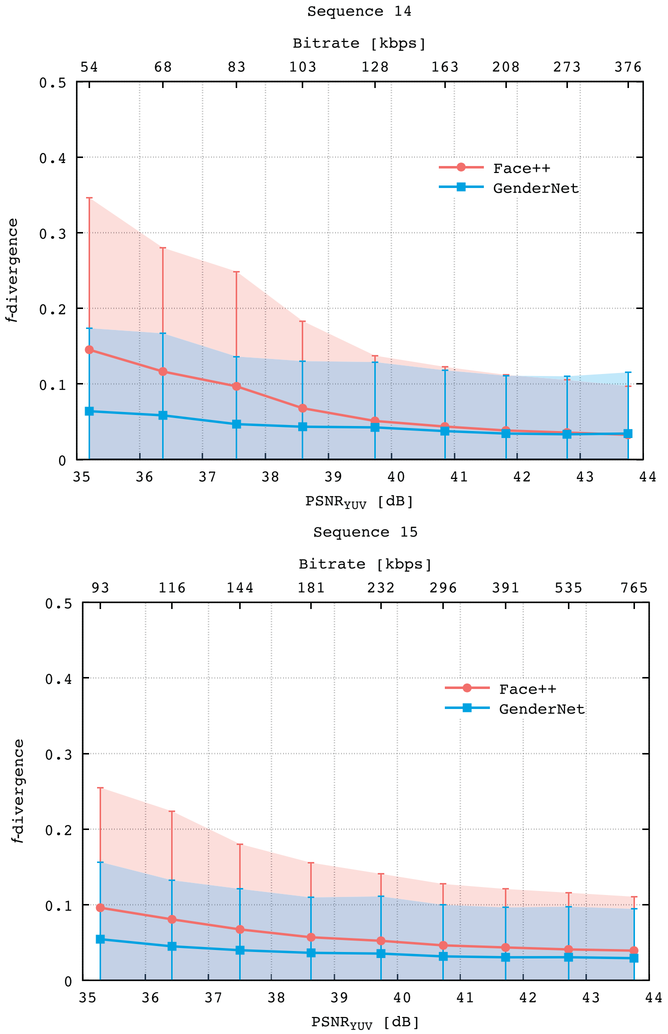


**Fig. 8.** Gender classification results $PSNR_{YUV}$ [dB] × $f-$divergence × bitrate [kbps]). Distances were computed between raw sequence $face_{i+trk}$ and compressed sequence $face^{QP}_{i+trk}$, considering several compression levels. Error bars indicate the standard deviation of the computed error. Above quality 41dB, $f-$divergence is lower than 0.05.

Please cite this article as: F.A. Andaló et al., TWM: A framework for creating highly compressible videos targeted to computer vision tasks, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.08.025

puted by a $f-$divergence function, more specifically, the normalized Hellinger distance. For two discrete probability distributions $M = (m_1, \dots, m_k)$ and $N = (n_1, \dots, n_k)$, their Hellinger distance is defined as

$$H(M,N) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{m_i} - \sqrt{n_i})^2}. \qquad (3)$$

Fig. 7 presents $PSNR_{YUV} \times$ accuracy curves for sequences 14 and 15. The plot shows accuracy values between gender classification performed in the raw sequence $face_{i+trk}$ (considered as labels) and in the respective compressed sequence $face^{QP}_{i+trk}$, considering different compression levels. For $PSNR_{YUV}$ greater than 41 dB, accuracy is near 98 ∼ 99%.

Fig. 8 presents $PSNR_{YUV} \times f-$divergence $\times$ bitrate [kbps] curves for sequences 14 and 15. They show the mean normalized Hellinger distance, and standard deviations, between gender classification performed in a raw sequence $face_{i+trk}$ and in the respective compressed sequence $face^{QP}_{i+trk}$, considering different compression levels. For $PSNR_{YUV}$ values greater than 41 dB, $f-$divergence is lower than 0.05, in a range from 0 (no divergence) to 1 (completely different distributions).

An interesting observation is that Face++ is more affected by compression than GenderNet. Because Face++ relies entirely on a compact face representation, compression artifacts, such as blocking and blur, may be preventing images with common attributes to be grouped in the same bin. Alternatively, GenderNet was trained specifically for gender classification, receiving as input face images originally in different resolutions. This may be the reason why it is more robust to compression.

## 5. Conclusions

We presented a framework – *Transmitting What Matters* (TWM) – for generating compressed videos aiming at a computer vision task. TWM creates videos containing only the information of interest for the desired task.

The solution is specially relevant in the increasingly common yet challenging scenarios that require the transmission and storage of UHD videos and where the simple reduction of spatial or temporal resolutions is not acceptable due to computer vision requirements.

Experimental results using 4K UHD videos and HEVC showed that TWM is very effective for video compression without harming the performance of the final task. The bitrate was reduced up to four times while the detection of landmarks was affected by only ∼ 1 pixel; and the accuracy of gender classification was affected by 1 ∼ 2 percentage points.

It is also safe to affirm that with TWM there is a significant gain in processing time, since it is only necessary to encode a much smaller resolution video, instead of a 4K video sequence.

We envision opportunities for future work by considering perceptual encoding and by evaluating TWM with other computer vision tasks (license plate recognition, for instance) and new datasets related to new scenarios.

## References

[1] F. Andaló, O. Penatti, V. Testoni, Transmitting what matters: Task-oriented video composition and compression, in: Conference on Graphics, Patterns and Images (SIBGRAPI), 2016, pp. 72–79.
[2] S. Avidan, Ensemble tracking, Trans. Pattern Anal. Mach.Intell. 29 (2) (2007) 261–271.
[3] Axis Communications, Pixel density, 2017, (http://www.axis.com/academy/pixel_count/pixel_density.htm). [Online; accessed 2017-02-26].
[4] A. Balk, M. Gerla, D. Maggiorini, M. Sanadidi, Adaptive video streaming: pre-encoded MPEG-4 with bandwidth scaling, Comput. Netw. 44 (4) (2004) 415–439.
[5] F. Bossen, Common test conditions and software reference configurations, Document JCTVC-F900 - Joint Collaborative Team on Video Coding (JCT-VC), 2011.
[6] F. Bossen, B. Bross, K. Suhring, D. Flynn, HEVC complexity and implementation analysis, Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1685–1696.
[7] G. Bradski, The OpenCV library, Dr. Dobb's J. Softw. Tools (2000).
[8] G.R. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technol. J. Q2 (1998).
[9] B. Bross, W. Han, G.J. Sullivan, J. Ohm, T. Wiegand, High efficiency video coding (HEVC) text specification draft 9, Document JCTVC-K1003 - ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), 2012.
[10] C. Bulla, C. Feldmann, M. Schink, Region of interest encoding in video conference systems, in: International Conferences on Advances in Multimedia, 2013, pp. 119–124.
[11] W.-S. Chu, F. De la Torre, J.F. Cohn, Selective transfer machine for personalized facial action unit detection, in: Conference on Computer Vision and Pattern Recognition, 2013, pp. 3515–3522.
[12] Cisco, Cisco visual networking index: forecast and methodology, 2013–2018, 2014.
[13] B. Ciubotaru, G. Ghinea, G.-M. Muntean, Subjective assessment of region of interest-aware adaptive multimedia streaming quality, IEEE Trans. Broadcasting 60 (1) (2014) 50–60.
[14] G. Correa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, Performance and computational complexity assessment of high-efficiency video encoders, Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1899–1909.
[15] D. DeForest, N. Lee, R. Pizzorni, C.P. Pace, Context based video encoding and decoding, 2013. US Patent App. 13/725,940.
[16] H. Fan, M. Yang, Z. Cao, Y. Jiang, Q. Yin, Learning compact face representation: packing a face into an int32, in: ACM Multimedia, 2014, pp. 933–936.
[17] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, Cascade object detection with deformable part models, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 2241–2248.
[18] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable part models are convolutional neural networks, in: Conference on Computer Vision and Pattern Recognition, 2015, pp. 437–446.
[19] D. Grois, O. Hadar, Region-of-interest: processing and coding techniques, in: Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools, 2013, pp. 126–155.
[20] R. Guntur, A. Shafiei, W.T. Ooi, Q.M.K. Ngo, System and method for enabling user control of live video stream(s), 2014. WO Patent App. PCT/SG2013/000,341.
[21] ITU-T/ISO/IEC JTC 1, Advanced video coding for generic audio-visual service, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC) (2003).
[22] JCT-VC, HEVC model (hm) repository, 2017, (https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/). [Online; accessed 2017-02-26].
[23] M. Karczewicz, Y. Ye, I. Chong, Rate distortion optimized quantization, Document VCEG AH21 - ITU T SG16/Q.6(2008).
[24] N.Q.M. Khiem, G. Ravindra, A. Carlier, W.T. Ooi, Supporting zoomable video streams with dynamic region-of-interest cropping, in: ACM Conference on Multimedia Systems, 2010, pp. 259–270.
[25] N.Q.M. Khiem, G. Ravindra, W.T. Ooi, Adaptive encoding of zoomable video streams based on user access pattern, Signal Process. Image Commun. 27 (4) (2012) 360–377.
[26] J.-S. Lee, T. Ebrahimi, Perceptual video compression: a survey, IEEE J. Sel. Top. Signal Process. 6 (6) (2012) 684–697.
[27] G. Levi, T. Hassner, Age and gender classification using convolutional neural networks, in: Conference on Computer Vision and Pattern Recognition Workshop, 2015, pp. 34–42.
[28] R. Lienhart, J. Maydt, An extended set of haar-like features for rapid object detection, in: International Conference on Image Processing, 1, 2002, pp. 900–903.
[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: European Conference on Computer Vision, 2016, pp. 21–37.
[30] H. Meuel, M. Munderloh, F. Kluger, J. Ostermann, Codec independent region of interest video coding using a joint pre-and postprocessing framework, in: IEEE International Conference on Multimedia and Expo (ICME), 2016, pp. 1–6.
[31] M. Murshed, A.R. Siddique, S. Islam, M. Ali, G. Lu, E. Villanueva, J. Brown, High quality region-of-interest coding for video conferencing based remote general practitioner training, in: International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED), 2013, pp. 240–245.
[32] J.-R. Ohm, G.J. Sullivan, H. Schwarz, T.K. Tan, T. Wiegand, Comparison of the coding efficiency of video coding standardsincluding high efficiency video coding (HEVC), Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1669–1684.
[33] H. Qiang, Z. Xiaoyun, G. Zhiyong, S. Jun, Analysis and optimization of x265 encoder, in: IEEE Visual Communications and Image Processing Conference, 2014, pp. 502–505.
[34] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real–time object detection, in: Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[35] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Neural Information Processing Systems, 2015, pp. 91–99.
[36] C. Rhodes, Systems and methods for adaptive transmission of data, 2012. US Patent 8,184,069.

Please cite this article as: F.A. Andaló et al., TWM: A framework for creating highly compressible videos targeted to computer vision tasks, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.08.025

[37] G.J. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1649–1668.

[38] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Conference on Computer Vision and Pattern Recognition, 1, 2001, pp. 511–518.

[39] T. Wiegand, G. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard, Trans. Circuits Syst. Video Technol. 13 (7) (2003) 560–576.

[40] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: Conference on Computer Vision and Pattern Recognition, 2013, pp. 532–539.

[41] M. Xu, X. Deng, S. Li, Z. Wang, Region-of-interest based conversational HEVC coding with hierarchical perception model of face, IEEE J. Sel. Top. Signal Process. 8 (3) (2014) 475–489.

[42] E. Zhou, H. Fan, Z. Cao, Y. Jiang, Q. Yin, Extensive facial landmark localization with coarse-to-fine convolutional network cascade, in: International Conference on Computer Vision Workshops, 2013, pp. 386–391.