The International Journal of Engineering and Science (IJES)
|| Volume || 8 || Issue || 8 Series II || Pages || PP 62-69 || 2019 ||
ISSN (e): 2319 – 1813 ISSN (p): 23-19 – 1805

Fast Face Detection System Based On HEVC Bitstream

Tien-Yang Hsu, Tung-Hung Hsieh, Han-Yu Huang, Chou-Chen Wang
Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan
Corresponding Author: Chou-Chen Wang

----------ABSTRACT----------
High efficiency video coding (HEVC) is the latest coding standard for ultrahigh definition (UHD) video applications. Face detection is one of the most important tasks for intelligent video surveillance (IVS) system. It is a fact that an accurate face detection facilitates subsequent face recognition and security system. Since most of existing computer vision methods used in IVS systems operate in a pixel domain, this means that IVS need to analyze video content after fully decoding HEVC bitstreams or compressed files. However, the new generation of IVS system will employ UHD, this results in a high computational load for intelligent video analytics in the pixel domain. Therefore, it is difficult to achieve a real-time IVS system. To solve time-consuming problem, we perform partially decode to obtain some significant information in entropy decoder (ED) by skipping subsequent decoding stages to reduce decoding complexity and save decoding time. In order to speed up face detection in HEVC compression domain, we firstly only decode NALU header and ED module to obtain the most information values including intra prediction mode (IPM), transform unit size (TUS) and bit number (BN).Secondly, we linearly map these values to a range 0~255 of gray levels and copy them into the corresponding location in the image. Thirdly, we create three feature images (ITB) that can be visualized and processed in a similar way to conventional 3-channel (RGB) images. Finally, we employ a deep learning for convolutional neural network (CNN) to finish face detection using our proposed ITB image based on HEVC bitstream. Simulation results show that the proposed ITB and RGB images can achieve F1-measure about $F1_{ITB} = 0.98$ and $F1_{RGB} = 0.99$ from image database (LFW and LSVRC), respectively, when image size is 128×128 and QP = 22.In the same experimental conditions, we can obtain $F1_{ITB} = 0.96$ and $F1_{RGB} = 0.98$ when using the real surveillance images from our laboratory to perform training and testing. It is clear that the proposed ITB image can not only finish a fast face detection but also achieve a very close value of F1-measure to RGB image.
KEYWORDS–*Face detection, HEVC, deep learning, convolutional neural network.*

Date of Submission: 22-08-2019 Date of acceptance: 05-09-2019

I. INTRODUCTION

High efficiency video coding (HEVC) is the more recent video coding standard for ultrahigh definition (UHD) video applications [1-2]. On the other hand, intelligent video analytics (IVA) including moving object detection, segmentation, classification and recognition are the most important tasks for intelligent video surveillance (IVS) system. It is a fact that an accurate detection of moving objects facilitates subsequent object recognition. However, since most of existing computer vision methods used in commercial IVS systems operate in a pixel domain, this means that IVS need to analyze video after fully decoding HEVC bitstreams or compressed files. Because the new generation of IVS system will employ UHD images, resulting in a high computational load for intelligent video analytics in the pixel domain. It is difficult to achieve a real-time IVS system. In this paper, we observe that the output of the entropy decoding module in HEVC decoder takes less than 20% of the overall decoding time. This implies that we can partially decode to obtain some significant information in entropy decoder (ED) by skipping subsequent decoding stages to save decoding time. Therefore, the most important contribution of the proposed paper is to develop a faster IVS in HEVC compressed domain.

Recently, there are many convolutional neural networks (CNN) based techniques adopted to perform face detection and/or localization. CNN-based recognition technology has achieved great success in face detection fields [3-5]. Through the deep training process using large datasets, the weights of the CNNs used in these methods are adjusted to find features that can effectively discriminate between the face and non-face image patches. Face detection is one of the most important tasks for IVS system. It is a fact that an accurate face detection facilitates subsequent face recognition and security system. In general, the IVS need to analyze video content after fully decoding HEVC bitstreams or compressed files since most of existing face detection methods used in a pixel domain. This will result in a high computational load for intelligent video analytics in the pixel domain for UHD video. In order to solve time-consuming problem, we perform partially decoding to obtain some significant information in ED to reduce decoding complexity. And then, we check whether it is possible to detect a face in a HEVC bitstream without full decoding image. Specifically, we look at the output of the HEVC entropy decoder

in intra-coded images in this paper. In the same topic, Alvar et al. proposed the first studying to detect a face quickly using HEVC compressed files [5]. However, their method meets a problem of accuracy drop due to simply mapping these features to gray levels.

In order to further accelerate face detection, we firstly only decode HEVC header and ED module to obtain the most important information including intra prediction mode(IPM), transform unit size (TUS) and bit number (BN).Secondly, we linearly map these values to a range 0~255 of gray levels and copy them into the corresponding location in the image. Thirdly, we create three feature images (ITB) that can be visualized and processed in a similar way to conventional 3-channel (RGB) images. Finally, we employ a deep CNN to finish face detection using the proposed ITB image.

II. BRIEF OVERVIEW OF HEVC DECODER

HEVC decoding procedure is shown in Fig.1.HEVC decoder mainly consists of many modules including network adaptation layer (NAL) header decoding, type decoding, entropy decoding (ED), inverse quantization and inverse integer cosine transform (IQ/IT), intra frame prediction (IFP), motion compensation (MC), sample adaptive offset(SAO)and de-blocking filter (DF) which integrated loop picture filter (IPF), and other modules. The high-level decoding syntax architecture of HEVC is similar to that of H.264 [6-7]. The two layer structures of NAL and video coded layer (VCL) have been reserved. The sequence parameter set (SPS) and picture parameter set (PPS) structures have been completed with a new video parameter set (VPS) structure. On the other hand, HEVC bitstream is an ordered sequence of the syntax elements. Each syntax element is placed into a logical packet called a NAL unit (NALU).VPS, SPS and PPS contain general video parameters. They provide a robust mechanism for conveying data that are essential to the decoding process. They can be either a part of bitstream or can be stored separately. Therefore, NAL header decoding is simple processing because it only take a look up table and get the corresponding parameters.

A series of statistical analysis is conducted to reveal the decoding complexity for intra frame. Four Class C video sequences including different frame rate and scenarios are first tested to analyze the consuming time of each decoder module of HEVC based on HM16.7 [8]. And, all intra (AI) encoder configurations are adopted in HM16.7 and quantization parameters (QP) are set to 32.Table 1 gives the average HEVC@ BP (baseline profile) complexity information of the decoding process after time-consuming statistics for every decoding module. From Table 1, we can find that the most consuming processes of decoder are IFP, ED, and IPF modules, separately. The MC module didn't work due to AI configuration. It is evident from Table 1 that ED module occupies only 20% of the overall decoding time, on average. As a result, we can perform partially decoding modules including NAL header, type decoding and ED decoding to obtain some significant information by skipping subsequent decoding stages to reduce decoding complexity and speedup face detection time. Using the output of the HEVC entropy decoder, we can train a simple shallow CNN to detect faces based on several HEVC syntax features.

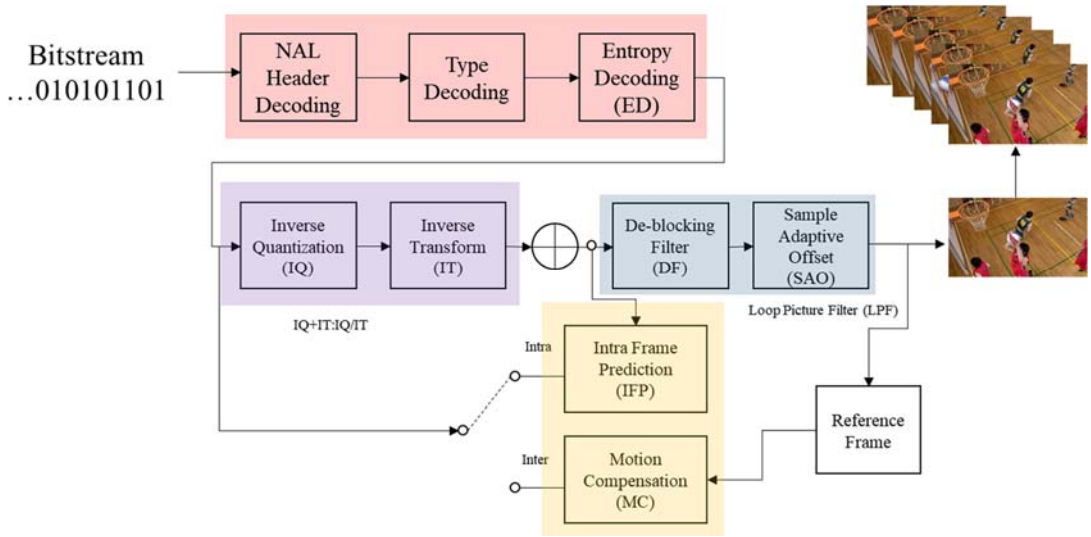


Fig. 1 HEVC decoding modules.

Table 1. Complexity of HEVC decoder.

Sequence		ED	IQ/IT	IFP	MC	LPF	Others
AI	BasketballDrill	16%	18%	35%	0%	19%	12%
	BQMall	19%	17%	32%	0%	18%	14%
	PartyScene	26%	12%	29%	0%	14%	20%
	RaceHorses	21%	15%	32%	0%	17%	16%
	Average	**20%**	**15%**	**32%**	**0%**	**17%**	**15%**

III. PROPOSED METHOD

To speedup face detection in HEVC bitstream, Alvar et al. firstly proposed a partially-decoded HEVC bitstream to detect a face quickly [5]. They reported create the intra prediction mode (IPM), prediction unit size (PUS) and bit number (BN) for each prediction unit (PU) during HEVC entropy decoding. They map these values to a range 0-255 and then copy them into the corresponding location in the image. In Alvar's method, IPM values in the range 0-34 are linearly mapped to integers 0-255, PUS values in {4, 8, 16, 32} are respectively mapped to {0, 85, 170, 255}, and BN in each PU are linearly quantized to integers 0-255 according to the minimum and maximum BN in the image.

Although Alvar's method can speed up processing of face detection, they encounter some problems of accuracy drop. These issues are described as follows.

(1) Due to using simple linear mapping rule, they didn't consider the symmetry of IPM modes resulting in the gray levels distribution of feature image becoming unobvious.

(2) Since PUS varies from 64×64 to 8×8 in each coding unit (CU), the contours of feature image become blurred in the larger CU size.

(3) The contours of BN feature image will be broken when the values of BN in CU size 32×32 and 64×64become larger.

3.1 Constructing feature image

In order to further improve the face detection performance in HEVC bitstream, we take the important information including intra prediction mode (IPM), transform unit size (TUS) and bit number (BN) as feature channel, and linearly map these values to gray levels according to their characteristics, respectively. And then, we create 3-channel ITB image that can be visualized and processed in a similar way to conventional 3-channel RGB images. Finally, we employ a shallow CNN to finish face detection using our proposed ITB image based on HEVC bitstream.

From detail observation of IPM in HEVC, we can find that these 35 modes exist a symmetrical relationship, as shown in Fig. 2. With mode 18 (-32°) as the central line, the mode 17 (-26°) is symmetric to mode 19 (-26°), and the mode 16 (-21°) is symmetric to mode (-21°), and so on. In order to combat the problem of gray level distribution becoming unobvious, we linearly mapped these 18 symmetric modes to integers 0-255 as follows.

$$\mathrm{IPB} = \frac{255 \times \mathrm{mode}}{18} \tag{1}$$

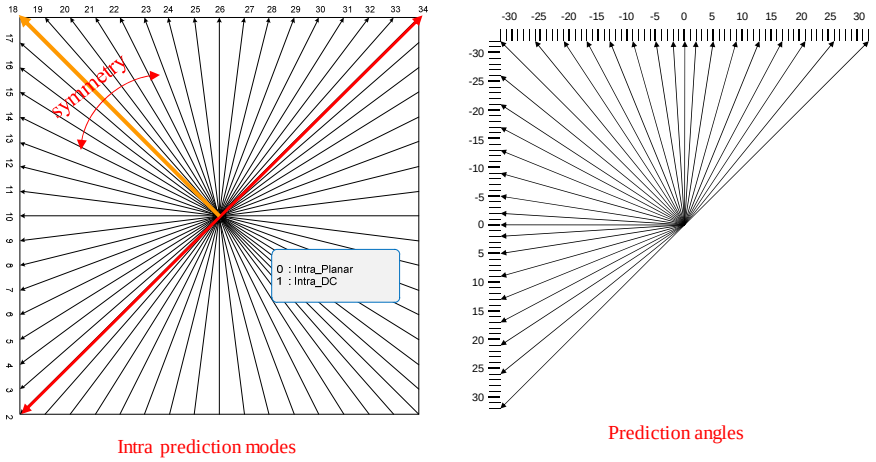
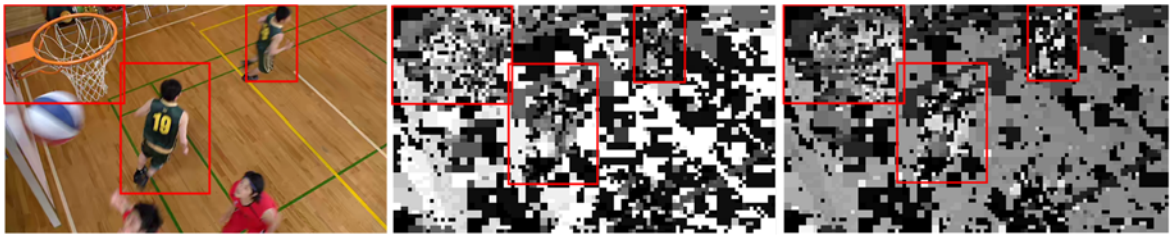


Fig. 2 Symmetry of intra frame modes.

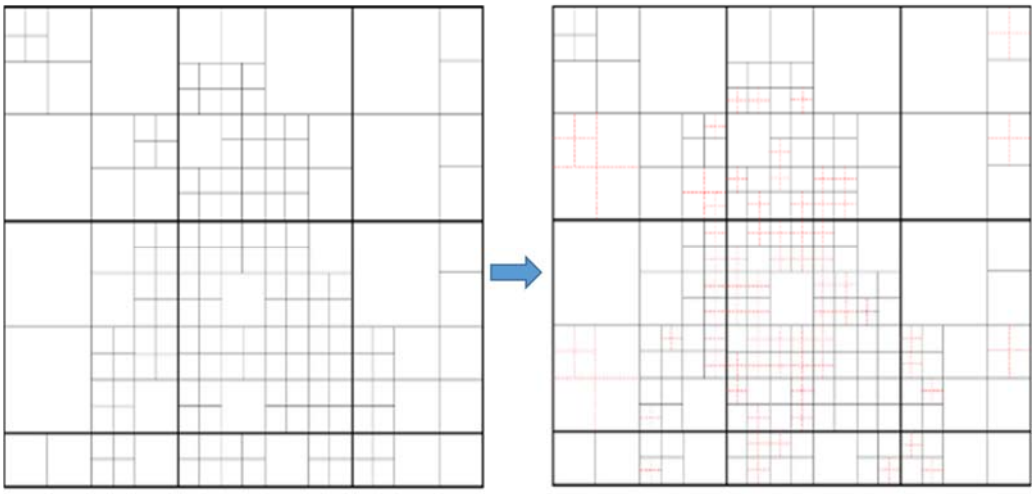
The proposed IPM feature image has a more obvious grey level distribution using a large quantization interval. In order to show the improvement of the proposed method, Figure 3 shows the decoded RGB image, the proposed IPM image and Alvar's IPM image, respectively. From Fig. 3, we can observe the proposed I channel image indeed reveals clearer grey level for feature image.



(a) (b) (c)

Fig. 3 Performance comparisons of I channel. (a) HEVC image (b) Proposed method (c) Alvar's method.

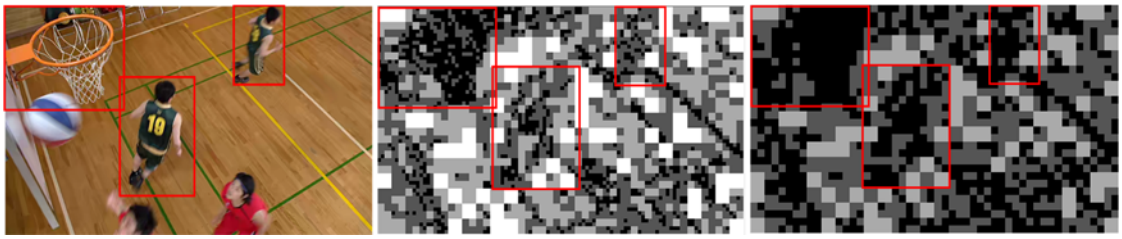
It is a big issue using PUS channel by Alvar's method because the larger CU size blurs the contours of feature image. To overcome this problem, we propose the TUS channel to substitute for PUS. Since the TUS varies from 32×32 to 4×4 pixels, we can obtain a more detailed contour of feature image by using a smaller size. Figure 3 shows the comparison of image contour between PUS and TUS. The further division for TUS is denoted as red line in Fig. 4. Therefore, TUS values is {32, 16, 8, 4} and they are linearly mapped to {0, 85, 170, 255}. It is obvious that the TUS contains more information than those of PUS, so that it can demonstrate sharp contours.



(a) (b)

Fig. 4 Comparison of image contour. (a) PUS (b) TUS.

In order to show the improvement of the proposed method, Figure 5 shows the decoded RGB image, the proposed TUS image and Alvar's PUS image, respectively. From Fig. 5, we can observe the proposed T channel image indeed reveals obvious contours of feature image.



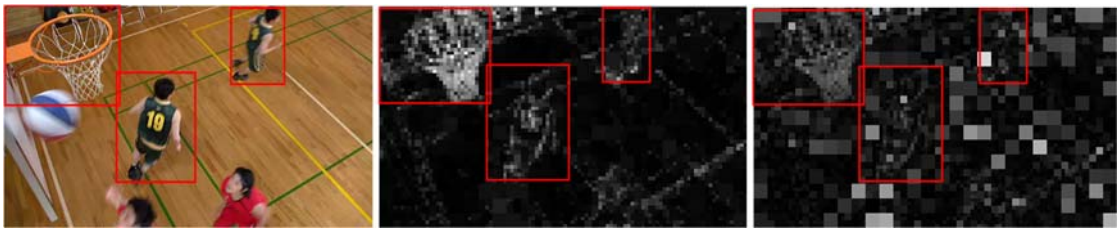
(a) (b) (c)

Fig. 5 Performance comparisons of T channel. (a) HEVC image (b) proposed method (c) Alvar's method.

Another problem for BN channel by Alvar's method is that contours of BN image are broken when CU sizes are 32×32 and 64×64. The reason of broken contour is when the size of the PU is larger and the value of BN becomes larger. This results in many white image feature, which gray level equals 255, in BN channel. To solve the problem of broken contour, we propose a smooth mapping method using an average by the numbers of 8×8 in each split CU. Firstly, we take an average of BN (BN_{ave}) into the size of 8×8.Secondly, the values of minimum BN (BN_{min}) and maximum BN (BN_{max}) in the feature image are found, and then each BN values is linearly mapped and rounded to integers in the range 0-255. The proposed BN values is linearly mapped as follows.

$$BN = \frac{255 \times BN_{ave}}{BN_{max} - BN_{min}} \tag{2}$$

In order to show the improvement of the proposed method, Figure 6 shows the decoded RGB image, the proposed BN image and Alvar's BN image, respectively. From Fig. 6, we can observe the proposed B channel image indeed maintain more complete contours of feature image.



(a) (b) (c)

Fig. 6 Performance comparisons of B channel. (a) HEVC image (b) proposed method (c) Alvar's method.

And then, we create three proposed channels including IPM, TUS and BN as an ITB images that can be visualized and processed in a similar way to conventional RGB images. In order to further test our method, we decided to extend the feature channels and the feature images to the full size of the input patch to facilitate easier visualization and compare with pixel-domain face detection. Figure 7 shows an example of 3-channel feature images and constructing feature image for the input image encoded using$QP = 32$. As seen in the figure, we create 3-channel (ITB) images that can be visualized and processed in a similar way to conventional 3-channel (RGB) images. Note that feature images change when the QP value changes, this is because encoding decisions of IPM, TUS and BN depend on QPs. From the comparisons of ITB and HEVC decoded images, we find the ITB images change more than the resulting fully reconstructed images. Therefore, we can expect that face detection from feature images may be more challenging than conventional pixel domain detection. Finally, we employ a simple CNN to perform face detection using the proposed ITB image based on HEVC bitstream.

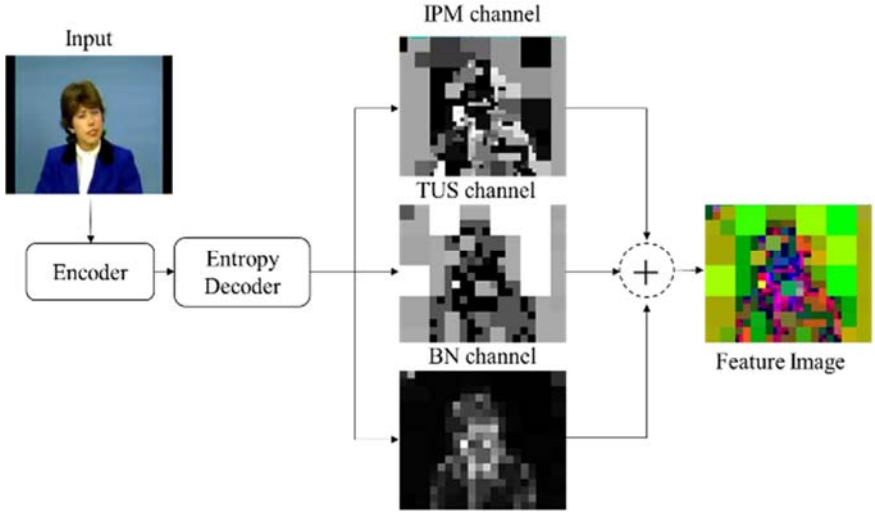


Fig. 7 Constructing feature image.

3.2 CNN for face detection

In the proposed method, we select a shallow CNN architecture to perform a fast face detection from HEVC feature images in the following work. We first take a number of 128×128 image patches with and without faces in datasets to perform HEVC encoder using QP = 32 and created ITB images from them. A very simple network implemented in Keras1 with Tensorflow backend is adopted. The CNN network comprises one convolutional layer with one 5×5×3filter, stride of 4, and one fully-connected layer with one unit connected to the output whose value is used for face/non-face decisions. The proposed CNN was trained using stochastic gradient descent (SGD) with the learning rate of 10^{-4}. The number of units in the fully connected layer is increased and stopped at 500 where the accuracy saturated. Then, we increase the number of filters in the convolutional layer and observe that the accuracy kept increasing until the number of filters reached 100.

We design the LeNet-5architecture of CNN model according to achieving higher efficiency, as shown in Table 2. Figure 8 shows the final CNN architecture with two convolutional layers, two fully-connected layers and a softmax classifier to perform the proposed face detection. Rectified linear unit (ReLU) functions are used for activation in convolutional layers and sigmoid is used in the output layer.

Table 2. The architecture of the proposed CNN model.

Layer	Layer 1		Layer 2	
Type	Conv.	ReLU Max(0, x)	Conv.	ReLU Max(0, x)
Filter size	5×5		3×3	
Filters	100		100	
Strides	(1,1)		(1,1)	
Padding	Same		Same	

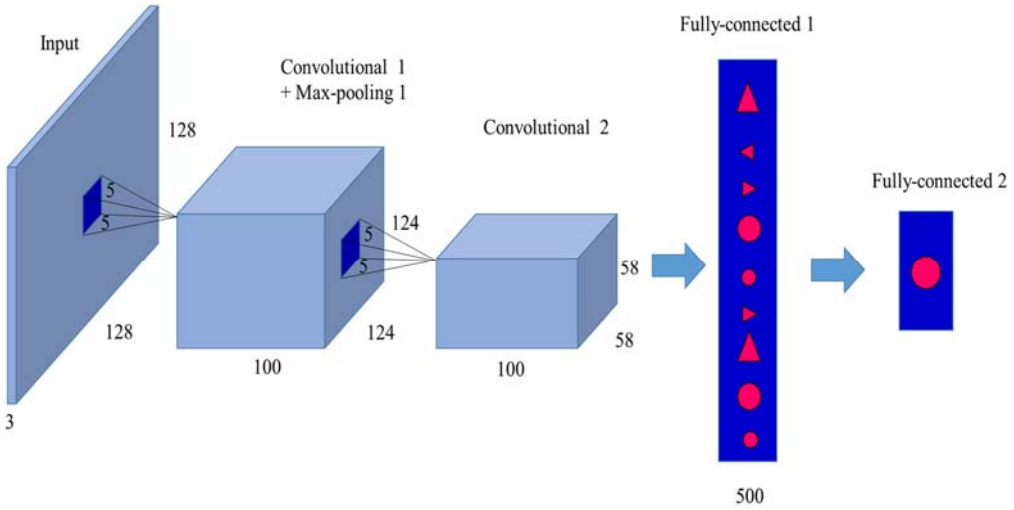


Fig. 8 Proposed CNN architecture of face detection.

IV. EXPERIMENTAL RESULTS

In order to compare the performance of face detection between the ITB feature image in compressed domain and the RGB image in pixel domain, we fed ITB and RGB images to a simple swallow CNN to determine whether or not the image contains a face, respectively. In our experiments, 6,000 face images were taken from the Labelled Faces in the Wild (LFW) dataset [9-10], and 6,000 non-face images were taken from the Large Scale Visual Recognition Challenge (LSVRC) [11-12]. In addition, some real surveillance images from our laboratory are also used as training and testing dataset.

In this paper, we have implemented HEVC video bitstream in HM16.7 [8] encoder test model, the encoding configuration is AI P with QP = 22, 32, 42. Simulations are conducted on a desktop with (1) Intel (R) Core (TW) CPU i7-3350P @ 3.60 GHz, (2) NVIDIA GeForce GTX 1060-3GB, (3) Window 10-64bit, (4) Visual Studio 2013, (5) Python 3.6.4. The performance of the trained CNN models based on compression-domain data is evaluated in terms of Accuracy (Acc), Precision, Recall and F1-measure.These evaluations are computed from true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) as following:

$$\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FP}} \tag{5}$$

$$\text{F1} = 2\frac{\text{Percision}\times\text{Recall}}{\text{Percision+Recall}} \tag{6}$$

Table 3 and Table4 show the performance of face detection obtained by RGB images, Alvar's IPB images [4] and proposed ITB images when performing Hallway images and real surveillance images. Simulation results show that the proposed method can achieve a faster face detection. From Tables 3 and 4, we can observe that the proposed method and Alvar's method both can accelerate the processing of face detection about 6 times when compared with RGB in pixel domain. In addition, we also can find that the detection accuracy of the proposed method is higher than those of Alvar's method. Note that feature images change when the QP value changes, this is because encoding decisions of IPM, TUS and BN depend on QPs. The contours of ITB feature image become less obvious when the values of QP increase. Therefore, it lead to the accuracy drop as higher QP value.

On the other hand, Table 4 shows the performance of the proposed face detection using real surveillance images. From Table1, we can find that the proposed ITB and RGB feature images can achieve $F1_{Proposed} = 0.96$ and $F1_{RGB} = 0.98$ (QP = 22) when using the real surveillance images from our laboratory to perform training and testing. It is clear that the proposed ITB feature image can not only finish a very fast face detection but also achieve a very close value of F1-measure to RGB image.

V. CONCLUSION

In this paper, we proposed ITB feature image can get faster face detection with a similar F1 value as compared with RGB feature image. On the other hand, the proposed method also achieve more accurate face detection when compared to Alvar's method.

Table 3. The comparisons of performance using Hallway images.

QP	Method	Acc	Precision	Recall	F1	time(s)
QP = 22	RGB	98%	1	0.96	0.98	6.36
	Alvar [5]	94%	0.96	0.9	0.93	1.42
	Proposed	97%	1	0.93	0.96	1.37
QP = 32	RGB	97%	0.98	0.97	0.97	6.17
	Alvar [5]	93%	0.93	0.853	0.89	1.31
	Proposed	95%	0.98	0.96	0.97	1.29
QP = 42	RGB	93%	0.95	0.9	0.92	5.88
	Alvar [5]	84%	0.89	0.836	0.86	1.12
	Proposed	89%	0.92	0.871	0.89	1.14

Table 4. The comparisons of performance using real surveillance images.

QP	Method	Acc	Precision	Recall	F1	time(s)
QP = 22	RGB	99%	1	0.99	0.99	6.32
	Alvar [5]	96%	0.98	0.94	0.96	1.36
	Proposed	98%	1	0.96	0.98	1.32
QP = 32	RGB	98%	1	0.97	0.98	6.09
	Alvar [5]	95%	0.98	0.93	0.95	1.23
	Proposed	96%	0.97	0.96	0.96	1.19
QP = 42	RGB	94%	0.97	0.93	0.95	5.94
	Alvar [5]	87%	0.98	0.798	0.88	1.18
	Proposed	90%	0.99	0.857	0.92	1.11

REFERENCES

[1]. G. J. Sullivan, J-R Ohm, W-J Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard". IEEE Trans. on Circuitsand Systems for Video Technology, vol. 22, pp. 1649-1668, Dec. 2012.

[2]. High Efficiency Video Coding, Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.

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

[4]. 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, pp.3676-3684, 2015.

[5]. S. R. Alvar, H. Choi, and I. V. Bajı´c, "Can you tell a face from a HEVC bitstream?," 2018 IEEE Conference on Multimedia Information Processing and Retrieval, pp.257-261, June 2017.

[6]. H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," IEEE Trans. Circuits Syst. Video Technol., vol. 17, pp. 1103-1120, Sep. 2007.

[7]. T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overviewof the H.264/AVC video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003.

[8]. Reference software HM16.7, https://hevc.hhi. fraunhofer.de/svn/svn_HEVCSoft are/branches/

[9]. G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[10]. G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller, "Learning to align from scratch," in Neural Information Processing Systems, 2012.
[11]. 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," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.
[12]. YUV sequences, http://trace.eas.asu.edu/yuv/index.html

Chou-Chen Wang "Fast Face Detection System Based On HEVC Bitstream" The International Journal of Engineering and Science (IJES), 8.8 (2019): 62-69