CONVOLUTIONAL NEURAL NETWORKS FOR VIDEO INTRA PREDICTION USING CROSS-COMPONENT ADAPTATION

Maria Meyer, Jonathan Wiesner, Jens Schneider, Christian Rohlfing

Institut für Nachrichtentechnik, RWTH Aachen University
Melatener Str. 23, 52074 Aachen, Germany
meyer@ient.rwth-aachen.de

ABSTRACT

Recently, neural networks were shown to improve video and image intra prediction significantly. In this paper, the properties of different architectures for neural network-based intra prediction are evaluated. This includes an analysis of the properties of convolutional neural networks used for this purpose, showing that they outperform fully connected ones especially for complex and low resolution content. Also, the usage of separate networks for luma and chroma prediction, which are able to perform a learned cross-component prediction, is proposed as this is clearly beneficial for the prediction quality. Furthermore, a new way of signaling a neural network-based intra prediction mode in HEVC is investigated. In total this improves the compression performance in terms of average BD-rate changes by -2.0% for the luma and by -1.5% for the chroma channels.

***Index Terms*—** intra prediction, convolutional neural networks, compression, chroma, cross-component prediction

1. INTRODUCTION

Most image and video coding algorithms use intra prediction to reduce the spatial redundancy within a picture or frame before transform coding the prediction residual. In current video compression standards such as HEVC [1], this prediction is performed by applying different versions of linear combinations to a reference sample set that contains the already decoded directly neighboring pixels. However, due to increasing hardware capabilities more complex methods for intra prediction have recently been developed, including neural network-based approaches.

Among the conventional, recently proposed methods are adding more [2] and wider [3] angles to the prediction modes, the usage of more reference lines [4], [3] and an improved cross-component prediction method that regards the available luma information of the same block when generating the chroma prediction [5], [6].

Conceptionally, intra prediction is very similar to inpainting tasks, for which neural network-based methods as in [7], [8] and [9] are already state-of-the-art. Unfortunately, there are two very important differences that hinder the direct application of these same methods for compression. First, hard complexity restrictions are required for real-time video decoding. Second, inpainting aims to generate a patch that looks reasonable to a human viewer, while for compression the difference between the original and its prediction needs to be minimized. Especially, in regions where several reasonable continuations of a context are equally probable, the best solution minimizing the expected coding costs of the residual is usually not looking reasonable to humans, but returning the statistically expected value for each samples.

A first approach for neural network-based intra prediction for video coding was published in [10] and improved in [11], showing significant BD-rate gains compared to HEVC. Interestingly, this method relies exclusively on fully connected neural networks although convolutional architectures have outperformed those in most other image related tasks. More recently, in [12] a network was tested that contains both convolutional and transpose convolutional layers, while in [13] recurrent networks were investigated for intra prediction. Both approaches also give further insights into preprocessing and training methods suitable for networks with this purpose.

Another approach using very shallow and likewise exclusively fully connected networks was proposed in [14] and further refined in [15], [16] and [17]. This method showed that significant gains can also be achieved as an addition to the VTM reference software [18] and that this is possible with a very small decoding time increase. From these results it can be concluded that neural networks can be beneficial for intra prediction even with hard complexity constraints. However, there are still a variety of issues related to the use of neural networks for this task that have not yet been investigated.

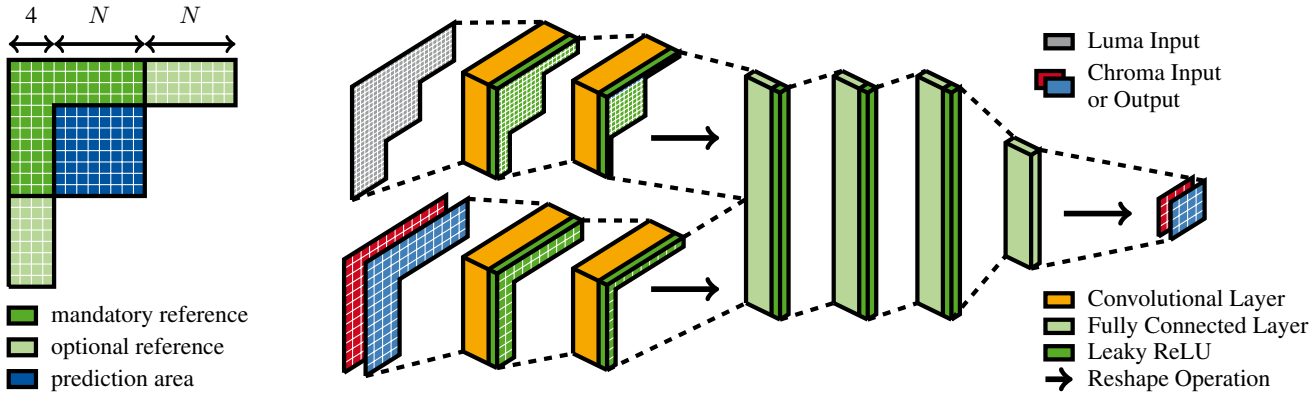
From the existing approaches introduced so far, it is hard to tell which network architecture is suited best, as most of them are trained on different training material which usually has a significant impact on the results. Only the authors of [13] compare the performance of their recurrent architecture to that of a fully connected network trained on the same set. However, this test was restricted to the relatively small block size of 8x8 samples. Therefore, it cannot show how the different architectures would perform with larger, more complex reference areas.

In addition, so far all approaches are either aiming only at the prediction of luma blocks [14] or generate the prediction for all channels with the same network [11] disregarding both the different statistics of chroma channels and the additional available information [6].

In this paper we propose a convolutional architecture for intra prediction that unlike the algorithm in [12] does not use transpose convolutional layers and compare its performance to that of a purely fully connected network trained on the same data. We extend the analysis of optimal loss functions from [13] by comparing the results of the SATD loss with the L1 norm and propose a novel approach for signaling the usage of a network base prediction mode. Likewise, the usage of separately trained networks for the chroma prediction integrating the cross-component information is evaluated.

2. PREDICTION NETWORKS

In HEVC [1], the intra prediction is performed at four different block sizes $N \in \{4, 8, 16, 32\}$ and separately for all three channels. We chose to predict the two chroma channels jointly, as the information available at the decoder for these predictions are identical and this reduces complexity. This leaves a total of eight different cases for which a prediction network needs to be designed and trained.



(a) Reference and prediction area for a NxN luma block

(b) Example architecture for the prediction of a 4x4 chroma block with two convolutional layers (C2) and the CRCO cross-component branch

Fig. 1: Illustration of the reference area and network architecture

2.1. Architecture

Both exclusively fully connected and convolutional neural networks were tested for all eight cases, in order to investigate the possible gains at various complexity levels as well as the property differences caused by the addition of convolutional layers. As the last four layers of the convolutional architectures used by us are unlike in [12] fully connected, the exclusively fully connected architecture (C0) can be seen as a variant of the convolutional one with zero convolutional layers.

In the luma case, we use four reference lines as a trade-off between the additional precision gained and the increasing complexity caused by using more lines. The length of the reference lines is kept at $2N$ as in HEVC, leading to the L-shaped reference area shown in Figure 1a. As we do not use any padding, the number of reference lines limits the possible kernel sizes and combinations thereof. Pretests showed that the combination (C2) of a 3x3 kernel as a first convolutional layer and a 2x2 kernel as second layer outperformed all other combinations. Using a 4x4 kernel (C1) as a single convolutional layer performed slightly worse, but is also by far less complex. Thus, these two architectures were further investigated. For all but the last layer a leaky rectified linear unit (ReLU) is used as activation function. The output layer has no activation function and as many nodes as the number of pixels to be predicted.

In the chroma case, the prediction of both chroma channels is generated jointly. Thus, the input is also used from both channels and the output has twice the number of nodes than the luma version. Otherwise, the network architecture is the same as for the luma case. Further, it was already shown for traditional prediction methods that including the information from the luma channel in predicting chroma blocks gives significant gains [6]. Hence, an additional option (CRCO) to include cross-component prediction into the network was implemented. As in the commonly used 4:2:0 subsampling the chroma channels have a lower resolution than the luma channel, these samples could not simply be fed into the network as a third input channel to the existing input. Instead, they are processed in a separate convolutional branch up to the first fully connected layer, as shown in Figure 1b. In all architecture variants the additional luma branch has as many convolutional layers as the main one.

2.2. Training

All networks are trained on samples from 104 sequences with varying resolutions applying the Adam-optimization algorithm [19], while using eleven additional videos to generate a validation set. As the number of training examples that can be extracted from these sequences decreases with increasing block size, overfitting effects occur more severely for large block sizes and chroma prediction networks than for smaller luma prediction blocks. Thus, it can be expected that an even larger training set would still strongly increase the prediction quality. To decrease the similarity of samples from consecutive frames, both horizontal and vertical flipping can be applied during preprocessing. Likewise, the channel-wise mean of the reference area is subtracted from both the reference and the prediction area. Furthermore, some samples with low variance are excluded from training, if too many samples of roughly the same variance are already used. This makes the network adapt better to high variance samples, which are less frequent in most videos, but are especially difficult to predict and thus cause high bit rates. A full description of the used training and validation sets as well as a list of further training and architecture hyperparameters can be found on the website accompanying this paper[1].

Depending on the position of the block to be predicted, the actually available reference area varies. In most cases, the mandatory reference area as marked in Figure 1a should be available for prediction, if the block is not at the upper or left boundary of a slice or frame. The availability of the area marked as optional reference depends on the position within the CTU. In order to use this optional reference whenever possible without training different netwoks for each availability case a masking scheme similar to the one presented in [12] was applied. In our case each of the two optional areas is either completely masked or completely available.

[1] http://www.ient.rwth-aachen.de/cms/icassp2019/

Loss Function	L1	SATD
BQTerrace	-1.51 %	**-1.61** %
BasketballDrive	-1.97 %	**-2.30** %
Cactus	-2.08 %	**-2.30** %
Kimono	-2.61 %	**-3.17** %
ParkScene	-2.75 %	**-2.85** %
AVG Class B	-2.18 %	**-2.45** %

Table 1: BD-rate change for the Y channel when using network-based intra prediction with different loss function compared to HEVC. Both versions use the C0 architecture with CRCO and the UP signaling mode.

It was already shown in [13], that the SATD loss function with a Hadamard transform performs better than the commonly used MSE loss, when restricting the block size to 8x8. We additionally compared the performance of the SATD loss in comparison to the L1 norm in an unconstrained video compression test. The L1 norm was chosen as it already puts more weight on structures instead of outliers than MSE. As shown in Table 1, networks trained with the SATD loss outperforms the other version on every tested sequence. Hence, it was used in all tests conducted in section 4. The result confirms that the SATD function provides the best estimation of the coding costs of the prediction residual among the functions tested so far. Even closer approximations of the actual coding costs for the transform coefficients are difficult to set up as loss functions as these have to be derivable. Note that in all cases a small weight regularization term was added to the loss function in order to reduce overfitting.

3. MODE SIGNALING AND HM INTEGRATION

The networks described in Section 2 were integrated into the HM16.9 reference software [20] as a 36th intra prediction mode (IntraNN). In order to efficiently signal the usage of this mode for the luma prediction, the most probable mode list was extended to hold a fourth option by sending an additional bit when the third list position would have been chosen otherwise. The IntraNN mode is always placed on that list. If it was used in one of the neighboring blocks, it is put in the position for that neighbor and the remaining list is filled following the same rules as if there were only three list positions and the neighbor using IntraNN had not been available. Two different options were tested varying in where to place the IntraNN mode, when it was not used in the neighborhood. In the version END it is always placed in the last position, while in the UP option the IntraNN mode is placed directly behind the modes used in the neighborhood. In both cases the other list positions are likewise filled according to the HEVC specifications.

For chroma prediction, the IntraNN mode can only be signaled by specifying to use the same mode as for luma. Thus, it can only be applied for chroma, if it has been used for the luma prediction of that block as well. In order to evaluate the benefit of using a network explicitly for chroma prediction, a version where IntraNN is only applied to the luma channel as in [14] was also tested. In this version the chroma intra mode will be signaled as if the corresponding luma prediction unit would be predicted with planar mode.

For both luma and chroma prediction, the IntraNN mode is evaluated during the rate-distortion optimization in the same way as other intra modes. However, the IntraNN mode will not be used, if any part of the mandatory reference is unavailable for prediction. The larger mandatory reference area in the chroma case can cause the IntraNN mode to be chosen for the luma channel of a prediction unit while being unavailable for the chroma prediction of the same block. This can lead to a decrease in the possible number of chroma prediction modes. Note that neither the inclusion of the luma samples in the chroma prediction nor any of the architecture variants presented here have any effect on the signaling or mode availability.

4. EXPERIMENTS AND RESULTS

All coding experiments presented in this section were performed on the first 100 frames from the classes B, C and D of the common testing conditions [21] in all intra mode.

4.1. Network Architecture Comparison

In a first test the three different architectures with varying numbers of convolutional layers (C0, C1 and C2) were compared to each other.

Architecture	C2	C1	C0
BQTerrace	**-1.79** %	-1.74 %	-1.61 %
BasketballDrive	**-2.33** %	-2.28 %	-2.30 %
Cactus	**-2.46** %	-2.43 %	-2.30 %
Kimono	-2.66 %	-3.02 %	**-3.17** %
ParkScene	-2.55 %	-2.66 %	**-2.85** %
AVG Class B	-2.36 %	-2.43 %	**-2.45** %
BQMall	**-2.00** %	-1.85 %	-1.85 %
BasketballDrill	**-1.99** %	-1.96 %	-1.81 %
PartyScene	**-1.46** %	-1.39 %	-1.34 %
RaceHorses	**-1.89** %	-1.84 %	-1.75 %
AVG Class C	**-1.84** %	-1.76 %	-1.69 %
BQSquare	**-0.98** %	-0.88 %	-0.79 %
BasketballPass	**-1.85** %	-1.51 %	-1.49 %
BlowingBubbles	-1.70 %	**-1.74** %	-1.63 %
RaceHorses	**-2.43** %	-2.30 %	-2.00 %
AVG Class D	**-1.74** %	-1.61 %	-1.48 %
AVG All Classes	**-2.01** %	-1.97 %	-1.91 %

Table 2: BD-rate change for the Y channel compared to HEVC for different architectures. In all three cases the SATD loss function, CRCO integration and the UP signaling mode were used.

As can be seen in Table 2, the C2 architecture outperforms the other architectures in terms of their BD-rate gains [22] on average. However, that does not hold true for all tested resolutions. While the architectures with convolutional layers outperform the fully connected variant by a comparably large margin on the lowest resolution, the C0 version gives much better results on the HD sequences "Kimono" and "ParkScene" and thus also on average on class B. An explanation for this behavior could be, that the convolutional networks capture the high variance content and complicated texture more often present in blocks from low resolutions sequences better, but are therefore also more easily disturbed by noise.

In [12] it was stated that convolutional networks work better for the prediction of larger blocks. This hypothesis was mainly based on the theoretical consideration, that convolutional networks are usually better at processing stationary and multi-resolution structures possibly occurring in the reference of larger blocks. However, when analyzing our results for class B in more detail, we found that the C0 network is chosen more frequently than the C2 type for the prediction of 32x32 luma blocks in every video, on average by 8.7%, while C2 network is 3.3% more often used for luma 4x4 blocks, which is the exact opposite of the expected behavior. This result is even more surprising when considering that the SATD loss on the validation set is always lower for the convolutional versions and that this difference increases with the block size. It indicates that convolutional architectures are better at predicting the more complex structures for which smaller block sizes are chosen by the RD-optimization.

On the other hand, it has to be noted that the architectures with convolutional layers are more complex than the fully connected ones. Averaged over all block sizes the C1 version increases the number of multiplications by a factor of 3.0 for the luma prediction compared to the C0 version. For the C2 version this is even an increase of 4.2 times. Thus, it depends on both the acceptable complexity and the properties of the content, which architecture is best suited for an intra prediction task.

4.2. Signaling Evaluation

In a second experiment, the positioning of the IntraNN mode on the most probable mode list was evaluated. As described in Section 3 two versions were tested. In principal, the END signaling version should cause less overhead when the IntraNN mode is not chosen,

Version	END, with CRCO			UP, with CRCO			Up, without CRCO			UP, no chroma IntraNN		
Channel	**Y**	**U**	**V**	**Y**	**U**	**V**	**Y**	**U**	**V**	**Y**	**U**	**V**
BQTerrace	-1.75%	-0.69%	-0.76%	**-1.79%**	**-0.84%**	-0.36%	-1.66%	-0.75%	**-0.79%**	-1.57%	-0.26%	-0.03%
Basket.Drive	-2.24%	**-1.64%**	**-2.08%**	**-2.33%**	**-1.64%**	-1.97%	-1.83%	-0.15%	-0.88%	-1.34%	-0.05%	-0.56%
Cactus	-2.35%	**-1.95%**	**-2.06%**	**-2.46%**	-1.89%	-2.05%	-1.99%	-1.24%	-1.12%	-1.60%	-0.89%	-0.50%
Kimono	-2.42%	-2.33%	-1.75%	**-2.66%**	**-2.46%**	**-1.84%**	-1.71%	-1.60%	-1.41%	-1.62%	-1.46%	-1.26%
ParkScene	-2.44%	-1.46%	**-1.99%**	**-2.55%**	**-1.75%**	-1.91%	-1.87%	-1.27%	-1.82%	-1.88%	-0.79%	-1.15%
AVG Class B	-2.24%	-1.61%	**-1.73%**	**-2.36%**	**-1.72%**	-1.63%	-1.81%	-1.00%	-1.20%	-1.59%	-0.69%	-0.62%
BQMall	-1.97%	**-1.63%**	-1.56%	**-2.00%**	-1.62%	**-1.57%**	-1.71%	-1.22%	-1.05%	-1.58%	-1.37%	-0.36%
BasketballDrill	**-2.00%**	-2.17%	-2.03%	-1.99%	**-2.42%**	**-2.26%**	-1.21%	-0.38%	-0.74%	-0.63%	-0.17%	-0.21%
PartyScene	**-1.46%**	-0.83%	-0.95%	**-1.46%**	**-0.86%**	**-0.96%**	-1.31%	-0.68%	-0.70%	-1.21%	-0.68%	-0.65%
RaceHorses	-1.84%	-1.20%	**-1.66%**	**-1.89%**	**-1.28%**	-1.44%	-1.55%	-0.90%	-0.60%	-1.22%	-0.69%	-0.49%
AVG Class C	-1.82%	-1.46%	-1.55%	**-1.84%**	**-1.55%**	**-1.56%**	-1.45%	-0.80%	-0.77%	-1.16%	-0.73%	-0.43%
BQSquare	-1.00%	-0.56%	-0.01%	-0.98%	**-0.65%**	**-0.28%**	**-1.04%**	-0.55%	0.00%	-1.00%	-0.28%	0.20%
BasketballPass	-1.78%	**-1.76%**	-1.19%	**-1.85%**	-1.67%	**-1.36%**	-1.39%	-1.37%	-0.82%	-1.21%	-0.26%	-0.58%
BlowingBubbles	-1.69%	**-1.74%**	-0.71%	**-1.70%**	-1.54%	**-0.97%**	-1.40%	-0.60%	-0.43%	-1.32%	-0.58%	-0.37%
RaceHorses	-2.35%	**-1.92%**	**-2.14%**	**-2.43%**	-1.40%	-2.13%	-1.91%	-1.34%	-1.34%	-1.58%	-0.79%	-0.78%
AVG Class D	-1.71%	**-1.50%**	-1.01%	**-1.74%**	-1.32%	**-1.19%**	-1.44%	-0.97%	-0.65%	-1.28%	-0.48%	-0.38%
AVG All Classes	-1.94%	-1.53%	-1.45%	**-2.01%**	**-1.54%**	**-1.47%**	-1.58%	-0.93%	-0.90%	-1.37%	-0.57%	-0.52 %

Table 3: BD-rate change compared to HEVC for the different signaling modes (column 1 and 2), without the cross-component integration (column 3) and without IntraNN for the chroma channel (column 4). In all cases the SATD loss function was applied for training and the C2 architecture was used.

while in the UP version the signaling costs for the IntraNN mode itself are lower. The results are shown in the first two columns of Table 3. In most cases, the UP variant gives slightly better BD-rate gains than the END version especially for the luma channel. Note that although there are additional signaling costs caused in both versions even if the IntraNN mode is not used, there is no sequence with any loss. However, the END version comes much closer to that than the UP version. This clearly shows that the IntraNN mode is chosen often enough that the costs for the usage of the IntraNN mode outweighs the additional costs for choosing other modes caused by the signaling scheme.

4.3. Dedicated Chroma Prediction

Finally, the effects of using a neural network-based chroma prediction as well as the integration of the cross-component information into this process were investigated. Therefore, a version where IntraNN is not used for the chroma channels is compared with versions where the chroma prediction is done with or without the CRCO branch. The resulting BD-rates are shown in the second, third and fourth column of Table 3. It can be seen, that the cross-component version outperforms the version not using network-based chroma prediction on every channel and sequence, on average by -0.6% on the luma and by -0.97% and -0.95% on the chroma channels.

The version which includes the network-based chroma prediction without CRCO likewise outperforms the version without the network-based chroma prediction on average for all resolutions and channels, but performs still clearly worse than the version with the cross-component prediction in most cases. On average, the BD-rate gain increases due to the cross-component mode by -0.43% for luma and -0.59% for chroma. On the other hand, using the cross-component prediction with the C2 architecture leads to an average of 12.15 times more multiplications than the version without the additional branch. Thus, again the most complex network clearly gives the best prediction and resulting BD-rate gains.

4.4. Comparison to Existing Methods

Overall, the BD-rate gains on the Y channel achieved by the approach presented here are still lower than those presented in [11], [13] and [12]. On average these methods respectively perform 0.05%, 0.27% and 1.05% better in terms of the Y BD-rate than IntraNN for classes B, C and D. It can however not be concluded, if this difference is due to architecture choices and training settings or simply due to the different training material, as each of these approaches uses completely different sets. Nonetheless, IntraNN outperforms those other approaches, that give results for the U and V channel BD-rates, [11] and [13], by at least 0.05% on the U and 0.29% on the V channel. This indicates that the dedicated chroma networks and cross-component integration clearly improve the prediction quality and the resulting PSNR for these channels.

5. CONCLUSION AND OUTLOOK

It was shown in this paper, that it is useful to train separate intra prediction networks for the chroma channels, which are able to benefit from the information in the luma component of the respective block and thus perform a learned cross-component prediction. This increases the luma BD-rate gains from -1.4% to -2.0% on average and outperforms state-of-the-art methods regarding the gains on the chroma channels. Also, the advantages and problems of convolutional network architectures were evaluated with the conclusion that these are especially useful for complex contexts, small block sizes and low resolutions. In addition, a new signaling scheme for the network-based intra prediction mode was proposed and the benefit of the SATD loss function for training prediction networks was confirmed.

However, as this is still a relativly new field of resarch, there are still numerous possibilities to enhance the current method. Most obvious is the use of more training material and better data augmentation, as so far more training samples have always significantly improved the prediction accuracy. Likewise, the use of references with coding artifacts during training will most probably increase the prediction quality especially for low bitrates. Modified masks to allow for partially available reference areas and the use of multiple network-based predictions similar to [11] or [14] will most likely improve the results further as well. At the same time it remains to be investigated how far the needed computational complexity during inference can be reduced by quantization, pruning and similar reduction techniques.

6. REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, 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, 2012.

[2] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 7 (JEM 7)," Doc. JVET-A1001-v1, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, July 2017.

[3] G. Van der Auwera, J. Heo, and A. Filippov, "Description of core experiment 3 (CE3): Intra prediction and mode coding," Doc. JVET-J1023-v1, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, Apr. 2018.

[4] J. Li, B. Li, J. Xu, and R. Xiong, "Intra prediction using multiple reference lines for video coding," in *Data Compression Conference (DCC)*, April 2017, pp. 221–230.

[5] G. Van der Auwera, J. Heo, and A. Filippov, "Description of core experiment 3 (CE3): Intra prediction and mode coding," Doc. JVET-L1023-v1, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, Oct. 2018.

[6] K. Zhang, J. Chen, L. Zhang, X. Li, and M. Karczewicz, "Enhanced cross-component linear model for chroma intra-prediction in video coding," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3983–3997, Aug 2018.

[7] L. Theis and M. Bethge, "Generative image modeling using spatial LSTMs," in *Advances in Neural Information Processing Systems 28*, pp. 1927–1935. Curran Associates, Inc., 2015.

[8] R. A. Yeh, C. Chen, T.-Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," *Computing Research Repository (CoRR)*, 2016.

[9] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *Computing Research Repository (CoRR)*, 2016.

[10] J. Li, B. Li, J. Xu, and R. Xiong, "Intra prediction using fully connected network for video coding," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017.

[11] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, July 2018.

[12] T. Dumas, A. Roumy, and C. Guillemot, "Context-adaptive neural network based prediction for image compression," 2018.

[13] Y. Hu, W. Yang, M. Li, and J. Liu, "Progressive spatial recurrent neural network for intra prediction," *Computing Research Repository (CoRR)*, 2018.

[14] J. Pfaff, P. Helle, D. Maniry, S. Kaltenstadler, B. Stallenberger, P. Merkle, M. Siekmann, H. Schwarz, D. Marpe, and T. Wiegand, "Intra prediction modes based on neural networks," Doc. JVET-J0037-v2, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, Apr. 2018.

[15] P. Helle, T. Hinz, R. Rischke, J. Pfaff, P. Merkle, M. Schäfer, B. Stallenberger, V. George, H. Schwarz, D. Marpe, and T. Wiegand, "CE3-related: Non-linear weighted intra prediction," Doc. JVET-K0196-v2, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, July 2018.

[16] P. Merkle, J. Pfaff, P. Helle, R. Rischke, M. Schäfer, B. Stallenberger, H. Schwarz, D. Marpe, and T. Wiegand, "CE3: Non-linear weighted intra prediction," Doc. JVET-K0266-v2, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, July 2018.

[17] P. Helle, J. Pfaff, M. Schäffer, R. Rischke, T. Hinz, P. Merkle, H. Schwarz, D. Marpe, and T. Wiegand, "CE3: Non-linear weighted intra prediction (tests 2.2.1 and 2.2.2)," Doc. JVET-L0199-v2, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, Oct. 2018.

[18] "VVC test model," https://vcgit.hhi.fraunhofer.de/, 2018.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computing Research Repository (CoRR)*, 2014.

[20] "HEVC test model, 16.9," https://hevc.hhi.fraunhofer.de/, 2016.

[21] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," Doc. JVET-G1010, Joint Video Exploration Team of ITU-T VCEG and ISO/IEC MPEG, July 2017.

[22] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," ITU-T SG16/Q6 VCEG, Austin, USA, VCEG-M33, 2001.