INTRA PREDICTION USING FULLY CONNECTED NETWORK FOR VIDEO CODING

Jiahao Li[†]*, Bin Li*[*]*, Jizheng Xu*[*]*, and Ruiqin Xiong*[†]

[†]Peking University, Beijing, 100871, China
[*]Microsoft Research Asia, Beijing, 100080, China
[†]{jhli.cn, rqxiong}@pku.edu.cn; [*]{libin, jzxu}@microsoft.com

ABSTRACT

Traditional intra prediction methods exploit some fixed rules to generate prediction, which might not be adaptive enough to handle complicated contents. In this paper, we investigate applying deep neural network to improve the state-of-the-art intra prediction. Considering the characteristics of block-based video coding framework, we propose a fully connected network for intra prediction where all layers except non-linear ones are fully connected. In the proposed network, the inputs are multiple reference lines of the current block and the output is the prediction for the block. When compared with the traditional intra prediction method, the richer context of current block is exploited. For this reason, the proposed network is capable of providing more accurate prediction. Experimental results demonstrate the effectiveness of proposed network. When integrated into the HEVC reference software, the proposed method can achieve up to 3.3% bitrate saving and an average of 1.6% bitrate saving for 4K sequences.

Index Terms— Video coding, intra prediction, deep learning, fully connected network, rate distortion optimization

1. INTRODUCTION

High Efficiency Video Coding (HEVC) [1] has become the the state-of-the-art video coding standard. When compared with its predecessor H.264/AVC, HEVC can achieve about 50% bit saving while providing similar visual quality.

For intra angular prediction, HEVC follows H.264/AVC, and still takes the directional prediction model which assumes that the visual content usually follows a single propagation direction. For each pixel in current block, it will be projected to the nearest reference line along the angular direction, and the projected pixel is used as the prediction [2]. Besides this model, there are some researches about non-directional models, such as the 2-D Markov model [3] and the inpainting model [4].

However, HEVC intra prediction and aforementioned methods [3, 4] only utilize the nearest reference line considering its strong statistical correlation. Actually, the farther non-adjacent region also can be utilized. In [5, 6], the multiple reference lines are exploited. Nevertheless, for prediction generation with a specified reference line, [5, 6] still takes the directional model.

Recently, the deep learning technology has shown superior performance in many tasks, such as image recognition [7], image restoration [8]. For video coding, there are also some related works. In [9], a convolutional neural network (CNN) is proposed to replace the sample adaptive offset (SAO) in HEVC. In [10], a CNN integrating variable filter size is designed for artifact reduction in HEVC intra coding. In [9, 10], the input of CNN is the whole reconstructed image.

In this paper, we propose a fully connected network for intra prediction, called IPFCN. Different from [9, 10], the proposed IPFCN is designed for each block. In particular, we use the fully connected layer to construct the bridge between the reconstructed reference pixels and current block. The inputs of IPFCN are multiple reference lines of current block, and the output is the prediction for the block. Compared with the methods using the individual nearest reference line, more context information of current block is utilized in IPFCN. For this purpose, IPFCN has the potential to provide more accurate prediction. Experimental results demonstrate the effectiveness of IPFCN. When compared with HM (HEVC reference software), IPFCN can achieve an average of 1.6% bitrate saving for 4K sequences, where the maximum one is 3.3%.

The rest of this paper is organized as follows. Section 2 describes the proposed method in detail. Experimental results are shown in Section 3. Conclusions and future work are presented in Section 4.

2. PROPOSED METHOD

In this section, the proposed method is described in detail. First, the structure of IPFCN is elaborated. Second, the train-

This work was supported in part by the National Natural Science Foundation of China under Grant 61370114, the National Basic Research Program of China under Grant 2015CB351800, the Beijing Natural Science Foundation under Grant 4172027, and also by the Cooperative Medianet Innovation Center. The work was done when J. Li was with Microsoft Research Asia.

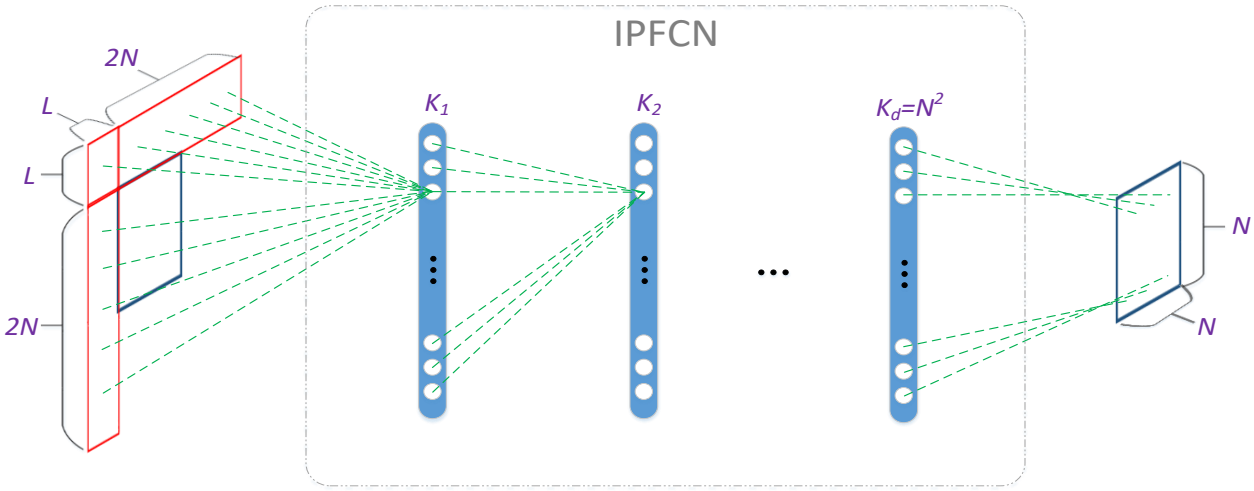

Fig. 1. The structure of proposed IPFCN. The depth of network is d. The size of the current block is $N \times N$. The inputs of IPFCN are L reference lines of the current block. The output of IPFCN is the prediction of the current block in a raster order. Except the last one, each fully connected layer is followed by a PReLU layer. For more concise presentation, the non-linear PReLU layer is not visualized.

ing process is described. Third, the parameters of IPFCN are investigated. Lastly, we introduce the integration in HM.

2.1. Network Architecture

For intra prediction, we propose a fully connected network. The configuration is outlined in Fig. 1. For a block with size of $N \times N$, we use its neighboring reference lines as the inputs of IPFCN. The number of reference lines is indicated by L. There are a total of $4NL + L^2$ reference pixels. By contrast, the traditional directional model only utilizes $4N + 1$ pixels. The output of IPFCN is a N^2-dimensional vector, corresponding to the prediction of current block in a raster order. The depth of network is indicated by d. In IPFCN, all of the d layers are the fully connected layers. For the ith layer, its output is a vector with K_i-dimensional, and it is calculated as:

$$F_i(x) = W_i x_i + b_i, \qquad 1 \leq i \leq d, \tag{1}$$

where W_i and b_i represents weights and biases. x_i is the input vector of the ith layer. When i equals one, namely the first layer, the x_1 is $4NL + L^2$-dimensional. When i is larger than one, the x_i is K_{i-1}-dimensional, i.e. the output of the $i-1$th layer, where this output will pass a non-linear activation layer first.

In IPFCN, we take the PReLU (Parametric Rectified Linear Unit) [11] as the activation layer. Compared with the commonly-used ReLU, PReLU has larger flexibility. For the positive part of input, both of PReLU and ReLU will not modify it. However, for the negative part, ReLU always fixes it to zero. By contrast, PReLU uses a factor a_i to scale it, and a_i is learnable. For this reason, the network cooperating with PReLU has larger capacity.

2.2. Training

Learning the entire mapping F from neighboring reference pixels to current prediction block needs to estimate the parameters $\Theta = \{W, b, a\}$ in the fully connected layers and PReLU layers. Specifically, given a collection of M training instances, where the jth instance consists of the ground truth block y^j and the neighboring reference pixels x_1^j, we minimize the mean squared error (MSE):

$$L(\Theta) = \frac{1}{M} \sum_{j=1}^{M} ||F(x_1^j, \Theta) - y^j||_2^2. \tag{2}$$

All parameters are optimized using stochastic gradient descent with the standard back propagation. We implement and train our network using Caffe package [12].

2.3. Network Parameters

In the proposed IPFCN, two factors have important influence on the performance. One is the network depth. The other is the dimension of fully connected layer. These two factors determine the size of network. For this reason, we investigate them separately.

Network Depth With the support of powerful computational devices, many deep learning networks become deeper. In [13], a 1001 deep residual network is proposed for image recognition. However, not for all kinds of networks, the

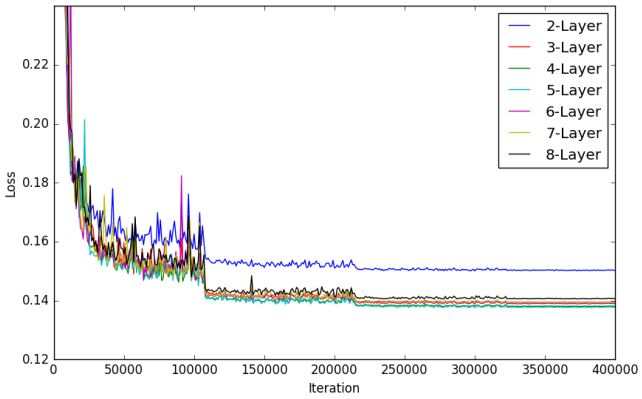

Fig. 2. The validation errors (scaled MSE) of proposed IPFCNs with different depths. The block size is 8×8, and the number of reference lines is 4. In a specified IPFCN, all fully connected layers except the last one are 128-dimensional. All IPFCNs are trained from the same training data set which is extracted from Netflix sequences. The validation set consists of *BasketballDrill*, *FourPeople*, *BQSquare*, *ParkScene*, and *Traffic*. The details of training and validation data sets can be found in Section 3.

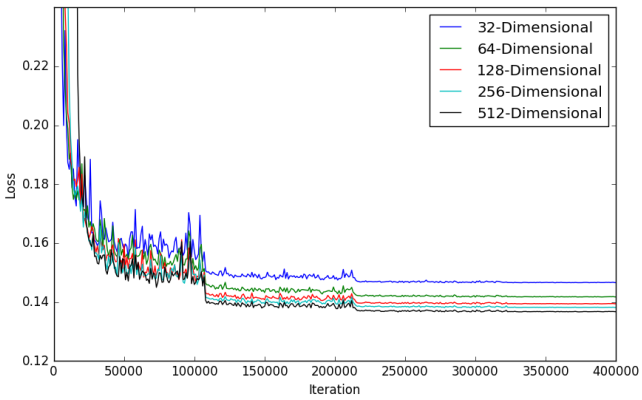

Fig. 3. The validation errors of proposed IPFCNs with different dimensions. In a specified IPFCN, all fully connected layers except the last one have the same dimension. All IPFCNs have the same depth 3. The block size, number of reference lines, training data set, and validation data set are same with those in Fig. 2.

deeper model can bring benefit. In [8], the phenomenon that the deeper model leads to accuracy degradation is observed. To investigate the influence of the network depth in proposed IPFCN, we collect the validation errors when using IPFCNs with different depths. The comparison is shown in Fig. 2. From this comparison, we can find that the 3-layer model can outperform the 2-layer model with a relatively large margin. However, the deeper model cannot further improve the performance. The 8-layer model even has performance loss.

Layer Dimension Except that the last layer is fixed to N^2-dimensional, the dimension of others layers can be adjustable. For this reason, we also investigate the influence of dimension. The validation errors of IPFCNs with different dimensions are compared in Fig. 3. From this figure, we can see that the larger the dimension is, the better performance can be obtained. However, when the dimension goes larger, the margin of improvement becomes smaller.

2.4. Integration in HM

In this paper, we propose that the IPFCN cooperates with the directional model in video coding framework, and use the rate distortion optimization to choose the best model. A binary flag will be transmitted to the decoder to indicate which model to use. As aforementioned, the IPFCN with 3-layer already learns the intra prediction well, and the benefit brought by increasing dimension goes smaller. Considering that the larger model costs more computation, we uses the 128-dimensional IPFCN with 3-layer when we integrate it in HM.

3. EXPERIMENTS

3.1. Experiment Setting

The proposed method in this paper is just a preliminary work. At present, we only investigate the case of 8×8 intra prediction. The number of reference lines is set as 4.

Training Setting The training data set comes from 39 4096x2160 sequences in [14] and 9 1920x1080 sequences in [15]. To accelerate the training process, the 39 4K sequences are down sampled to 1024x540[2]. As we expect IPFCN to handle the complex textures which direction model cannot predict well, the training data set should contain these textures. For this reason, we will screen the 48 sequences to extract the internal blocks with complex textures. For each block, we calculate the minimum SATD (sum of absolute transformed differences) cost in HEVC 35 intra modes. If this cost is larger than the average cost of all blocks in a frame, we think this block has relatively complex texture, and should be added into training data set. In addition, if the cost is too much larger than the average cost, it still will not be used for training because we think this block is too complex to predict. The threshold of detecting the too large cost is the 2.5 times of the average cost. After the screening procedure, a total of 1,375,232 blocks are selected. The base learning rate is set to decay exponentially from 0.1 to 0.00001, changing every 20 epochs. Thus, in total the training takes 100 epochs and uses no more than one hour on a *Tesla K40m* GPU.

Video Coding Setting We implement the IPFCN into HEVC reference software HM-16.9 [16]. The intra main

[2]Generally speking, it takes a network less time to converge in a smaller training data set.

Table 1. The BD-Rate Results of The Proposed Method

Sequence		BD-Rate		
		Y	U	V
4K	Tango	−3.3%	−4.8%	−4.5%
	Drums100	−1.5%	−1.9%	−1.6%
	CampfireParty	−0.5%	−0.8%	−0.9%
	ToddlerFountain	−2.1%	−2.4%	−2.7%
	CatRobot	−1.0%	−1.7%	−1.3%
	TrafficFlow	−1.4%	−1.8%	−1.9%
	DaylightRoad	−1.6%	−3.4%	−2.8%
	Rollercoaster	−1.8%	−2.5%	−2.1%
Class A	Traffic	−1.0%	−1.4%	−1.6%
	PeopleOnStreet	−1.3%	−1.6%	−2.0%
	Nebuta	−1.6%	−1.4%	−1.5%
	SteamLocomotive	−1.7%	−2.3%	−2.6%
Class B	Kimono	−3.2%	−4.1%	−4.0%
	ParkScene	−1.1%	−1.3%	−1.4%
	Cactus	−0.9%	−1.3%	−1.7%
	BasketballDrive	−0.9%	−2.1%	−1.2%
	BQTerrace	−0.5%	−0.1%	−0.1%
Class C	BasketballDrill	−0.3%	−1.5%	−1.5%
	BQMall	−0.3%	−0.3%	−0.5%
	PartyScene	−0.4%	−0.5%	−0.4%
	RaceHorsesC	−0.8%	−1.5%	−1.1%
Class D	BasketballPass	−0.4%	−1.4%	−1.0%
	BQSquare	−0.2%	−1.0%	0.5%
	BlowingBubbles	−0.6%	−0.2%	−1.0%
	RaceHorses	−0.6%	−1.2%	−1.4%
Class E	FourPeople	−0.8%	−1.0%	−2.3%
	Johnny	−1.0%	−1.3%	−1.4%
	KristenAndSara	−0.8%	−1.1%	−1.1%

Table 2. The Average Results

Sequence	BD-Rate		
	Y	U	V
4K	−1.6%	−2.4%	−2.2%
Class A (1600P)	−1.4%	−1.7%	−1.9%
Class B (1080P)	−1.3%	−1.8%	−1.7%
Class C (WVGA)	−0.5%	−1.0%	−0.9%
Class D (WQVGA)	−0.5%	−0.9%	−0.7%
Class E (720P)	−0.9%	−1.1%	−1.6%
All Average	−1.1%	−1.6%	−1.6%
Encoding Time	148%		
Decoding Time	290%		

configuration in the common test conditions (CTC) [17] is used. The anchor and proposed method only allow 8×8 intra coding. The QP (quantization parameter) values are set to 22, 27, 32, and 37. It is noted that the encoders/decoders with the four QPs will share the same IPFCN model. The results are measured by BD-Rate [18]. The testing sequences include the class A∼E sequences in CTC. In addition, to test our method on sequences with higher resolution, we choose eight 4K sequences from [19]. The first 5 frames are tested. It is noted that the testing sequences have no overlap with the training sequences.

3.2. Experimental Results

The results of the proposed IPFCN for video coding are shown in Table 1. In addition, we collect the average results according to the resolution, as shown in Table 2. From these two tables, we can see that the proposed IPFCN can achieve an average of 1.1% bitrate saving on luma component. The maximum one is 3.3% for *Tango*. At the same time, the two chroma components both have 1.6% bitrate saving. In particular, from Table 2, we can find that the proposed IPFCN has 1.6%, 1.4%, 1.3%, and 0.9% gain on 4K, 1600P, 1080P, and 720P sequences, respectively. By contrast, the performances on WVGA and WQVGA are both 0.5%. This phenomenon is possibly caused by the training data set, which only contains 1080P and 1024x540 sequences. The IPFCN trained from high resolution sequences cannot handle the low resolution sequences very well. For encoding and decoding time, the proposed method bring additional 48% and 190% cost. This mainly comes from the forward computation of IPFCN. In our current implementation, the parameters are in float precision, which is not computationally friendly for video coding. In addition, the inputs of IPFCN will be scaled into the range [0,1], where the float division operation is introduced. However, the complexity can be reduced by using code optimization, and this will be implemented in the future.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose using deep neural network for intra prediction in video coding. The proposed IPFCN can learn an end-to-end mapping from the neighboring reference pixels to the current block. Experimental results show that the IPFCN can obtain 1.1% bitrate saving on average when compared with the HEVC anchor. In particular, the performance of 4K sequences can reach 1.6% gain.

In this paper, we find that the 128-dimensional IPFCN with 3-layer can work well. The larger model seems not much helpful. This is based on the training set containing 48 sequences. However, when using a larger training set, the results may be different. This will be investigated in the future. In addition, this paper only investigates the block size of 8×8. We will research other block sizes in the future.

5. REFERENCES

[1] Gary Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] Jani Lainema, Frank Bossen, Woo-Jin Han, Junghye Min, and Kemal Ugur, "Intra coding of the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[3] Fatih Kamisli, "Intra prediction based on markov process modeling of images," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3916–3925, 2013.

[4] Dong Liu, Xiaoyan Sun, Feng Wu, Shipeng Li, and Ya-Qin Zhang, "Image compression with edge-based inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1273, 2007.

[5] Jiahao Li, Bin Li, Jizheng Xu, and Ruiqin Xiong, "Efficient Multiple Line-Based Intra Prediction for HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[6] Jiahao Li, Bin Li, Jizheng Xu, and Ruiqin Xiong, "Intra prediction using multiple reference lines for video coding," in *Data Compression Conference (DCC), 2017*. IEEE, 2017, pp. 221–230.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[8] Chao Dong, Chen Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[9] Woon-Sung Park and Munchurl Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*. IEEE, 2016, pp. 1–5.

[10] Yuanying Dai, Dong Liu, and Feng Wu, "A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," *arXiv preprint arXiv:1603.05027*, 2016.

[14] Andrey Norkin, Ioannis Katsavounidis, Anne Aaron, and Jan cock, *Netflix test sequences for next generation video coding*, VCEG-AZ09, 52nd Meeting: Warsaw, June 2015.

[15] Andrey Norkin, *Proposed test sequences for 1080p class*, JVET-C0041, 3rd Meeting: Geneva, May 2016.

[16] *https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.9/.*

[17] Frank Bossen, *Common test conditions and software reference configurations*, JCTVC-L1100, 12th Meeting: Geneva, Jan 2013.

[18] Gisle Bjntegaard, *Improvements of the BD-PSNR model*, Document VCEG-AI11, July 2008.

[19] Karsten Suehring and Xiang Li, *JVET common test conditions and software reference configurations*, JVET-B1010, 2nd Meeting: San Diego, Feb 2016.