CAN YOU FIND A FACE IN A HEVC BITSTREAM?

Saeed Ranjbar Alvar, Hyomin Choi, and Ivan V. Bajić

School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada

ABSTRACT

Finding faces in images is one of the most important tasks in computer vision, with applications in biometrics, surveillance, human-computer interaction, and other areas. In our earlier work, we demonstrated that it is possible to tell whether or not an image contains a face by only examining the HEVC syntax, without fully reconstructing the image. In the present work we move further in this direction by showing how to localize faces in HEVC-coded images, without full reconstruction. We also demonstrate the benefits that such approach can have in privacy-friendly face localization.

Index Terms— Face detection, face localization, HEVC, deep learning, privacy, scrambling

1. INTRODUCTION

Finding faces in images is one of the most important tasks in computer vision [1], with applications in biometrics, surveillance, human-computer interaction, and other areas. Recent advances in Deep Neural Networks (DNN) have broken new ground in this field [2, 3, 4, 5]; modern approaches achieve well over 90% true positive rate on popular benchmark datasets such as FDDB [6]. However, real-world deployment of these technologies has lagged behind research advances for several reasons. One is the computational resources needed to run advanced face detection on a large scale, especially on high-resolution images. Another reason is privacy concerns. If a vision system can find a face in the image, it might also be able to recognize that face. This idea makes many people uncomfortable. In this paper, we describe a way to find faces in images that requires less computation and offers higher privacy protection than conventional approaches.

In our previous work [7], we asked if it is possible to *tell a face from an HEVC bitstream*. That is to say, is it possible to distinguish images containing faces from those that do not, just from the High Efficiency Video Coding (HEVC) [8] syntax? We gave a constructive answer to that question by designing a Convolutional Neural Network (CNN)-based face detector for HEVC-coded images that performed equally well, on average, as a more conventional pixel-domain face detector that was also based on a CNN. We refer to this problem as face *detection*, in line with the common use of the term "detection" in statistical signal processing [9]. The ben-

efit of face detection directly from the bitstream is that full image reconstruction can be avoided, which saves over 60% of HEVC decoding time, on average, across various image resolutions [7].

In the present work, we extend this approach to face *localization* by showing that it is possible not only to detect faces, but also find where they are in HEVC-coded images without full image reconstruction. We also demonstrate the potential of this approach in privacy protection. Privacy-friendly visual analytics are becoming increasingly important with the growth of public awareness of the widespread use of private data for commercial (and sometimes illegal) purposes. A recent proposal on this topic [10] advocates modifying the face region in an image in order to hinder face detection and thereby also hinder face recognition. Our approach is different. We can scramble transform coefficients over the entire image, without knowing beforehand where the faces are. Due to scrambling, conventional face detectors' performance is hindered, similar to [10]. But because our face localization relies on HEVC syntax and not on pixel values, our method can still find faces in scrambled images, without being able to recognize them. Hence, we achieve the benefit of enabling simple analytics (such as counting people, estimating their location, etc.) in a privacy-friendly manner, without the need for complex computer vision processing (such as face detection) prior to encoding.

The paper is organized as follows. Section 2 presents the proposed face localization method, including feature creation from HEVC syntax. The scrambling method used to demonstrate privacy-friendly properties of the proposed face localization is briefly described in 3. Results are presented in Section 4 followed by conclusions in Section 5.

2. PROPOSED METHOD

Multimedia data is generally only available in compressed form. Conventional face localization implicitly requires full pixel reconstruction from the compressed data. An example is shown in Fig. 1(a), where the input is an HEVC-compressed image. By contrast, the proposed approach only requires HEVC entropy decoding to reconstruct various syntax elements that will be used as features for face localization. This way, a number of stages in the decoding process can be avoided: inverse quantization, inverse transforms, prediction, and pixel reconstruction.
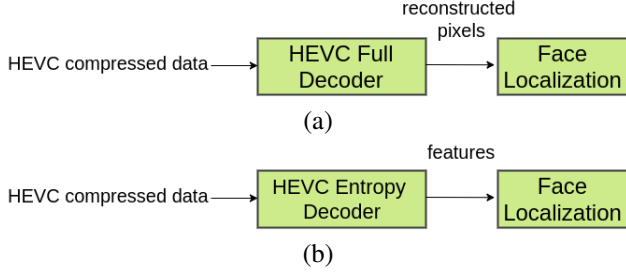
Fig. 1. (a) Conventional face localization; (b) proposed face localization.

In order to perform face localization, we construct a *feature image* from HEVC syntax elements. Specifically, during HEVC entropy decoding, the Intra Prediction Mode (IPM), Prediction Unit Size (PUS) and Bin Number (BN) are reported for each Prediction Unit (PU). We construct a 3-channel feature image based on these parameters by mapping each one to the range 0-255 and copying it into the corresponding location in the feature image.

IPM values are integer numbers in the range 0-34 [8]. These are linearly mapped and rounded to integers in 0-255 to create the IPM channel. PUS values can take one of the values $\{4, 8, 16, 32\}$; they are mapped to $\{0, 85, 170, 255\}$, respectively, to create the PUS channel. BN values vary depending on the number of bits used in a given PU. For the BN channel, the minimum and maximum BN values in the image are found, and then each BN value is linearly mapped and rounded to integers in the range 0-255. An example is shown in Fig. 2. Further examples can be found in our earlier work [7].

We build our face localization upon the state-of-the-art object detector called You Only Look Once (YOLO) [11]. YOLO is based on a DNN that can find, in a single pass, various objects in the input image along with their bounding boxes. The network is trained to do both object localization and classification using a loss function that includes both
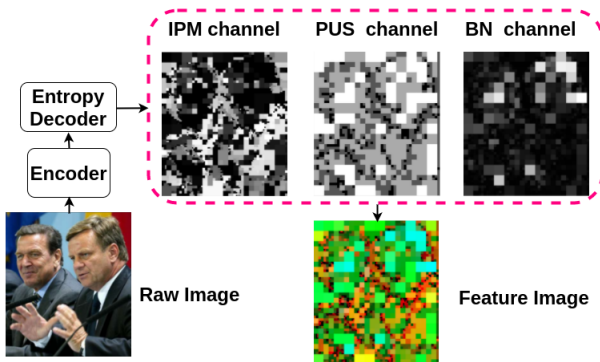
bounding box error and class error terms [11]:

$$
\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2 \right]
$$

$$
+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w_i}})^2 + (\sqrt{h_i} - \sqrt{\hat{h_i}})^2 \right]
$$

$$
+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C_i})^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C_i})^2
$$

$$
+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p_i}(c))^2 \quad (1)
$$

where (x_i, y_i) is the center of the ground truth bounding box, w_i and h_i are its width and height, $(\hat{x_i}, \hat{y_i})$ is the center of the predicted bounding box whose width and height are $\hat{w_i}$ and $\hat{h_i}$, respectively. C_i and $\hat{C_i}$ are the groud truth and predicted confidence scores corresponding to cell i, $p_i(c)$ and $\hat{p_i}(c)$ are the ground truth and predicted conditional probabilities for the object class c in cell i, $\mathbb{1}_{ij}^{obj}$ is equal to 1 if the j-th bounding box in cell i is responsible for prediction (i,e. box j has the largest Intersection-over-Union, IoU, among all boxes in cell i), and $\mathbb{1}_{ij}^{noobj} = 1 - \mathbb{1}_{ij}^{obj}$. The scaling factors used are $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$.

The YOLO architecture can be trained to detect different object classes. However, since we are interested in faces only, we used its recent version YOLO9000 [12] and modified it to detect one object class - faces. The modified network produces a map of 13×13 cells, with each cell returning 5 candidate bounding boxes and a confidence score for each box. The confidence score represents how confident the model is that the corresponding box contains a face. The confidence values can be thresholded to make final predictions: boxes with high enough confidence are predicted to contain faces, and others are ignored. In order to evaluate such a system, a range of thresholds on confidence values is used, and both the prediction accuracy and localization accuracy are taken into account [6]. The complete evaluation is described in detail in Section 4, along with model training.

3. PRIVACY FRIENDLINESS

Since our proposed method does not rely on pixel values, it opens up the opportunities for privacy-friendly face localization. To demonstrate this, we adapt the scrambling methods from [13] to HEVC. The scrambling schemes in [13] were developed to scramble the Region Of Interest (ROI) in H.264/AVC-based video coding. Two basic schemes were proposed: random sign inversion of AC transform coefficients and random permutation of AC coefficients based on the Knuth shuffle [14].

We adapt the methods from [13] to HEVC and apply them across the entire image. Since the transform coefficients

Fig. 2. Creating the feature image.

are computed in each Transform Unit (TU), we apply random sign inversion and random permutation within each TU. These changes can be undone by an authorized decoder that knows the (pseudo)random sequences involved in sign inversion and coefficient permutation. An unauthorized decoder will only be able to reconstruct scrambled images.

The above changes have a significant effect on the final reconstructed images, rendering conventional face localization (and presumably face recognition) useless. However, they have only a minor effect on our feature images, hence our face localization is largely unaffected by such scrambling. Specifically, the IPM and PUS channel remain unaffected. Random permutation and sign changes do increase BN values over the whole image. But because BN channel is produced by normalizing BN values using the minimum and maximum BN values in the image, the net effect on the BN channel is minor. We measured the mean of absolute intensity difference in the BN channel between scrambled and non-scrambled images, and found that the difference is only 0.1, averaged over all the training images.

An example of the effects of scrambling is shown in Fig. 3 for four quantization parameter (QP) values. One can see that scrambling has a major effect on the final reconstructed images, but not on our feature images. Therefore, using our approach, one would still be able to detect and localize faces in the scrambled bitstreams, but would not be able to reveal their identity.

4. EXPERIMENTAL RESULTS

4.1. Experimental Setting

Face Detection Dataset and Benchmark (FDDB) [6] is used for evaluating the performance of the proposed face localization method. FDDB includes 2845 images with 5171 annotated faces. FDDB comes with a standard evaluation method that allows comparison among various face localization methods. The evaluation is based on the Intersection-over-Union (IoU) with the ground truth; if IoU is larger than 0.5, the detection is considered a True Positive (TP) [6].

HEVC reference software HM16.5 [15] is used for intra coding the images using the configurations in [16]. The QP values used in the evaluation are $\{22, 27, 32, 37\}$, which covers the range typically used in practice.

Our CNN model is based on the Darknet framework [17]. The training data were the feature images extracted from the (non-scrambled) HEVC bitstream obtained with QP $= 32$. Stochastic gradient descent with learning rate 10^{-3}, momentum of 0.9, and weight decay of 5×10^{-4} is used for training. The training batch size was set to 64, and the training was terminated after 10k epochs. Training was initialized with YOLO900 model weights obtained on ImageNet [18]. For testing, non-maximum suppression [19] on the outputs with the threshold of 0.4 was employed.

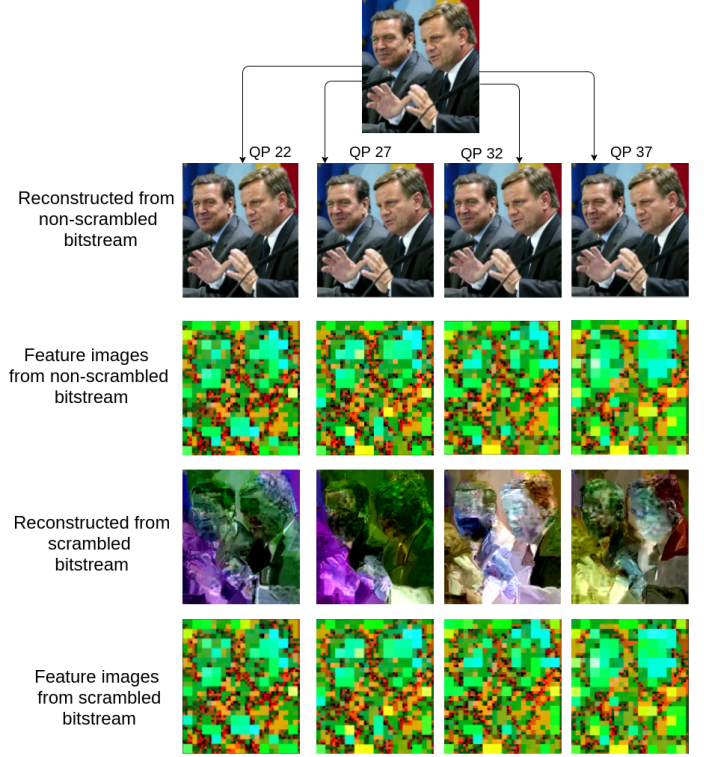Fig. 4 shows the performance of several notable face

Fig. 3. An example of feature images and fully reconstructed images for the input encoded with no-scramling case and with scrambling for 4 different QP values.

localization methods on FDDB, including TinyFace [5], MTCNN [20], Faceness [3], Hyperface [4], CascadeCNN [2], PICO [21] and Viola-Jones [22]. We have chosen TinyFace as the benchmark to compare against, since it represents the current state-of-the-art.

4.2. Face localization results

The test data consists of FDDB images encoded in the HEVC intra mode using QP $\in \{22, 27, 32, 37\}$, as mentioned before. We used both scrambled and non-scrambled bitstreams to investigate the effect of scrambling on face localization. The input to our model were feature images generated as described in Section 2. The input to TinyFace were the fully-decoded images. The FDDB accuracy results are shown in Fig. 5.

As seen in Fig. 5, the benchmark model achieves over 97% TP rate with 1,000 False Positives (FP) on non-scrambled images. The effect of HEVC compression on this model is minor in the range of QP values we used. But on scrambled images, its TP rate drops to 50% or less. This shows the significant effect that scrambling has on face localization, as we could have expected from Fig. 3.

Meanwhile, our model achieves roughly the same performance on both scrambled and non-scrambled images, as discussed in Section 3. For our model, QP has a larger influence than scrambling, because feature images change with QP
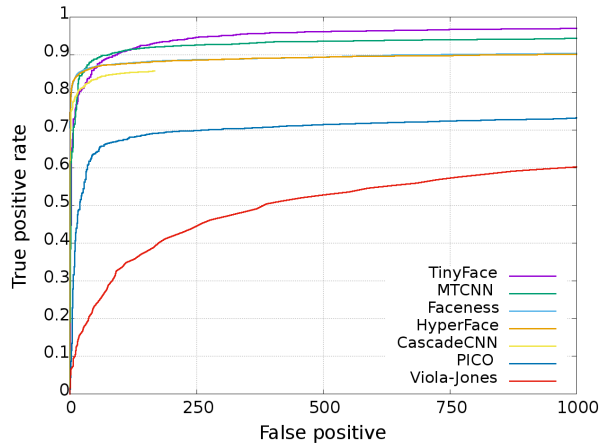
Fig. 4. Several notable face localization models on FDDB, including the chosen benchmark.

Fig. 5. At FP=1000, TinyFace [5] has over 97% TP rate on raw images as well as images decoded from non-scrambled bitstreams. But on images decoded from scrambled bitstreams, its TP rate drops to 50% or less. Meanwhile, our method has consistent performance on both scrambled and non-scrambled images. The different colors correspond to different QP values.

(Fig. 3). While we could have trained an ensemble of models, each on a different QP, for possibly better performance, we opted to use a single model trained on QP = 32 in order to examine its robustness to QP variation. Indeed, even when tested on data produced by different QPs, our model still outperforms the benchmark significantly on privacy-friendly scrambled images. It is worth noting that neither the proposed method nor the benchmark were trained on scrambled images.

Fig. 6 shows a few examples of face localization. In each row, the first image is the original, the second image is decoded from the scrambled bitstream and the last image is the feature image extracted from the scrambled bitstream. Tiny-Face [5] cannot find faces in scrambled images, but our model finds faces in feature images extracted from scrambled bitstreams.

5. CONCLUSION

In this paper we presented a method for finding faces in HEVC-coded images. Our approach takes advantage of HEVC syntax, rather than the actual pixel values, which allows it to find faces even in scrambled images. This opens up the possibilities for privacy-friendly visual analytics, such as counting people without revealing their identity. Unlike existing approaches, our methodology does not require running complex computer vision engines prior to encoding.

The proposed method runs on HEVC intra-coded bitstreams, primarily because still images are the common setting for evaluating face detectors/localizers [6]. But the methodology lends itself to extension to video as well. In the case of HEVC-coded video, faces could be detected and localized in I-frames, then tracked through the inter-coded frames using motion vectors, for example using [23].
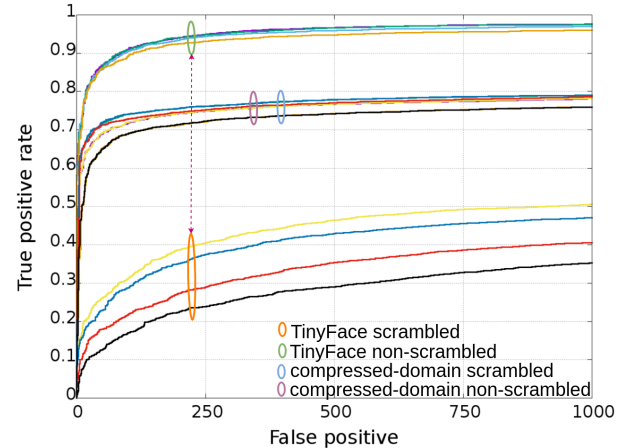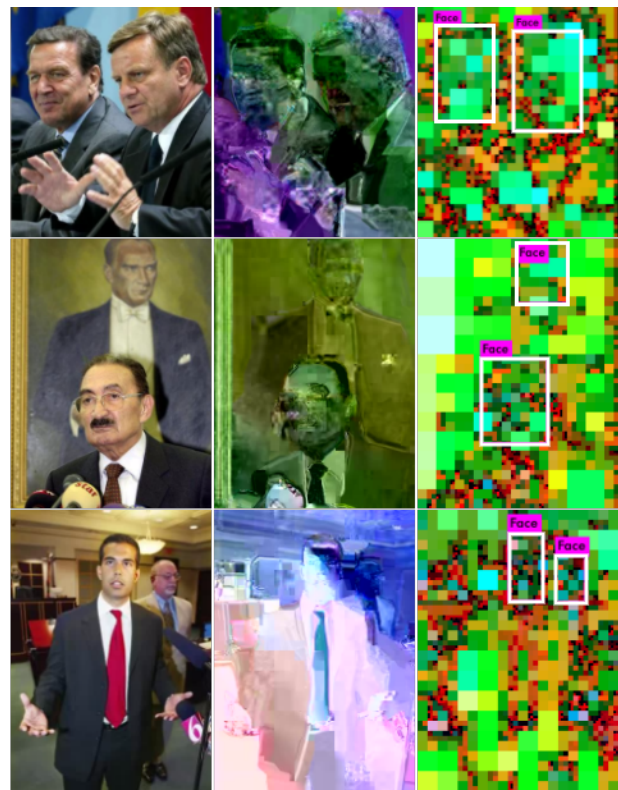

Fig. 6. TinyFace [5] cannot find faces in the scrambled images, but our model finds faces in feature images extracted from scrambled bitstreams.

6. REFERENCES

[1] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," *Comput. Vis. Image Und.*, vol. 138, pp. 1–24, Sep. 2015.

[2] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE CVPR'15*, 2015, pp. 5325–5334.

[3] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proc. IEEE ICCV'15*, 2015, pp. 3676–3684.

[4] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *arXiv preprint arXiv:1603.01249*, 2016.

[5] P. Hu and D. Ramanan, "Finding tiny faces," in *Proc. IEEE CVPR'17*, Jul. 2017.

[6] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Tech. Rep. UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts, Amherst, 2010.

[7] S. R. Alvar, H. Choi, and I. V. Bajić, "Can you tell a face from a HEVC bitstream?," in *Proc. IEEE International Conference on Multimedia Information Processing and Retrieval*, 2018, to appear.

[8] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, vol. II, Prentice Hall, 1998.

[10] P. Chriskos, J. Munro, V. Mygdalis, and I. Pitas, "Face detection hindering," in *IEEE GlobalSIP*, Nov. 2017, to appear.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE CVPR'16*, Jun. 2016, pp. 779–788.

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. IEEE CVPR'17*, Jul. 2017.

[13] F. Dufaux and T. Ebrahimi, "H.264/AVC video scrambling for privacy protection," in *Proc. IEEE ICIP'08*, Oct. 2008, pp. 1688–1691.

[14] D. Knuth, "Seminumerical algorithms," in *The Art of Computer Programming*, pp. 139–140. AddisonWesley, 1969.

[15] "HEVC reference software (HM 16.15)," https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.15, Accessed: 2017-05-27.

[16] F. Bossen, "Common HM test conditions and software reference configurations," in *ISO/IEC JTC1/SC29 WG11 m28412, JCTVC-L1100*, Jan. 2013.

[17] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[20] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.

[21] N. Markus, M. Frljak, I. S. Pandzic, J. Ahlberg, and R. Forchheimer, "Object detection with pixel intensity comparisons organized in decision trees," in *arXiv preprint arXiv:1305.4537*, 2014.

[22] P. Viola and M.J Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[23] S. H. Khatoonabadi and I. V. Bajić, "Video object tracking in the compressed domain using spatio-temporal Markov random fields," *IEEE Trans. Image Processing*, vol. 22, no. 1, pp. 300–313, Jan. 2013.