Neural network based intra prediction for video coding

J. Pfaff, P. Helle, D. Maniry, S. Kaltenstadler, W. Samek, H. Schwarz, D. Marpe, T. Wiegand
Video Coding and Analytics Department,
Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany

ABSTRACT

Today’s hybrid video coding systems typically perform an intra-picture prediction whereby blocks of samples are predicted from previously decoded samples of the same picture. For example, HEVC uses a set of angular prediction patterns to exploit directional sample correlations. In this paper, we propose new intra-picture prediction modes whose construction consists of two steps: First, a set of features is extracted from the decoded samples. Second, these features are used to select a predefined image pattern as the prediction signal. Since several intra prediction modes are proposed for each block-shape, a specific signalization scheme is also proposed. Our intra prediction modes lead to significant coding gains over state of the art video coding technologies.

Keywords: High Efficiency Video Coding (HEVC), intra prediction, neural network

1 INTRODUCTION

The demand for streaming and storing videos is rising while transmission capacities and memory are limited. This discrepancy is one of the driving forces behind research on video coding technologies with higher compression efficiency. A benchmark in this area is the state-of-the-art High Efficiency Video Coding (HEVC) standard [1] which uses a block-based architecture. For each block, predictive coding is used. Thus, when a receiver of a video signal wants to reconstruct the content of a transmitted video for a given block by using information that is already available, he generates a prediction signal. The prediction signal serves as a first approximation of the video signal to be reconstructed. In a second step, a prediction residual is added to the prediction signal to generate the reconstructed video signal. The content of the prediction residual needs to be transmitted in the bitstream and thus the quality of the prediction signal greatly influences the compression efficiency.

There are two methods to generate a prediction signal: Inter- and intra-picture prediction. In the case of inter-picture prediction, or in abbreviated form, inter prediction, the prediction signal is generated by motion-compensated prediction. This means that the content for blocks that belong to already decoded video frames, which are different from the current frame, serves as the input for the generation of the prediction signal. On the other hand, in the case of intra prediction, the prediction signal is generated out of already reconstructed sample values that belong to the same frame and are typically spatially adjacent to the current block. Due to the scan order in which blocks are processed, these are sample values left and above of the current block.

Although in typical video sequences, inter prediction is used for most of the blocks, intra prediction is crucial for video coding due to the following two reasons. First, in order to guarantee the random access capability of a coded representation of a video sequence, a video sequence needs to be split up in subsequences of consecutive frames where each subsequence needs to be decodable independent from any other subsequence. For the first frame of each such subsequence, a so-called IDR- or key-frame, inter prediction is not possible and thus, only intra prediction can be used. In addition, every subsequent frame of the IDR-frame of this subsequence must be decodable *without* reference to any frame of another subsequence. Second, there might be temporal scene changes within a video sequence where inter prediction fails. The scene changes where intra prediction needs to be applied may not necessarily occur for a whole frame but rather for smaller parts of it. Therefore, it is crucial that intra-coding tools that are applicable in video coding are designed such that they can be applied on blocks of variable, possibly rather small block size. This stands in contrast to a lot of still-image coding techniques which, in order to be efficient, typically require a rather large number of samples.

It can be observed that, although the fraction of blocks coded in intra mode is typically rather small for natural video sequences, their impact on compression performance is often very large. In other words, the bits spent to signal the content of blocks of a video sequence that are coded in intra mode amount to a significant portion of the total number of bits spent to signal the whole video sequence. This is mainly due to the fact that for a lot of blocks where inter prediction is used, the prediction quality is often so good that no prediction residual is present at all. For blocks in intra prediction mode, such a phenomenon occurs rather rarely. Thus, an improvement of the intra prediction quality can typically lead to significant bitrate savings.

In the HEVC standard, the intra prediction is performed in two different ways [1], [2]. First, there are 33 angular prediction modes. These modes essentially copy the already reconstructed sample values on the line above and the column left of the block along a specific direction that is parametrized by an angular parameter. Moreover, there are the DC and the planar mode, the first generating a constant prediction signal that corresponds to the mean sample value on the adjacent samples, the second interpolating between a prediction along the horizontal and the vertical direction. These modes are relatively easy to implement and make sense for all rectangular block shapes.

In the present paper, we describe the results of a data-driven approach to generate intra prediction modes. Namely, intra prediction modes based on a neural network were trained offline and integrated into the video codec. These modes lead to a significant coding gain.

The paper is structured as follows. In section 2.1, the generation of a specific intra prediction signal by a neural network is described. In section 2.2, the signalization scheme for our intra modes, which is based on a second neural network, is outlined. In section 3, experimental results are presented. In section 4, our work is compared to a recent paper by other authors [7]. Moreover, further simplifications to be discussed elsewhere are briefly mentioned.

2 DESIGN OF THE INTRA PREDICTION MODES

2.1 Generation of the proposed prediction signal

The predictions that we designed perform the following two key steps. First, a set of features is extracted from the decoded samples. Second, these features are used to select an affine linear combination of predefined image patterns as the prediction signal.

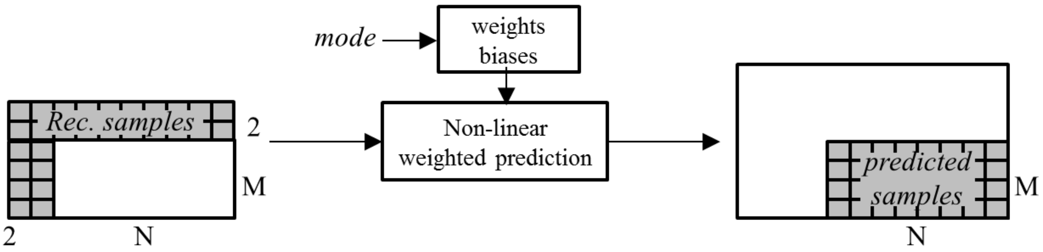

Figure 1. Prediction of MxN intra block from reconstructed samples using a neural network.

More precisely, on a given MxN block with $M \leq 32, N \leq 32$, M and N integral powers of two, the generation of a proposed luma prediction signal $pred$ is performed by processing a set of reference samples r through a neural network as follows. The reference samples r consist of K rows of size N+K above and K columns of size M left of the block, see Figure 1. The number K is set to 2 for all M and N.

The proposed neural network first extracts a vector ftr of features from the reconstructed samples r as follows. If d_0=K*(N +M+K) denotes the number of samples of r, then r is regarded as a vector in the real vector space of dimension d_0. For fixed integral square-matrices A_1 and A_2 which have d_0 rows resp. columns and for fixed integral bias vectors b_1 and b_2 of dimension d_0 one first computes

$$t_1 = \rho(A_1 \cdot r + b_1).$$

Here, $\cdot$ denotes the ordinary matrix-vector product. Moreover, the function ρ is an integer-approximation of the ELU function ρ_0, where the latter function is defined on a p-dimensional vector v by putting

$$\rho_0(v)_i = \begin{cases} v_i, if\ v_i > 0 \\ \exp(v_i) - 1, else, \end{cases}$$

where $\rho_0(v)_i$ and v_i denote the i-th component of the vectors. One applies similar operations to t_1 and computes

$$t_2 = \rho(A_2 \cdot t_1 + b_2).$$

Finally, there is a predefined integral matrix A_3 with d_0 rows and d_0 columns and there is a predefined integral bias vector b_3 of dimension d_0 such that one computes the feature vector ftr as

$$ftr = \rho(A_3 \cdot t_2 + b_3).$$

Out of the feature vector ftr, the final prediction signal $pred$ is generated using an affine linear map followed by the standard Clipping operation $Clip$ that depends on the bit-depth. Thus, there is a predefined matrix $A_{4,k}$ with M*N rows and d_0 columns and a predefined bias vector $b_{4,k}$ of dimension M*N such that one computes

$$pred = Clip\left(A_{4,k} \cdot ftr + b_{4\,,k}\right).$$

Here, $k = predmode$ represents the prediction mode, see the next section. The above generation of the prediction signal $pred$ is depicted at the right hand side of Figure 3 below.

Although we designed several intra prediction modes, the feature extraction, i.e. all but the last layer operations of our networks, is the same for all our proposed modes. This greatly limits the number of parameters needed, since for each block shape we only need one set of matrices A_1 , A_2 and A_3 and bias vectors b_1 , b_2 and b_3 . Also, in this way an encoder that tests several of our intra modes in a rate-distortion-based search can save computations since it needs to compute the feature vector ftr only once.

2.2 Signalization of the specific proposed intra mode

In order to capture different types of image content, we designed n different intra prediction modes, where n is set to 35 for max(M,N) < 32 and to 11 else. At each block where our prediction is to be applied, exactly one of the n modes is to be used. Thus, we face the problem to additionally signal the specific mode, i.e., to signal an index $predmode$ with $0 \leq predmode < n$.

To put the above problem in a broader context, we briefly recall how in the HEVC standard it is signaled which of the 35 HEVC intra prediction modes is used [1], [2]. Here, on a given block, out of the intra modes chosen on the neighboring, already reconstructed blocks left and above of the current block one most probable mode and two second most probable modes are extracted. Then one bin is used to signal whether the current intra prediction mode belongs to one of these three most probable modes or not. If this is the case, one or two additional bins are used to signal the most respectively the two second most probable modes. If this is not the case, the current intra prediction mode belongs to the remaining 32 modes and is signaled in a fixed length code using five bins. This method guarantees in particular that it is possible to propagate the same intra prediction mode over different blocks with very cheap signalization costs.

Unfortunately, this method of mode coding cannot be applied directly for our intra prediction modes. The reason is that, in contrast to the HEVC intra prediction modes, we cannot compare modes between different block shapes. Thus, as an alternative, we also predict the modes from the already reconstructed samples using a second neural network. Using this network, the conditional probability of each of our modes given the reconstructed samples is computed and, depending on their probability, the most probable modes are signaled with less bins than other modes in a way that is exactly parallel to the aforementioned signalization of the HEVC intra prediction modes.

We shall now describe this in more detail. One has

$$n = 3 + 2^k,$$

where $k = 3$ if max(M,N) = 32 and $k = 5$, else. In a first step, an index $predIdx$ with $0 \leq predIdx < n$ is signaled using the following code. First, one bin encodes whether $predIdx < 3$ or not. If $predIdx < 3$, a second bin encodes if $predIdx = 0$ or not, and, if $predIdx \neq 0$, another bin encodes whether $predIdx$ is equal to 1 or 2. If $predIdx \ \geq 3$ then the value of $predIdx$ is signaled in the canonical way using k bins.

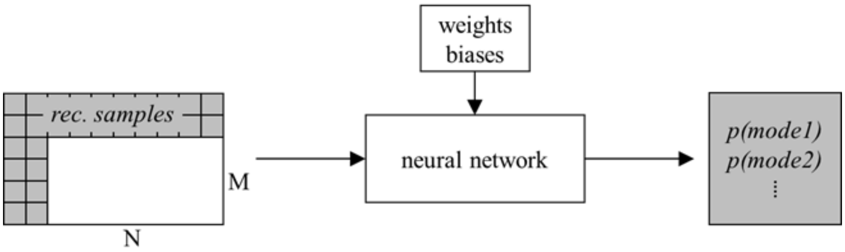

Figure 2. Prediction of mode probabilities from reconstructed samples using a neural network.

From the index $predIdx$, the actual index $predmode$ that determines the mode to be used is derived using a fully connected neural network with one hidden layer that has the reconstructed samples r' on the two rows of size N+2 above and the two columns of size M left of the block as input, see Figure 2.

The reconstructed samples r' are regarded as a vector in the real vector space of dimension 2*(M+N+2). There is a fixed square-matrix A_1' which has 2*(M+N+2) rows resp. columns and there is a fixed bias vector b_1' in the real vector space of dimension 2*(M+N+2) such that one computes

$$t_1{}' = \rho(A_1{}' \cdot r' + b_1{}').$$

Next, there exists a matrix A_2' which has n rows and 2*(M+N+2) columns and there is a fixed bias vector b_2' in the real vector space of dimension n such that one computes

$$lgt = A_2' \cdot t_1' + b_2' .$$

The index $predmode$ is now derived as being the position of the $predIdx$-th largest component of lgt. Here, if two components $(lgt)_k$ and $(lgt)_l$ are equal for k≠l , $(lgt)_k$ is regarded as larger than $(lgt)_l$ if $k < l$ and $(lgt)_l$ is regarded as larger than $(lgt)_k$, else. The derivation of the index $predmode$ proposed in this section is depicted on the left-hand side of Figure 3Figure 3 below.

To explain the above derivation of the prediction mode, we remark that by means of the softmax-function, the i-th entry lgt_i of the vector lgt is to be interpreted, up to normalization, as the logarithm of the conditional probability $p(i|r')$ of the i-th proposed intra prediction mode given the reference samples r'. Namely, one has

$$p(i|r') = \frac{\exp(lgt_i)}{\exp(lgt_0) + \cdots + \exp(\, lgt_n)}.$$

Thus, the index $predIdx$ indicates that the $predIdx$-th most probable mode is selected

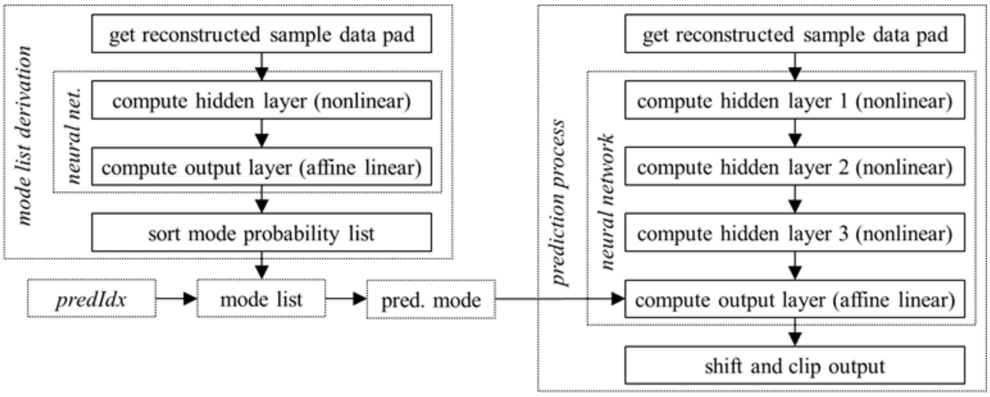
mode list derivation
neural net.
get reconstructed sample data pad
compute hidden layer (nonlinear)
compute output layer (affine linear)
sort mode probability list
predIdx
mode list
pred. mode
prediction process
neural network
get reconstructed sample data pad
compute hidden layer 1 (nonlinear)
compute hidden layer 2 (nonlinear)
compute hidden layer 3 (nonlinear)
compute output layer (affine linear)
shift and clip output

Figure 3. Block diagram of decoder-side reconstruction process ($predIdx$ is sent in bit stream).

We remark that one would be tempted to use the conditional probability $p(i|r')$ directly in the entropy coding of our prediction mode by feeding it into the underlying arithmetic coding engine. However, this would result in a parsing dependency of symbols which is commonly regarded as highly undesirable since it makes impossible any error resilience: An error in computing the reconstructed samples r' could lead to an error in the estimation of the probability $p(i|r')$ which in turn could lead to an error in parsing all symbols from the bitstream.

3 EXPERIMENTAL RESULTS

The intra prediction modes described in the previous section were integrated in a software that was equivalent to the HEVC reference software anchor with the extension that it also supported non-square partitions, namely blocks of size MxN, M and N being integral powers of two with $4 \leq \min(M, N)$ and $64 \geq \max(M, N)$. This partitioning method, described in [5] is essentially equivalent to the QTBT-partitioning method proposed in the JEM-reference software [6].

Our prediction modes were integrated as complementary to the HEVC intra prediction modes. Thus, for each coding unit in intra mode, a flag indicating whether one of the prediction modes described in this paper is to be used or not was sent in the bitstream. If this flag was set true, the prediction signal was generated as described in section 2. Our method to generate a prediction signal was applied to the luma component only.

Five different quantization parameters (QPs) have been tested ranging from 22 to 42 and two BD-rate values have been calculated. The first BD-rate calculation employed the QPs in the set [22; 27; 32; 37] representing the high operation points, referred to as High Tier, while the BD-rate calculation using the QPs in the set [27; 32; 37; 42] represents the low operation points, referred to as Main Tier. All test sequences described in the JVET common test conditions [3] were used. Additionally, the HDR and 360° content proposed in the JVET call for proposals [4] were included.

Table 1. BD-rate savings achieved by the proposed method in the AI configuration.

Sequence Category	Sequence Name	Bit Rate Savings in % for High Tier	Bit Rate Savings in % for Main Tier
360° 4K	Balboa	-3,63	-3,55
	Chairlift Ride	-3,46	-3,59
	Harbor	-2,79	-2,58
	Kite File	-4,59	-4,33
	Trolley	-2,97	-2,71
HDR 4K	Cosmos	-1,34	-1,34
	Hurdles	-1,14	-0,25
	Market	-1,76	-1,56
	Show Girl	-4,77	-4,16
	Starting	-1,84	-1,42
	Day Street	-4,10	-4,10
	People In Shopping Center	-4,77	-4,56
	Sunset Beach	-2,75	-2,87
Class A1 4K	Campfire Party	-0,37	-0,10
	Drums	-2,95	-3,43
	Tango	-4,53	-4,72
	Toddler Fountain	-4,43	-4,26
Class A2 4K	Cat Robot	-2,27	-1,73
	Daylight Road	-3,02	-3,46
	Rollercoaster	-3,27	-3,12
	Traffic Flow	-3,55	-3,65
Class A3 4K	Food Market	-6,50	-6,63
	Park Running	-1,74	-1,95

Class B	Kimono	-3,61	-4,00
	Park Scene	-4,33	-4,50
	Cactus	-2,70	-2,44
	Basketball Drive	-1,89	-1,63
	BQ Terrace	-2,34	-2,74
Class B1	Ritual Dance	-4,48	-4,74
	Market Place	-3,82	-4,10
Class C	Basketball Drill	-0,15	-0,27
	BQ Mall	-2,22	-2,57
	Party Scene	-1,71	-1,90
	Race Horse	-2,08	-2,05
Class D	Basketball Pass	-1,07	-0,65
	BQ Square	-2,54	-3,20
	Blowing Bubbles	-1,97	-2,06
	Race Horses	-2,98	-3,16
Class E	Four People	-4,67	-4,32
	Johnny	-3,93	-3,94
	Kisten and Sarah	-4,18	-4,23
Average Bit Rate Savings in %		-3,01	-2,99

The measured average encoding time for the High Tier configuration was 218%, the decoding time was 248%. The encoding time for the Main Tier configuration was 241%, the decoding time was 286%.

4 Related work and outlook

In the paper [7], for each of the block shapes 4×4, 8×8, 16×16 and 32×32, one or two intra prediction modes are proposed which are also represented by fully connected neural networks with three hidden layers. While less modes (one or two modes) for each block-size are proposed in that paper, the complexity to generate a prediction signal is higher there than in the present case.

For example, for blocks of size 32× 32, according to section 2, in the present case all three hidden layers have dimension 132, i.e. each of the matrices A_1, A_2 and A_3 is a 132× 132-square matrix. Moreover, for a fixed mode indexed by k, the matrix $A_{4,k}$ is a 1024×132 matrix. On the other hand, in the paper [7], for blocks of size 32× 32, as an input 8 lines left and above the block are taken which amounts to an input size of 576. Moreover, in [7], the dimension of the hidden layers except the last one is set to 2048. Thus, the matrix A_1 would be a 2048×576 matrix. The matrices A_2 and A_3 would be 2048×2048 matrices and the matrix A_4 representing the last layer would be a 1024×2048 matrix. For the lighter model presented in [7], the dimension of the hidden layers is reduced to 256. Thus, the matrix A_1 would be a 256×576 matrix, the matrices A_2 and A_3 would be 256×256 matrices and the matrix A_4 would be a 1024×256 matrix.

In the present paper, we tried to limit the computational burden of each individual intra prediction mode by introducing a variety of new modes. In the training process for our modes, this implies the challenge to perform a clustering. Moreover, by introducing more modes, the signalization scheme for each mode becomes a significant problem that we tried to deal with as described in section 2.2.

However, we remark that our prediction modes become more complex the larger the blocks are. This is in particular true for the generation of the output layer, i.e., for computing the matrix vector product $A_{4,k} \cdot ftr$: While for blocks of size 4×4, in the present case 20 multiplications per pixel need to be carried out to generate the output layer, for blocks of size 32×32, we need to carry out 132 multiplications per pixel to generate the output layer.

Thus, instead of predicting into the sample domain, in subsequent work we designed predictors that predict into the frequency domain each following a fixed sparsity pattern: For a lot of frequency components, the prediction signal is constrained to zero in that component, independent of the input. For each such frequency component, the row of the matrix $A_{4,k}$ corresponding to that component consists only of zeros and no multiplications need to be carried out in the matrix vector product $A_{4,k} \cdot ftr$ for that row. This simplification shall be, among other things, discussed in a subsequent paper.

REFERENCES

[1] W. Han G. J. Sullivan, J.-R. Ohm and T. 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] V. Sze, M. Budagavi, G. Sullivan, “High Efficiency Video Coding (HEVC)”, Integrated Circuits and Systems, Springer Publishing House, 2014.

[3] K. Suehring, X. Li, “JVET common test conditions and software reference configurations,” JVET-H1010, October 2017.

[4] A. Segall, V. Baroncini, J. Boyce, J. Chen, T. Suzuki “Joint Call for Proposals on Video Compression with Capability beyond HEV”, JVET-H1003, October 2017.

[5] J. Ma et al., “Quadtree plus binary tree with shifting”, JVET-J0035, April 2018.

[6] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, J. Boyce, “Algorithm Description of Joint Exploration Test Model 7 (JEM 7) “, JVET-G1001, July 2017.

[7] J. Li, B. Li, J. Xu, R. Xion, “Fully Connected Network-Based Intra-Prediction for Image-Coding”, IEEE Transactions on Image Processing, Volume 27, Issue 7, July 2018