

Introduction to the training process of JVET-Z0091

Version: 2.0

Liqiang Wang*, Xiaozhong Xu*, Shan Liu*, Franck Galpin⁺

*Tencent

{liqiangwang, xiaozhongxu, shanl}@tencent.com

⁺Interdigital

franck.galpin@interdigital.com

1 Overview

A neural network based filter with a single model is proposed in JVET-Z0091 [1], where the test 1.2.2 is deployed on SADL and shows a good trade-off in terms of performance (both subjective and objective), MAC and memory size. Under the RA configuration, 8.67%, 18.63% and 18.97% YUV BD-rate savings are reported. In this document, the training process of JVET-Z0091 is described in brief. More details can be found in the attached scripts or guidance. Different from the conventional training process, an iterative training method is designed to further improve the performance.

2 Training process

The consistency between the inference process and the training process is important for improving the performance. However, compared with the training data, which is generated by the anchor, the quality of B slices in the inference stage is largely improved before it is fed into the inference network, when the neural network models are integrated into the in-loop filter. The main reason is that the quality of reference pictures has been improved by the neural network based filter ahead. Therefore, there is an inconsistency between the inference process and the training process, especially for the B slices in the higher temporal layer. As for I slices, there is no such issue, because its reconstruction can not be affected by other slices.

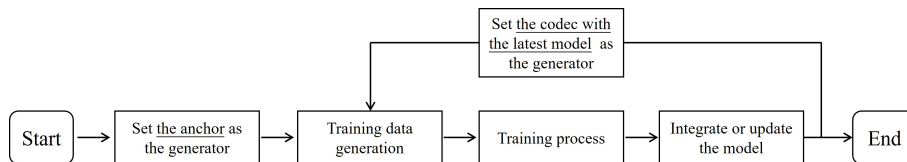


Figure 1: The iterative training method.

One conventional training process is enough for the model used by I slices where an enhanced training process is needed for the model used by B slices. In order to solve the mentioned problem above, an iterative training method is used in the training process, whose flow chart is shown in Figure 1.

2.1 The iterative training method

2.1.1 The initial training stage

In the initial training stage, the training data is generated by the anchor configured with the common test conditions.

2.1.2 The iterative training stage

In the iterative training stage, the training data is generated by the coarse neural network based filter, which is enhanced by the initial training model or the latest training model. Each iterative training stage is fine-tuned based on the parameters of the training model derived by the last training stage. By the iterative training stage, the performance can be improved for its higher consistency.

If individual models are used for I slices and B slices, the consistency can also be improved even if only the model for I slices is integrated. However, it is better to integrate both the model for I slices and the model for B slices when generating the iterative training data.

Theoretically, the more times of the iterative training stages, the better performance. However, only up to two times of training stages, including the initial training stage, are used in the previous proposals. For example, compared with the filter proposed in [2], the filter with 22 models proposed in [3] achieves an additional BD-rate saving of about 2.4%, which is mainly because of the use of iterative training.

Additionally, the training data used for each training stage can also be generated by the codec whose performance is comparable with the codec enhanced by the latest training model. Sometimes, the time for generating the iterative training data can be saved. For example, this training method is utilized by [1]. Actually, the initial training stage can be skipped if the codec with the comparable performance is available.

Using this alternative approach, the performance may not be the best, but the filter can be trained to achieve a higher performance than the conventional neural network based filter quickly. Certainly, the performance can be further improved by the subsequent iterative training stage.

2.2 The application in the training process of JVET-Z0091

All the dataset of DIV2K [4], all the 10-bit dataset of TVD [5] and all the 1920×1088 dataset of BVI-DVC [6] are used for training the model. DIV2K is used for generating the training data of I slices where TVD and BVI-DVC are used for generating the training data of B slices.

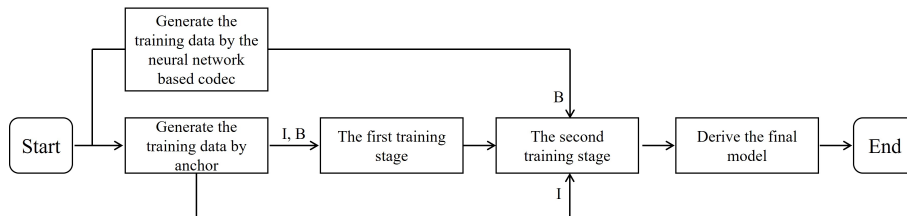


Figure 2: The training process of JVET-Z0091.

As shown in Figure 2, two times of training stages are used for JVET-Z0091. For the first training stage, the training data is generated by the anchor configured with the common test conditions [7, 8] where DBK and SAO are additionally disabled when generating the training data for I slices. For the second training stage, the training data is generated by another neural network based

codec [2], whose performance is estimated to be comparable with the codec enhanced by the model derived by the first training stage. In this way, the dependency of the training data used in the second training stage on the outcome of the first training stage can be avoided, then the training data for the first training stage and the second training stage can be generated in parallel.

The parameters of the training model derived by the first training stage is used to initialize the start model in the second training stage. In the second training stage, there is no need to regenerate the training data of I slices, but the training data of B slices is needed to be regenerated by the neural network based codec to improve the consistency between the inference process and the training process.

Other more detailed information can be found in the training scripts.

3 Implementation

The proposed filter are deployed on SADL [9, 10], and the corresponding codes can be found in the EE1 git repository within the JVET group [11].

References

- [1] L. Wang, X. Xu, S. Liu, “F. Galpin. EE1-1.2: neural network based in-loop filter with a single model,” *Doc. JVET-Z0091*, 26th Meeting, by teleconference, 20–29 April 2022.
- [2] L. Wang, X. Xu, and S. Liu, “AHG11: neural network based in-loop filter with adaptive model selection,” *Doc. JVET-X0054*, 24th Meeting, by teleconference, 6–15 October 2021.
- [3] L. Wang, X. Xu, and S. Liu, “EE1-1.1-related: alternative filter designs,” *Doc. JVET-Y0080*, 25th Meeting, by teleconference, 12–21 January 2022.
- [4] E. Agustsson and R. Timofte, “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.
- [5] X. Xu, S. Liu, and Z. Li, “A Video Dataset for Learningbased Visual Data Compression and Analysis,” *IEEE International Conference on Visual Communications and Image Processing*, 2021.
- [6] D. Ma, F. Zhang, and D. Bull, “BVI-DVC: A Training Database for Deep Video Compression,” *IEEE Transactions on Multimedia*, Sep. 2021, DOI: 10.1109/TMM.2021.3108943.
- [7] https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/VVCSoftware_VTM/-/tree/VTM-11.0.nnvc.
- [8] S. Liu, A. Segall, E. Alshina, and R.-L. Liao, “JVET common test conditions and evaluation procedures for neural network-based video coding technology,” *Doc. JVET-X2016*, 24th Meeting, by teleconference, 6–15 October 2021.
- [9] F. Galpin, T. Dumas, P. Bordes, P. Nikitin, F. Le Léannec, E. François, “AHG11: Small Ad-hoc Deep-Learning Library,” *Doc. JVET-W0181*, 23rd Meeting, by teleconference, 7–16 July 2021.
- [10] <https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/sadl>.
- [11] https://vcgit.hhi.fraunhofer.de/jvet-z-ee1/VVCSoftware_VTM/-/tree/EE1-1.5.