*Proceedings of the 14th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2014
3–7July, 2014.*

Perceptual adaptive QP using a hybrid CNN-ANN model for Versatile Video Coding

Javier Ruiz Atencia[1], Otoniel López Granado[1], Manuel Pérez Malumbres[1] and Miguel Onofre Martínez-Rach[1]

[1] *Computer Engineering department, Miguel Hernández University of Elche*

emails: `javier.ruiza@umh.es`, `otoniel@umh.es`, `mels@umh.es`, `mmrach@umh.es`

Abstract

This paper introduces a hybrid neural network model combining Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs) to optimize the quantization parameter (QP) for 64×64 blocks in the Versatile Video Coding (VVC) standard, enhancing both video quality and compression efficiency. Leveraging CNNs for spatial feature extraction and ANNs for structured data handling, the model addresses the limitations of current heuristic and Just Noticeable Distortion (JND) based methods. The methodology includes generating and preprocessing a dataset of luminance channel image blocks encoded with various QP values and designing a hybrid network with convolutional layers and dense layers. Performance evaluations using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) demonstrate the model's efficacy, achieving significant BD-Rate gains for resolutions, particularly 720p and 1080p, when assessed with WPSNR and MS-SSIM metrics. These results indicate that the proposed model not only improves compression efficiency but also maintains or enhances visual quality, suggesting its practical application in video coding.

Key words: hybrid, CNN, perceptual, QP, VVC

1 Introduction

Video compression plays a crucial role in the efficient storage and transmission of multimedia content. With the advent of high-resolution video formats and the increasing demand for streaming services, developing effective compression techniques has become more important than ever. The Versatile Video Coding (VVC) standard, also known as H.266, is the latest

ISBN: 978-84-616-9216-3

advancement in video compression technology, offering improved coding efficiency compared to its predecessors. However, optimizing the quantization parameter (QP) for individual blocks within a video remains a significant challenge, as it directly impacts the balance between video quality and compression efficiency.

Current methods for determining the QP value often rely on heuristic approaches or traditional Just Noticeable Distortion (JND) models [1]–[4]. While these methods have shown some success, they are limited in their ability to accurately predict the optimal QP value, especially when dealing with diverse video content. Moreover, these methods typically do not leverage the full potential of available data, such as the spatial characteristics of video blocks and other structured information.

To address these limitations, we propose a hybrid neural network model that combines a convolutional neural network (CNN) with structured data inputs to determine the optimal QP value for 64×64 blocks in the VVC standard. Our approach aims to enhance compression prediction accuracy by integrating the strengths of CNNs in capturing spatial features from image data and artificial neural networks (ANNs) in handling structured data.

2 Methodology

This section details the methodology used for designing and evaluating the proposed hybrid model. First, the generation and preprocessing of the dataset consisting of 64×64 pixel image blocks using the luminance channel are described. Next, the architecture of the hybrid neural network, which integrates a convolutional neural network (CNN) for image processing and an artificial neural network (ANN) for handling structured data, is presented. Finally, the normalization and preprocessing techniques applied to the data before being fed into the neural network are explained.

2.1 Dataset preparation

A dataset of 64×64 pixel image blocks, using only the luminance channel, has been developed. This dataset is designed to train and evaluate a hybrid convolutional neural network model. The images were extracted from a series of test sequences conforming to the VVC coding standard, following the specifications outlined in the document [5].

For dataset generation, the VVC (Versatile Video Coding) reference software, known as VTM (VVC Test Model) [6], was used and modified to partition the video exclusively into 64×64 blocks and store them into a CSV (comma-separated values) file. Each video sequence was encoded in All-Intra mode, with a wide range of QP (Quantization Parameter) values from 12 to 47. This value is stored in the dataset as QP_{base} and will be one of the input elements to the neural network.

During the encoding process, each frame is partitioned into 64×64 blocks, and Rate-Distortion Optimization (RDO) is used to decide how to encode in a way that minimizes visual distortion (i.e., loss of quality) while controlling the amount of data needed to represent that block (i.e., bit rate). This cost function is mathematically formalized as follows:

$$\min_{\mathbf{p}_k} \ D_k^{\text{SSE}}(\mathbf{p}_k) + \lambda \cdot R_k(\mathbf{p}_k) \tag{1}$$

where D_k^{SSE} denotes the sum of squared errors (SSE) for a block $\mathbf{B}_k$, $R_k(\mathbf{p}_k)$ is the rate for a block $\mathbf{B}_k$, λ is the Lagrange multiplier, which depends on the QP value, and $\mathbf{p}_k$ is the vector of encoding decisions for the block $\mathbf{B}_k$.

At this stage, further modifications were made to the VTM reference software. In the RDO, a weighted SSE distortion metric based on the WPSNR (Weighted Peak Signal-to-Noise Ratio) [7] was used instead of the conventional SSE distortion measure. Therefore, Equation 1 is modified as follows:

$$\min_{\mathbf{p}_k} \ w_k \cdot D_k^{\text{SSE}}(\mathbf{p}_k) + \lambda \cdot R_k(\mathbf{p}_k) \tag{2}$$

where w_k is the weighting factor for a $\mathbf{B}_k$. In addition, a process of searching for the perceptually optimal QP value has been conducted. For this purpose, we have added a new stage to the encoding process that allows us to use a range of QP values around the QP_{base}. The variable that controls this QP offset is called ΔQP, and it is defined as:

$$\Delta\text{QP} \in \{-6, -5, \ldots, 5, 6\} \tag{3}$$

This means that, for each block, a total of thirteen different QP values are evaluated. For each of these encodings, the weighted RDO is performed (Equation 2), and after all the encoding processes, the ΔQP value that minimizes the cost of the RDO is considered the ground truth and is stored in the dataset, along with the block pixels in the luminance channel. Figure 1 summarizes the entire process described for a given QP_{base} value.

After processing the CSV, the dataset is stored in a pandas DataFrame (Python) with the following columns and data types:

- `QP_base` (int): Initial quantization parameter value.
- `QP_delta` (int): Optimal ΔQP value for the block.
- `BPF` (float): Number of 64 × 64 blocks per frame (BPF). Needed because WPSNR metric is frame size dependent.
- `pix_xxxx` (int): Luminance value of the pixel xxxx.

Here is a sample of the dataset structure:

ISBN: 978-84-616-9216-3

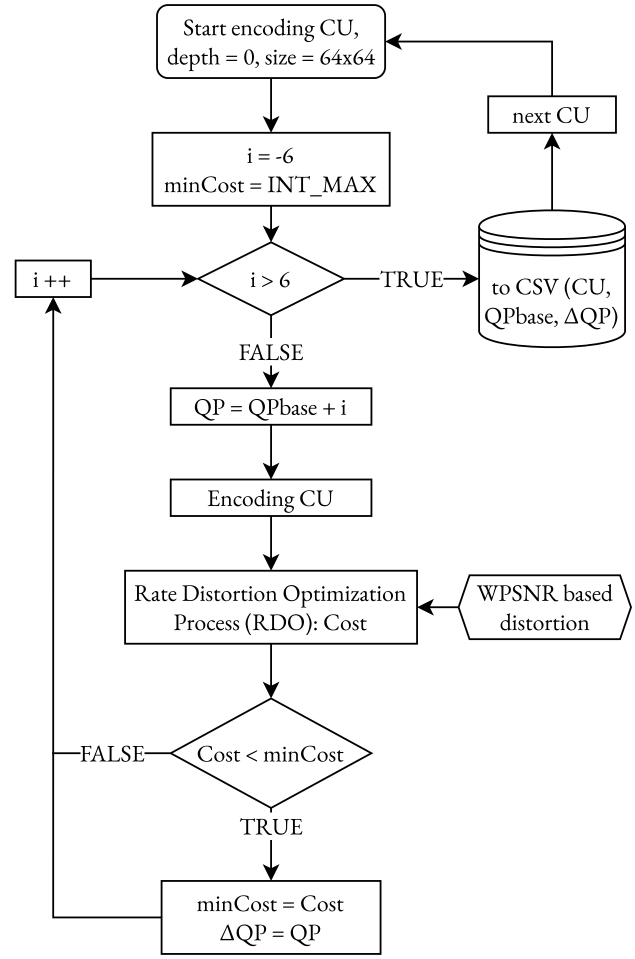


Figure 1: Flow chart of image database extraction.

QP_base	QP_delta	BPF	pix_0001	pix_0002	...	pix_4096	pix_4096
32	-3	225.00	217	218	...	212	215
17	-2	2025.00	115	146	...	26	31
26	3	506.25	101	105	...	29	36
35	-1	2025.00	210	210	...	103	100

Table 1: Example structure of the dataset.

ISBN: 978-84-616-9216-3

2.2 Hybrid neural network proposal

Our proposal model comprises two main subnetworks: a convolutional neural network (CNN) for processing images and an artificial neural network (ANN) for handling structured data. These subnetworks are integrated into dense layers to produce the final output.

The image input is fed into a CNN. Initially, we based our approach on simple, pre-trained CNN models such as EfficientNet [8] and MobileNetV3-Small [9], adapting their architectures to meet our specific requirements. However, our experiments demonstrated that a straightforward neural network consisting of only two convolutional layers converged to a solution with a lower error rate. Consequently, we decided to abandon the use of these pre-trained networks in favor of our simpler, yet more effective, architecture.

To design the architecture of our neural network and to search for optimal hyperparameters, we utilized Keras with TensorFlow as the backend. Additionally, Keras Tuner was employed to perform an extensive hyperparameter search, ensuring that our model configuration was both efficient and effective. This combination allowed us to streamline the development process, leveraging the robust features of Keras and the comprehensive tuning capabilities of Keras Tuner.

The input layer expects images of shape (64, 64, 1). Two convolutional layers (98 and 146 filters with a kernel size of 5×5 and ReLU activation) are used to capture complex features from the images, followed by a MaxPooling layer with a 2×2 window to reduce the dimensionality. A Dropout layer before the second convolutional is applied to prevent overfitting. The output is then flattened into a one-dimensional vector.

Parallel to this, the structured data input is processed through an ANN. The input layer expects data of shape (2,), which is the QP_{base} and the BPF. A dense layer with 42 units and ReLU activation is applied, followed by a Dropout layer.

The outputs from the CNN and ANN are concatenated into a single vector. This combined vector passes through additional dense layers: the first with 134 units and ReLU activation, followed by a Dropout layer, and a second dense layer with 71 units and ReLU activation. The final output layer has a single unit with a linear activation function, where the output, $\Delta\mathrm{QP}$ is constrained within the range $[-6, 6]$. Figure 2 shows the architecture of our proposed model.

Prior to feeding the data into the neural network, the input values were preprocessed. The image blocks were scaled by dividing each pixel value by 255, as we set the input bitdepth to 8 at the encoder configuration (Equation 4). Additionally, the structured data vector was normalized using the StandardScaler method, which standardizes the features by removing the mean and scaling to unit variance (Equation 7).

$$\mathrm{Input}_{pixel} = \frac{\mathrm{Data}_{pixel}}{255} \tag{4}$$

ISBN: 978-84-616-9216-3

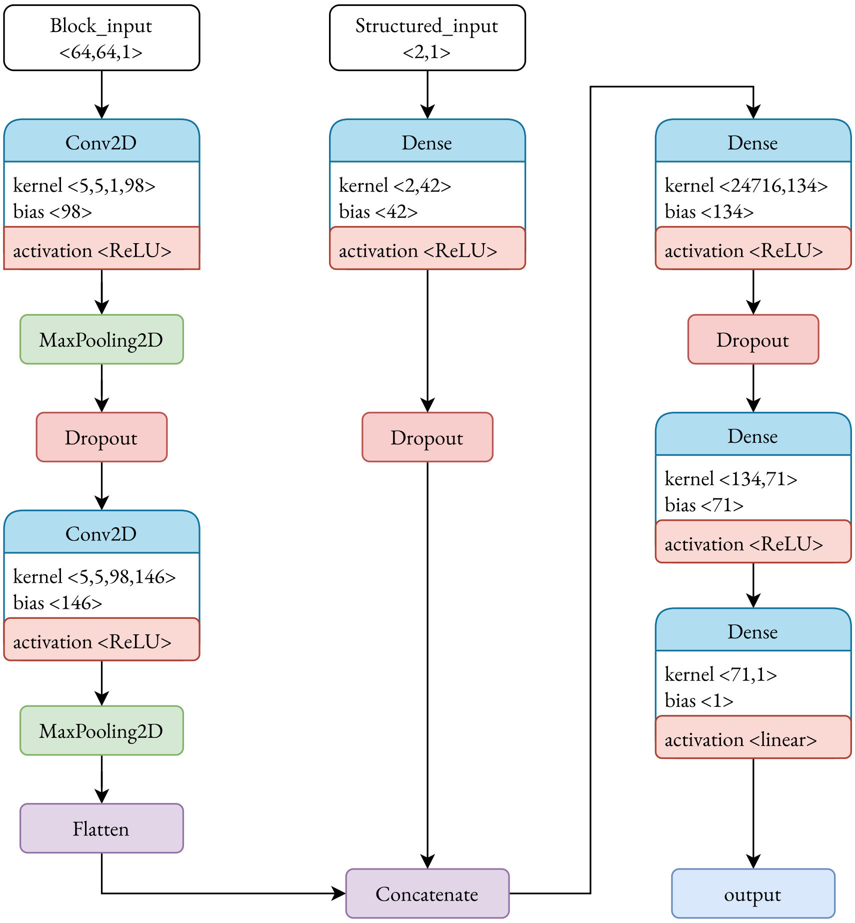


Figure 2: Diagram of our proposed hybrid CNN+Ann model.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \text{Data}_{struct} \tag{5}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{Data}_{struct} - \mu)^2} \tag{6}$$

$$\text{Input}_{struct} = \frac{\text{Data}_{struct} - \mu}{\sigma} \tag{7}$$

ISBN: 978-84-616-9216-3

3 Results and Discussion

In this part, the result of the training of our hybrid neural network will be shown. In addition, the result of its implementation in the VVC reference software will be shown, as well as its comparison with the native perceptual coding algorithm.

To begin, Figure 3 shows the evolution of the training and validation loss of our proposed hybrid model across 100 training epochs. The plot provides valuable insights into our model's performance. Initially, both losses decrease rapidly, indicating effective learning and good generalization of our proposed architecture. However, around epoch 50, while the training loss continues to decrease, the validation loss stabilizes and fluctuates, suggesting overfitting, which negatively impacts its performance on new data.

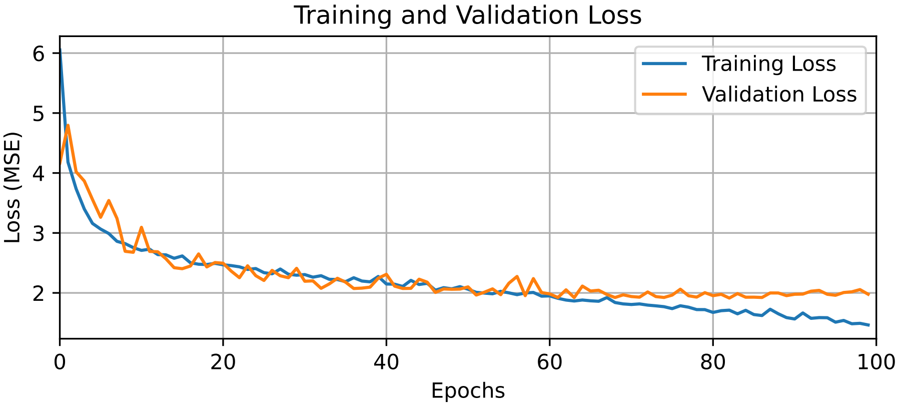


Figure 3: Training and Validation Loss. The plot shows the Mean Squared Error (MSE) loss for both training and validation datasets across 100 epochs.

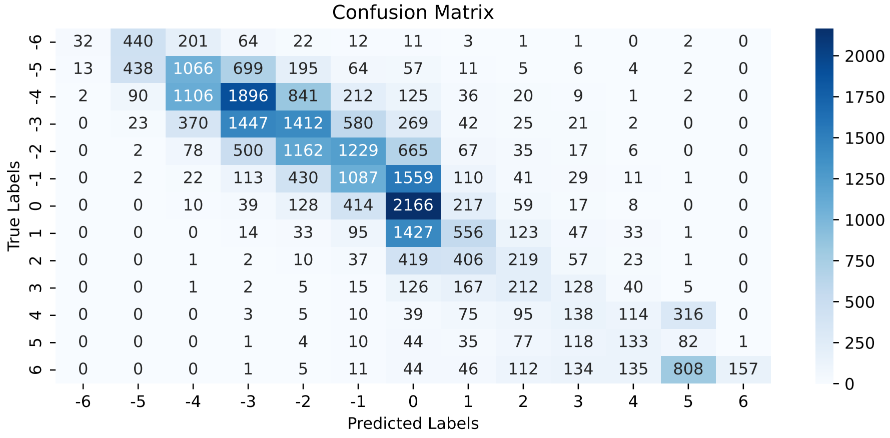
Choosing epoch 53 for the final model strikes a balance between minimizing training loss and avoiding overfitting. At this point, the model benefits from sufficient training without significantly compromising its generalization ability. This careful selection ensures robust performance across unseen blocks of images, optimizing the trade-off between learning and generalization. This can be seen in Table 2, where the different performance metrics, such as MSE, RMS and MAE, applied to the different dataset partitions obtain practically identical results.

The confusion matrix (Figure 4) for the test dataset provides a comprehensive overview of the model's performance. The diagonal elements of the matrix, representing the correctly predicted instances for each class, show high values, indicating that the model performs well in accurately classifying most of the labels. Notably, classes like -1, 0, and 1 have particularly high correct predictions, reflected by the large numbers on the diagonal.

Table 2: Loss values for the model (epoch 39)

	MSE	RMSE	MAE
Train	2.067	1.438	0.994
Validation	2.012	1.418	0.986
Test	2.078	1.442	1.002

Given that the model is not a classification but a regression function, the misclassifications observed in the confusion matrix are minor. Most of misclassified predictions are only one or two positions away from the true labels, which implies that the model's errors are small and localized. This close alignment between true and predicted labels, even when incorrect, suggests that the model maintains a reasonable level of accuracy and that the errors are not drastically off the mark.



	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	32	440	201	64	22	12	11	3	1	1	0	2	0
-5	13	438	1066	699	195	64	57	11	5	6	4	2	0
-4	2	90	1106	1896	841	212	125	36	20	9	1	2	0
-3	0	23	370	1447	1412	580	269	42	25	21	2	0	0
-2	0	2	78	500	1162	1229	665	67	35	17	6	0	0
-1	0	2	22	113	430	1087	1559	110	41	29	11	1	0
0	0	0	10	39	128	414	2166	217	59	17	8	0	0
1	0	0	0	14	33	95	1427	556	123	47	33	1	0
2	0	0	1	2	10	37	419	406	219	57	23	1	0
3	0	0	1	2	5	15	126	167	212	128	40	5	0
4	0	0	0	3	5	10	39	75	95	138	114	316	0
5	0	0	0	1	4	10	44	35	77	118	133	82	1
6	0	0	0	1	5	11	44	46	112	134	135	808	157

Figure 4: Confusion matrix for Test dataset.

Once our neural network model was trained and evaluated, it was integrated into the VVC reference software, VTM, to perform inference on 64 × 64 CU blocks to be encoded. To import the model, we utilized the TensorFlow C API.

Following the integration of our code, we proceeded to evaluate our implementation using the sequences specified in the VTM common test conditions [5], which include video sequences of varying resolutions, from 240p to 4K.

For comparative analysis, we selected the QPA algorithm [7], which derives the ΔQP value on a visual sensitivity measure based on a local energy measure of high-pass filtered original samples. The default QPA algorithm performs two steps. The first one at CTU level (128 × 128 pixels blocks), obtaining a ΔQP_{128}. Then, at CU level (64 × 64 pixels blocks), it obtains another ΔQP_{64}, which uses as base QP the ΔQP_{128} inherited from its CTU. In order to compare our model, based only on blocks of size 64×64, we have disabled the QPA algorithm at the CTU level.

Upon executing the tests for base QP values of 22, 27, 32, 37, and 42, we obtained the following Bjøntegaard delta rate (BD-Rate) [10] values (see Table 3), with executions that did not apply any perceptual mechanism serving as the reference.

Negative values indicate gains, while positive values indicate losses. The table below shows the results for the WPSNR metric, which our model was trained on, and the MS-SSIM metric [11].

Table 3: BD-Rate results of default QPA algorithm (applied only to 64×64 block sizes) and our proposed algorithm.

Sequence resolution	BD-Rate (%)			
	WPSNR		MS-SSIM	
	QPA 64	Ours	QPA 64	Ours
2160p	-3.698	-4.606	-5.135	-1.659
1080p	0.508	-4.764	-0.942	-7.203
720p	0.674	-6.035	-1.029	-18.715
480p	0.520	-5.087	-1.647	-8.045
240p	1.347	3.140	0.610	-5.592

Our analysis indicates that the proposed model consistently achieves negative BD-Rate values, demonstrating gains in most resolutions, particularly at higher resolutions like 720p and 1080p. For the WPSNR metric, our model outperforms QPA algorithm in all but the 240p resolution, where there is a slight increase in the BD-Rate. This suggests that, based on WPSNR metric, our model effectively improves compression efficiency while maintaining visual quality, but it does not perform well at lowest video resolutions.

When evaluated with the MS-SSIM metric, our model again shows significant improvements over QPA, especially at 720p and 1080p resolutions, indicating better perceptual quality preservation. The only notable exception is at 2160p (4K sequences), where although our model has gains, the QPA algorithm obtains a significantly higher value than our proposal.

Overall, the results affirm the effectiveness of our hybrid neural network model in enhancing the perceptual performance of video compression, demonstrating its potential for practical application in the framework of video coding in the VVC standard, outperforming the default perceptual algorithm.

4 Conclusion

In this study, we proposed a hybrid neural network model combining a convolutional neural network (CNN) and structured data for determining the quantization parameter (QP) value in 64×64 blocks in the Versatile Video Coding (VVC) standard. Our model, integrating a CNN for image processing and an artificial neural network (ANN) for structured data, aimed to enhance compression prediction accuracy. Performance was evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

The results demonstrate that our hybrid model consistently outperforms existing methods, achieving lower error rates across various dataset partitions. Notably, our model achieved significant gains in BD-Rate for most resolutions, particularly at 720p and 1080p, when assessed with both the WPSNR and MS-SSIM metrics. This indicates that our approach not only improves compression efficiency but also maintains or enhances visual quality, especially at higher resolutions.

In future work, we aim to further optimize the hybrid neural network model by exploring advanced hyperparameter tuning techniques and incorporating additional features that impact video compression. Specifically, we plan to work with 32×32 blocks to evaluate if this leads to even greater perceptual gains. Regarding the neural network architecture, we intend to add more structured data to the network, which could help the model converge faster and achieve lower loss.

Acknowledgements

This research was funded by MICIU/AEI/10.13039/501100011033 and by "ERDF A way of making Europe" under grant PID2021-123627OB-C55 and by the Valencian Ministry of Innovation, Universities, Science and Digital Society (Generalitat Valenciana) under grant CIAICO/2021/278.

References

[1] Y. Yan, G. Xiang, Y. Li, X. Xie, W. Yan, and Y. Bao, "Spatiotemporal Perception Aware Quantization Algorithm For Video Coding," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6. DOI: 10.1109/ICME46284.2020.9102882.

[2] G. Xiang, X. Zhang, X. Huang, *et al.*, "Perceptual Quality Consistency Oriented CTU Level Rate Control for HEVC Intra Coding," *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 69–82, 2022. DOI: 10.1109/TBC.2021.3120916.

[3] S. Jin, X. Guan, and Z. Liu, "VVC adaptive QP offset algorithm based on visual perception," in *Third International Conference on Signal Image Processing and Communication (ICSIPC 2023)*, G. Wang and L. Chen, Eds., International Society for Optics and Photonics, vol. 12916, SPIE, 2023, 129161Z. DOI: 10.1117/12.3005138. [Online]. Available: https://doi.org/10.1117/12.3005138.

[4] M. ZHANG, Z. ZHANG, Y. LI, R. CHENG, H. JING, and Z. LIU, "CTU-level Adaptive QP Offset Algorithm for V-PCC Using JND and Spatial Complexity," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. advpub, 2024EAL2021, 2024. DOI: 10.1587/transfun.2024EAL2021.

[5] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, "VTM common test conditions and software reference configurations for SDR video," in *20th Meeting of the Joint Video Experts Team (JVET)*, Doc. JVET-T2010, Oct. 2020.

[6] J. Chen, Y. Ye, and S. H. Kim, "Algorithm description for Versatile Video Coding and Test Model 11 (VTM 11)," in *20th Meeting of the Joint Video Experts Team (JVET)*, Doc: JCTVC-T2002, Oct. 2020.

[7] C. R. Helmrich, S. Bosse, M. Siekmann, H. Schwarz, D. Marpe, and T. Wiegand, "Perceptually Optimized Bit-Allocation and Associated Distortion Measure for Block-Based Image or Video Coding," in *2019 Data Compression Conference (DCC)*, 2019, pp. 172–181. DOI: 10.1109/DCC.2019.00025.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *ArXiv*, vol. abs/1905.11946, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:167217261.

[9] A. Howard, M. Sandler, B. Chen, *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00140.

[10] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," in *Proc. of the ITU-T Video Coding Experts Group - Thirteenth Meeting*, Apr. 2001.

[11] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.