CNN-Based Transform Index Prediction in Multiple Transforms Framework to Assist Entropy Coding

Saurabh Puri
Technicolor and
Université de Nantes, France
Email: saurabh.puri@technicolor.com

Sébastien Lasserre
Technicolor
Cesson-sévigné, France
Email: sebastien.lasserre@technicolor.com

Patrick Le Callet
IRCCyN
Université de Nantes, Nantes, France
Email: patrick.lecallet@univ-nantes.fr

Abstract—**Recent work in video compression has shown that using multiple 2D transforms instead of a single transform in order to de-correlate residuals provides better compression efficiency. These transforms are tested competitively inside a video encoder and the optimal transform is selected based on the Rate Distortion Optimization (RDO) cost. However, one needs to encode a syntax to indicate the chosen transform per residual block to the decoder for successful reconstruction of the pixels. Conventionally, the transform index is binarized using fixed length coding and a CABAC context model is attached to it. In this work, we provide a novel method that utilizes Convolutional Neural Network to predict the chosen transform index from the quantized coefficient block. The prediction probabilities are used to binarize the index by employing a variable length coding instead of a fixed length coding. Results show that by employing this modified transform index coding scheme inside HEVC, one can achieve up to 0.59% BD-rate gain.**

I. Introduction

Transforms are a key element in all block-based video coding system which in conjugation with quantization, is important for the overall compression efficiency of the system. Historically, the transform coding part in a video codec has remained conservative in order to keep the complexity low. Recent works have shown that it is possible to gain substantially by using adaptive multiple transforms instead of a single transform. These multiple transforms are either (1) systematic fixed transforms [1], [2], (2) learned offline on a large training set [3]–[5] or (3) learned on-the-fly [6].

The motivation to use multiple or adaptive transforms comes from the fact that a single transform is not efficient to model different statistical variations that may be present in an intra-predicted residual [4]. By using multiple transform candidates, the encoder is given a choice to select the transform for a particular residual block that provides minimum cost in terms of both rate and distortion. This is usually done using an exhaustive Rate Distortion Optimization (RDO) search. However, this complexifies the encoder significantly due to the many possible combinations of coding modes. Several schemes have been proposed to reduce the encoder complexity [7].

The multiple transform schemes proposed in the literature falls into two categories depending on whether or not an additional syntax to indicate the choice of the transform is added to the bitstream. If not, this information is implicitly derived from the causal information available at the decoder side. If yes, the index is explicitly encoded in the bitstream. It has been shown in the literature that this additional syntax has a huge impact on the overall performance of the scheme. In particular, the impact is maximum for small residual blocks of size 4×4 [8].

Therefore, in order to reduce the overhead of transmitting the syntax, this work aims to predict the transform index from the causal part of the bit-stream. In the proposed method, the quantized transform coefficient block (TCB) is utilized to predict the index as it is available at the decoder and is parsed before the parsing of the transform index. Predicting the transform index can be seen as a classification problem where each TCB belongs to a class labeled by its transform index.

Deep convolution neural networks (CNNs) have shown remarkable results in classification tasks where the correlation is not easy to model through simple linear models [9]. For this reason, a CNN is utilized to model the correlation between the TCB and the transform index.

Therefore, our contribution involves designing a novel CNN-based transform index prediction method that is trained on a large offline dataset containing a large collection of TCBs from different transforms. We have implemented the trained CNN inside the video codec to perform the classification, and the prediction from the CNN is used to improve the entropy coding of the syntax.

This paper is organized as follows: Section II will provide some of the recent related works. Section III will detail the proposed CNN-based transform index prediction method. Section IV will present the CNN architecture used to train the model. Finally, simulation results will be presented in section V.

II. Related Works

Video compression in virtually any video codec is mainly achieved by employing prediction followed by entropy coding of the residual. The prediction is usually computed using the causal information that is available to both encoder and decoder. By designing a better prediction model, one may achieve higher compression. However, it is extremely difficult to construct a model when there is no prominent pattern in the signal. For such cases, neural networks are suspected to be more efficient and therefore, many interesting works have

ISBN 978-0-9928626-7-1 © EURASIP 2017

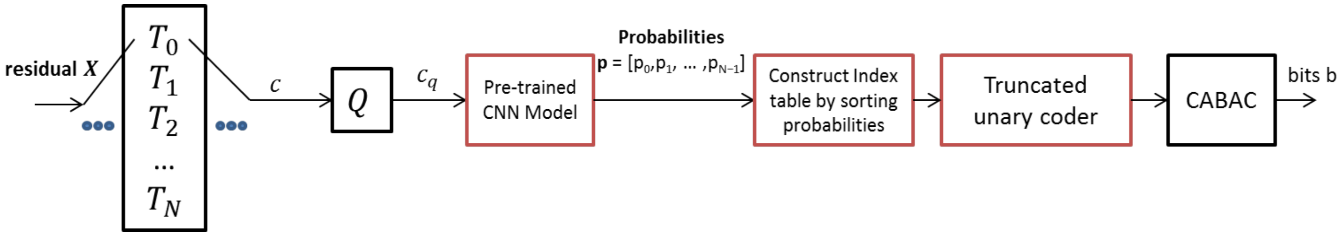

Fig. 1: Block Diagram of proposed CNN-based transform index coding

emerged recently that utilize CNN based model to extract difficult features or characteristics present inside data.

Most of the recent works are focused on making early decisions to speed-up the encoding process. A work carried by Yu et. al. in [10] proposes to use CNNs to provide a binary decision on whether to split the block or not by taking into consideration several features extracted from the block. Similar works have been carried in [11], [12] where a CNN is used to make a-posteriori decision of splitting of coding unit (CU) and prediction unit (PU) blocks. Laude et. al. in [9] have shown that one can mimic the intra-prediction mode decision taken by RDO using a CNN model which is trained on a large training set.

Our work is related to the above works as we utilize CNNs to perform a classification of coding modes. However, in this work, CNNs are used specifically for the transform index prediction. Moreover, the decision of the CNN is used to drive the entropy coder inside the codec. Finally, in this work, the quantized coefficients obtained after the transform and quantization step are used as the input features instead of the hand-crafted features as done in some of the related works. The features are extracted using CNN learning process which is able to model the complex structures present in the coefficient block.

III. Proposed CNN-Based Transform Index Prediction

In this section, we describe our novel CNN-based transform index prediction method which is employed at both encoder and decoder sides. Firstly, the multiple transform competition scheme (MDTC) as proposed in [5] and various indexing schemes proposed in the literature that are considered as baseline for our method are described. Then, the proposed transform index coding scheme is detailed.

The MDTC scheme proposed in [5] tests N offline learned transform matrices in competition with the core HEVC transform (DCT/DST) inside a RDO loop and selects the transform matrix that provides the minimum RD cost. A transform index is coded to indicate the choice amongst N+1 transforms to the decoder for proper reconstruction of the block. This is done by first coding a flag that indicates whether the DCT/DST transform is used or not. If the flag stipulates it is not, the offline learned transforms are used and a fixed length coding is used. The scheme clearly favors the DCT/DST as it requires fewer bits to encode.

An alternative way of signaling the transform choice would be to directly binarize the transform index using a fixed length coding, to indicate N+1 transform candidates on b bits where

$$b = ceil(log_2(N+1))$$

These bits are entropy coded using CABAC. This approach does not favor DCT/DST over offline learned transforms inside the RDO loop. In this work, this indexing scheme is used as a baseline in order to compare the new proposed indexing scheme.

In the proposed scheme, a variable length coding is used instead of a fixed length coding as done in the literature so far. A pre-trained CNN-based model is used to predict the probabilities of a particular transform index which is then used to construct a truncated unary code per block. Figure 1 illustrates the block diagram of the CNN-based transform index coding scheme for a 4×4 luma residual block X. The blocks highlighted in red in figure 1 shows the modifications over the fixed length coding. Inside the modified HEVC codec [5], the core DST transform (T_0) is tested in competition with offline learned transforms (T_1 to T_N). The CNN-based model is put inside an RDO loop which takes quantized coefficients c_q as input and outputs a vector p of probabilities of predicting a particular transform index i. The vector p is utilized to construct a truncated unary code which is simply done by re-arranging the probabilities in p in the decreasing order and using minimum bits (1 bit) for the transform index that is predicted with highest probability and maximum bits (N bits) for least probable transform index.

In order to understand it better, let us consider N to be equal to 3. The residual X is tested with four transform candidates (i.e. T_0 to T_3) and in each case, the quantized coefficient is passed through the pre-trained CNN model to obtain the probabilities. Let us suppose that T_2 is the selected transform, the residual block is thus transformed with T_2. The quantized coefficients c_q of the transformed block are passed through the trained CNN model which outputs the probability values, say [0.15, 0.1, 0.45, 0.30]. Table I shows the truncated unary

Transform Index	Probabilities	Truncated Unary Code
2	**0.45**	**0**
3	0.30	1 0
0	0.15	1 1 0
1	0.10	1 1 1

TABLE I: Truncated Unary Code for example I

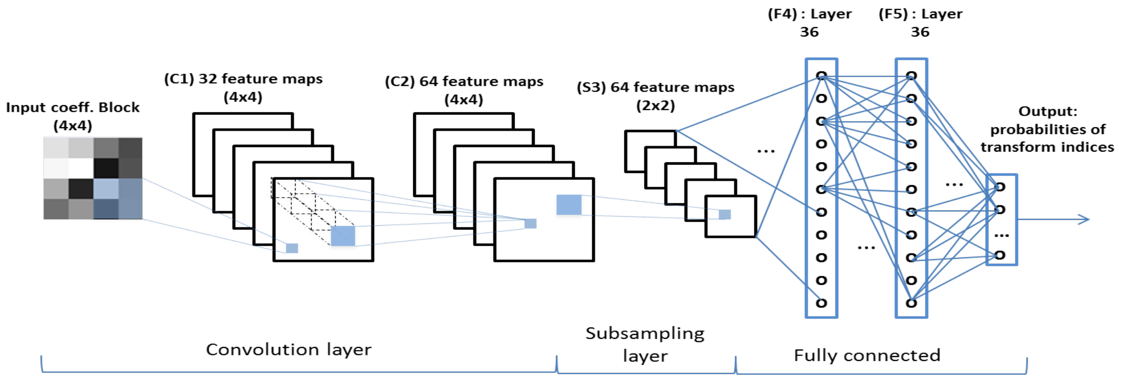
Input coeff. Block (4x4)
(C1) 32 feature maps (4x4)
(C2) 64 feature maps (4x4)
(S3) 64 feature maps (2x2)
(F4) : Layer 36
(F5) : Layer 36
Output: probabilities of transform indices
Convolution layer
Subsampling layer
Fully connected

Fig. 2: Proposed CNN architecture for classification of transform index using 4×4 size coefficient block

Transform Index	Probabilities	Truncated Unary Code
2	0.45	0
0	**0.30**	**1 0**
3	0.15	1 1 0
1	0.10	1 1 1

TABLE II: Truncated Unary Code for example II

code for this example I. In this case, the original index 2 is well predicted by the CNN model and therefore, only 1 bit is coded, namely ‘0’.

In another example with N=3 and T_0 being the selected transform, the quantized coefficients of the transformed block are passes through the trained CNN model which outputs the probability values, say [0.30, 0.1, 0.45, 0.15]. Table II shows the truncated unary code for this example. In this case, the original index 0 is coded in the bitstream with 2 bits, namely ‘10’.

It should be noted that the performance of this approach greatly depends on the classification accuracy of the CNN model that is designed for the task of classifying different transform candidates using the quantized coefficients.

Therefore, the algorithm involves two major steps:

1) offline training of the CNN model per intra prediction direction on a large independent data set as described in section IV, and
2) applying the trained CNN model inside the HEVC RDO search to predict the transform index as described in this section

In the next section, the architecture of different layers of the CNN is described in detail along with the method to train the CNN-models for different intra-prediction directions.

IV. Architecture and Training of CNN Models

In this section, we will describe the CNN-architecture as illustrated in Figure 2 for a 4×4 coefficient block. As mentioned in the previous section, the performance of the proposed scheme relies on the classification accuracy of the CNN-models trained offline on a large data set. The selection of training parameters for the model is therefore a critical part in the design of the method.

Table III summarizes all the model parameters chosen for different CNN layers. The architecture is inspired from laude et. al. [13] which was adapted to handle smaller input block sizes of a block based codec. The capacity of the network has been further reduced by using fewer filters and neurons and by using even smaller filter size of 2×2. However, one additional fully connected layer with 36 neurons has also been added.

The first convolutional layer takes coefficient block of size 4×4 as input and is passed through 32 filters of size 2×2 and a stride of one. The second convolution layer operates over the output of the first layer which uses 64 filters of size 2×2 and stride of one. A max-pooling layer is used to reduce the size to 2×2×64. This is then fed to the fully connected layers with 36 perceptron. The final softmax layer outputs the probabilities.

This CNN model is trained using Keras framework which is a well-known high-level python neural network library that runs on top of TensorFlow or Theano [14]. Keras is configured to use Theano as the backend. Additionally, the model parameters are optimized by using gradient-based optimizer called Adam [15] which is available in Keras.

In order to improve the prediction accuracy of the CNN model, following pre-processings of the training samples are performed.

- Firstly, training samples have been extracted from an independent dataset [16] which contains images of various buildings.
- only coefficient blocks with at-least 3 non-zero coefficients are considered.
- the coefficient blocks where the above and left samples are not available are not taken into account.

Layer	Type	Outputs	Filter size	Stride
1	Convolutional C1	32	2x2	1
2	ReLU	-	-	-
3	Convolutional C2	64	2x2	1
4	ReLU	-	-	-
5	Max Pooling S3	64	2x2	2
6	Flatten	256	-	-
7	Fully-connected	36	-	-
8	ReLU	-	-	-
9	Fully-connected	36	-	-
10	ReLU	-	-	-
11	Fully-connected	2	-	-
12	Softmax	-	-	-

TABLE III: Parameters of CNN models

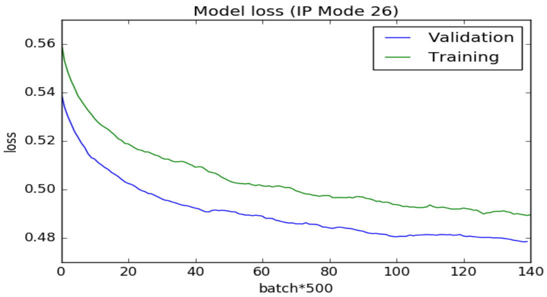

Fig. 3: Training and Validation loss curves using CNN based learning

Accuracy for IPM 26	N=1		N=3	
	Validation	Test	Validation	Test
CNN	0.75	0.72	0.54	0.45
PCA	0.71	0.67	0.43	0.37

TABLE IV: Trained CNN-model classification accuracy

- Finally, imbalanced classes are avoided by manually balancing the number of coefficients in each class.

V. Simulation Results

The proposed CNN-based transform index coding scheme has been implemented and evaluated in the HEVC test software HM version 15.0 which is modified to test multiple transform candidates inside the RDO loop similar to [5]. Two sets of results from various experiments are presented: binary classification (N=1) and general case (N=3) for only 4×4 luma intra residual block size. For all experiments, all-intra (AI) configuration as per HEVC CTC [17] is used.

For the training of the CNN-model, we have taken the Zurich Building dataset [16] which contains over 1000 images in PNG format that are converted to a YUV format of resolution 640×480. The selected transform index based on RDO at the encoder is used to label the corresponding quantized coefficient block in order to obtain the training classes. Only four CNN-models are trained on the four major intra-prediction modes (IPM), namely DC, Planar, Vertical and Horizontal. Training is done on a batch of 32 TCBs and the number of iterations on the data set is set as 20.

In the first experiment, the performance of the training process in terms of classification accuracy on both validation and test data set is evaluated. The validation data set is generated by choosing randomly 10% of the data samples from the training set and is not used for training the model. Validation data set prevents over-fitting. The test data set is generated from the first frame of the HEVC CTC sequences [18]. Figure 3 shows the classification loss curve for both training and validation data set. Clearly, both training and validation loss reduce with the number of iterations over the batch of TCB.

Table IV shows the CNN-model classification accuracy for both N=1 and N=3 in case of vertical IP mode (i.e. IPM 26). For comparison, the classification accuracy of using a Principle Component Analysis (PCA) along with a decision tree classifier is presented in Table IV. From Table IV it is observed that CNN-based classifier outperforms the PCA based classifier on both the validation and test data set. Moreover, an accuracy of over 70% and 45% in case of N=1 and N=3 respectively on both test and validation data set shows that a correlation between quantized transform coefficients and their corresponding transform indexes exist across different contents and is well captured by the CNN-model. Similar trends were observed for other IP modes.

Class	Sequence	EP	CTXT	CNN	NoIndex
A	Nebuta	-0.32	-0.28	-0.28	-0.20
	PeopleOnStreet	-0.71	-0.73	-0.80	-1.16
	SteamLocomotive	0.02	-0.01	-0.02	-0.11
	Traffic	-0.81	-0.77	-0.84	-1.19
	Overall	-0.45	-0.45	-0.49	-0.66
B	BasketballDrive	-0.38	-0.57	-0.54	-0.42
	BQTerrace	-1.62	-1.65	-1.79	-2.22
	Cactus	-1.31	-1.20	-1.36	-1.16
	Kimono	0.34	-0.10	0.02	0.04
	ParkScene	-0.44	-0.47	-0.56	-1.35
	Overall	-0.68	-0.80	-0.84	-1.11
C	BasketballDrill	-4.54	-4.66	-5.01	-4.29
	BQMall	-2.13	-1.92	-2.11	-2.72
	PartyScene	-2.19	-2.31	-2.37	-2.87
	RaceHorses	-1.69	-1.49	-1.73	-2.17
	Overall	-2.64	-2.60	-2.80	-3.01
D	BasketballPass	-1.86	-1.51	-1.76	-2.34
	BlowingBubbles	-2.28	-2.22	-2.69	-2.80
	BQSquare	-3.05	-3.04	-3.06	-3.34
	RaceHorses	-2.72	-2.39	-2.50	-2.59
	Overall	-2.48	-2.29	-2.50	-2.77
E	FourPeople	-0.96	-0.97	-0.99	-1.36
	Johnny	-0.26	-0.49	-0.73	-0.91
	KristenAndSara	-1.46	-1.57	-1.75	-1.87
	Overall	-0.89	-1.01	-1.16	-1.38
F	BaskeballDrillText	-4.68	-5.13	-5.24	-4.71
	ChinaSpeed	-1.97	-2.02	-2.12	-2.00
	SlideEditing	-1.54	-1.78	-1.69	1.97
	SlideShow	-1.61	-1.73	-1.98	-1.44
	Overall	-2.45	-2.66	-2.76	-2.53
Overall		-1.60	-1.63	-1.76	-1.91

TABLE V: BD-Rate gain in % on first frame for N=1 case

In the next experiments, we incorporated the CNN-model inside the HEVC to assist the entropy coding process as illustrated in Figure 1. Table V and VI show the BD-rate gains for different HEVC CTC sequences for N=1 and N=3. In order to show the performance enhancement with the proposed scheme, the results are compared to a conventional fixed length (FL) coding schemes under two variants. The first variant encodes the bits b equi-probably (bypass mode) and the second variant utilizes entropy coding with CABAC context (regular mode) when coding the bits. The results of these two variants are presented under *EP* and *CTXT* respectively.

Finally, the BD-rate gains obtained by employing the proposed approach are presented in tables V and VI under *CNN*. A consistent gain across all classes of sequences is observed. The proposed method provides an overall gain of around 0.1% and 0.2% in case of N=1 and N=3 respectively. For sequences like Blowing bubble and Race horses, around 0.5% gain is observed over the conventional transform index signaling approach. In order to illustrate the upper bound of

Class	Sequence	EP	CTXT	CNN	NoIndex
A	Nebuta	-0.37	-0.38	-0.40	-0.37
	PeopleOnStreet	-0.75	-0.69	-0.90	-1.75
	SteamLocomotive	0.03	-0.03	-0.03	-0.19
	Traffic	-0.85	-0.83	-1.10	-1.82
	Overall	-0.49	-0.48	-0.61	-1.03
B	BasketballDrive	-0.22	-0.42	-0.48	-0.47
	BQTerrace	-1.44	-1.56	-1.70	-3.11
	Cactus	-1.02	-1.07	-1.25	-2.48
	Kimono	0.18	-0.27	-0.05	-0.08
	ParkScene	-0.45	-0.59	-0.74	-2.81
	Overall	-0.59	-0.67	-0.84	-1.79
C	BasketballDrill	-3.14	-3.36	-3.34	-3.29
	BQMall	-1.74	-1.81	-1.91	-3.33
	PartyScene	-2.03	-2.14	-2.15	-3.89
	RaceHorses	-1.59	-1.57	-1.82	-3.38
	Overall	-2.12	-2.22	-2.31	-3.47
D	BasketballPass	-1.12	-1.32	-1.20	-2.02
	BlowingBubbles	-1.91	-1.76	-2.25	-2.98
	BQSquare	-2.37	-2.22	-2.50	-3.54
	RaceHorses	-2.29	-1.91	-2.50	-2.74
	Overall	-1.92	-1.80	-2.11	-2.82
E	FourPeople	-0.64	-1.00	-1.24	-1.79
	Johnny	-0.58	-0.46	-0.69	-0.83
	KristenAndSara	-0.87	-1.08	-1.09	-2.08
	Overall	-0.70	-0.85	-1.00	-1.57
F	BaskeballDrillText	-3.06	-3.56	-3.83	-3.57
	ChinaSpeed	-1.86	-1.83	-1.89	-2.15
	SlideEditing	-1.21	-1.26	-1.56	-1.93
	SlideShow	-1.42	-1.23	-1.44	-1.67
	Overall	-1.89	-1.97	-2.18	-2.33
Overall		-1.28	-1.33	-1.51	-2.17

TABLE VI: BD-Rate gain in % on first frame for N=3 case

Sequences	Index Prediction accuracy in %		non-DCT/Total in %
	outside RDO	inside RDO	
PeopleOnStreet	71	85	42
BQSquare	73	91	56
SlideShow	81	93	63

TABLE VII: CNN-model prediction accuracy in HEVC vs actual transform usage statistics

the performance with perfect prediction of index, the BD-rate gain without coding of index for above used IP modes is computed and is presented under *NoIndex* in tables V and VI. It is observed that with the help of the proposed CNN model, this performance gap is partially covered.

Further, Table VII presents prediction accuracy of the CNN in percentage when the CNN is used outside and inside RDO respectively. Clearly, RDO favors the prediction decision made using CNN model. This is expected as it is less expensive to encode the bit when CNN predicts correctly.

The proposed method has a negligible impact on the decoder side complexity but a high impact on the encoder side complexity as the CNN-model is used inside the RDO loop. However, the complexity can be substantially reduced by efficient hardware implementation of CNN as shown in literature [19]. Finally, the complexity reduction is not addressed in this paper and is considered for future work.

VI. Conclusion

In this paper, a novel CNN-based transform index prediction method has been proposed. It is demonstrated that it is possible to partially infer the transform index from the quantized coefficient values by employing a CNN prediction model trained offline on a large independent data set. The proposed method shows a consistent improvement in the BD-rate gain over the fixed length coding scheme in almost all cases. This method provides an average gain of around 0.2% and a maximum gain up to 0.59%. The future work will focus on improving the CNN-based prediction model by using better loss functions and other causal information from the bit-stream.

References

[1] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W.-J. Chien, “Enhanced multiple transform for video coding,” in *Data Compression Conference (DCC), 2016*. IEEE, 2016, pp. 73–82.

[2] B. Zeng and J. Fu, “Directional discrete cosine transforms-a new framework for image coding,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 305–313, 2008.

[3] Y. Ye and M. Karczewicz, “Improved h. 264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 2116–2119.

[4] X. Zhao, L. Zhang, S. Ma, and W. Gao, “Rate-distortion optimized transform for intra-frame coding,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1414–1417.

[5] A. Arrufat, P. Philippe, and O. Déforges, “Mode-dependent transform competition for hevc,” in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1598–1602.

[6] S. Puri, S. Lasserre, and P. L. Callet, “Annealed learning based block transforms for hevc video coding,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 1135–1139.

[7] X. Zhao, L. Zhang, S. Ma, and W. Gao, “Video coding with rate-distortion optimized transform,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 1, pp. 138–151, 2012.

[8] A. Arrufat, P. Philippe, and O. Déforges, “Rate-distortion optimised transform competition for intra coding in hevc,” in *Visual Communications and Image Processing Conference, 2014 IEEE*. IEEE, 2014, pp. 73–76.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] X. Yu, Z. Liu, J. Liu, Y. Gao, and D. Wang, “Vlsi friendly fast cu/pu mode decision for hevc intra encoding: Leveraging convolution neural network,” in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1285–1289.

[11] Z. Liu, X. Yu, S. Chen, and D. Wang, “Cnn oriented fast hevc intra cu mode decision,” in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 2270–2273.

[12] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, “Cu partition mode decision for hevc hardwired intra encoder using convolution neural network,” *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5088–5103, 2016.

[13] T. Laude and J. Ostermann, “Deep learning-based intra prediction mode decision for hevc,” in *Proceedings of 32nd Picture Coding Symposium (PCS)*, 2016.

[14] F. Chollet, “Keras,” https://github.com/fchollet/keras, 2015.

[15] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[16] H. Shao, T. Svoboda, and L. Van Gool, “Zubud-zurich buildings database for image based recognition,” 2003.

[17] F. Bossen *et al.*, “Common test conditions and software reference configurations,” *Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-F900*, 2011.

[18] F. Bossen, B. Bross, K. Suhring, and D. Flynn, “Hevc complexity and implementation analysis,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1685–1696, 2012.

[19] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding,” *CoRR*, vol. abs/1510.00149, 2015. [Online]. Available: http://arxiv.org/abs/1510.00149