

Article

A VVC Video Steganography Based on Coding Units in Chroma Components with a Deep Learning Network

Minghui Li ^{1,†}, Zhaohong Li ^{1,†} and Zhenzhen Zhang ^{2,*,†}¹ School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China² School of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China

* Correspondence: zhangzhenzhen@bigc.edu.cn

† These authors contributed equally to this work.

Abstract: Versatile Video Coding (VVC) is the latest video coding standard, but currently, most steganographic algorithms are based on High-Efficiency Video Coding (HEVC). The concept of symmetry is often adopted in deep neural networks. With the rapid rise of new multimedia, video steganography shows great research potential. This paper proposes a VVC steganographic algorithm based on Coding Units (CUs). Considering the novel techniques in VVC, the proposed steganography only uses chroma CUs to embed secret information. Based on modifying the partition modes of chroma CUs, we propose four different embedding levels to satisfy the different needs of visual quality, capacity and video bitrate. In order to reduce the bitrate of stego-videos and improve the distortion caused by modifying them, we propose a novel convolutional neural network (CNN) as an additional in-loop filter in the VVC codec to achieve better restoration. Furthermore, the proposed steganography algorithm based on chroma components has an advantage in resisting most of the video steganalysis algorithms, since few VVC steganalysis algorithms have been proposed thus far and most HEVC steganalysis algorithms are based on the luminance component. Experimental results show that the proposed VVC steganography algorithm achieves excellent performance on visual quality, bitrate cost and capacity.

Keywords: video steganography; VVC; CNN; CU partition modes**Citation:** Li, M.; Li, Z.; Zhang, Z.A VVC Video Steganography Based on Coding Units in Chroma Components with a Deep Learning Network. *Symmetry* **2023**, *15*, 116. <https://doi.org/10.3390/sym15010116>

Academic Editor: Zichi Wang

Received: 28 November 2022

Revised: 23 December 2022

Accepted: 26 December 2022

Published: 31 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Steganography is the science of hiding secret information into digital media without arousing the suspicions of users. Compared with the image which is used as the steganography carrier in [1–4], video has more redundancies and unique coding characteristics for hiding, and it is spreading more and more widely across social networks and social applications.

Common video steganographic algorithms include modifying the transform domain, motion vectors, inter-prediction modes, intra-prediction modes and block partitioning types. For transform domain-based algorithms, Chang et al. [5] first proposed a data-hiding algorithm based on modifying the Discrete Cosine Transform (DCT) coefficients. For the motion vectors-based algorithms, Rana et al. [6] proposed a steganographic algorithm to embed motion vectors in the homogeneous regions of the reference frame. For algorithms based on inter-prediction modes, Yang et al. [7] and Li et al. [8] embedded messages by modifying the prediction unit (PU) partition modes. Zhang et al. [9] proposed an algorithm based on the intra-prediction mode (IPM). For algorithms based on block partitioning types, Tew et al. [10] proposed an information-hiding algorithm by modifying the coding block size decision. Shanableh et al. [11] altered the coding units to hide secret information.

Most of these steganographic algorithms are used in the HEVC coding standard. However, the latest international video coding standard is VVC. There are lots of novel technical aspects used in the VVC standard that provide more possibilities for steganography. Compared with the HEVC standard, the block partitioning structure of VVC is one

of the most essential changes among these new techniques. In VVC, the coding tree unit (CTU) is extended to a 128×128 size for more flexible block partitioning [12]. VVC uses both quaternary tree (QT)-based partitioning and multi-type tree (MTT)-based partitioning structures [13]. Furthermore, VVC introduces the chroma separate tree (CST) [14]. In the intra-coded slice, the CST enables separate partitioning for luma and chroma. Overall, there are many differences between the HEVC and VVC; hence, the HEVC steganographic algorithms are difficult to use in the VVC standard. In addition, as far as we know, few VVC steganalytic algorithms have been reported in the literature. Therefore, VVC steganography is harder to detect. Thus, in this paper, steganographic algorithm based on a chroma block partitioning structure for VVC videos is proposed.

However, stego-videos always face some disadvantages such as the increased bitrate and the decreased visual quality. Recently, with the development of deep learning techniques, many researchers utilize CNNs instead of the in-loop filters to obtain better visual quality. Huang et al. [15] and Chen et al. [16] proposed a variable convolutional neural network and a dense residual convolutional neural network as an additional in-loop filter for the VVC standard. Inspired by the above literature, we propose a novel multi-scale residual neural network (MSRNN) as an additional in-loop filter in the VVC standard to improve its disadvantages such as distortion in visual quality and increased bitrate caused by a steganographic algorithm.

The contributions of this paper are as follows:

(1) An VVC steganographic algorithm based on chroma block partitioning is proposed, which takes full advantage of the VVC block partitioning structure's characteristics. In this algorithm, secret information is embedded by modifying the chroma component's block partitioning structure in the VVC standard.

(2) A four-embedding-level algorithm is proposed that can satisfy the different needs of the visual quality, bitrate cost and capacity.

(3) MSRNN is proposed as an additional in-loop filter in the VVC standard to decrease the negative influence caused by steganographic algorithms.

The experiment results illustrate that the proposed steganographic algorithm performs well in terms of the visual quality, bitrate cost and capacity. As for the security, we use an universal steganalytic algorithm and an open-source steganalysis tool to test our steganographic algorithm.

The rest of this paper is organized as follows. Section 2 introduces the block partitioning structure in VVC. Section 3 presents the proposed steganographic algorithm and MSRNN algorithm. The experimental results and analyses are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Block Partitioning Structure

2.1. Quadtree Plus Multi-Type Tree Structure

As in HEVC, a picture to be encoded is partitioned into non-overlapping CTUs in VVC. For the purpose of improving coding efficiency, the CTU size is enlarged from 64×64 in HEVC to 128×128 in VVC. Furthermore, HEVC only applies a recursive quaternary tree (QT) split to each CTU. In order to adapt to the picture content better, the VVC block structure adopts a QT and a multi-type tree (MTT). The multi-type tree structure includes split vertical binary trees (VBT), split horizontal binary trees (HBT), split vertical ternary trees (VTT) and split horizontal ternary trees (HTT). For binary tree splitting, the splitting is equal. For ternary tree splitting, the splitting ratio is 1:2:1.

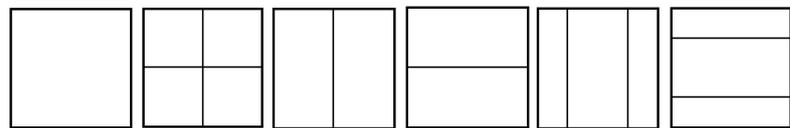
Figure 1 illustrates the splitting types of the MTT. In VVC, each CTU is partitioned by a QT at first. Then, the MTT structure is applied to partition each QT node further. Once the current node is partitioned by the MTT, the QT structure is forbidden for the subsequent nodes. Figure 2 shows two redundant partitions in the block partition process. The final partition mode of a CU is decided by minimizing the Rate Distortion cost (RD cost) [17] among all the possible partition modes. The VVC block partition process is shown in Algorithm 1. Figure 3

shows an example of a CTU partition in VVC. If the block partitioning structure is altered, it will result in the distortion of visual quality and compression efficiency.

Algorithm 1: Partition process.

```

modelist = QT, VBT, VTT, HBT, HTT, ...;
Calculate the current RD cost for current CU;
depth = depthQT + depthMTT;
if CU can be further split into sub-CUs then
    if depthQT = 0 then
        mode = QT;
        depthQT = depthQT + 1;
    else
        for mode in modelist do
            if mode is allowed then
                | Add and save the cost for sub-CUs
            else
                | Continue
            end
            Compare the cost of current CU and sub-CUs;
            if Splitting then
                if mode = QT then
                    | depthQT = depthQT + 1
                else
                    | depthMTT = depthMTT + 1;
                    | Remove QT in modelist
                end
            end
        end
    end
end
    
```



(a) Non-split (b) QT split (c) VBT split (d) HBT split (e) VTT split (f) HTT split

Figure 1. Illustration of splitting types in MTT.

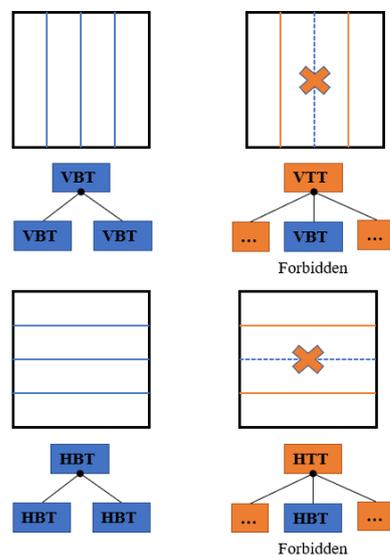


Figure 2. Illustration of redundant partitions.

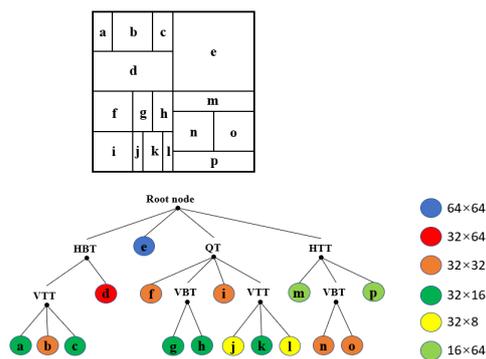


Figure 3. Example of a CTU partition.

2.2. Chroma Separate Tree

In the HEVC standard, the coding tree is shared by the luma component and the chroma components. As a result, a CU includes a luma coding block (CB) and two chroma CBs. This single-tree structure is still used for P and B slices in the VVC standard. However, VVC introduces the chroma separate tree (CST), which enables the luma component and chroma components to be encoded separately in I slices. Figure 4 shows an example of CU partitioning of an encoded picture. The partition structure is marked by the open-source player YUView [18]. As shown in Figure 4, luma has a finer texture than chroma, which causes the amount of small-sized CUs in luma to be larger than that in chroma. The CST enables chroma to not be split into smaller CUs. Moreover, if the CST is applied, there is no dependency between the luma component and the chroma components, but the processing latency still exists.

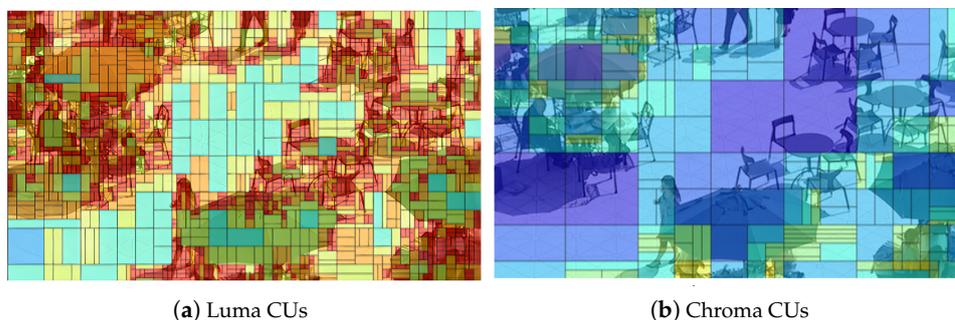


Figure 4. Example of CU partitions.

It can be concluded that whether we modify the block partitioning structure of luma or chroma components, the degree of influence on visual quality and compression efficiency is similar. However, in the human visual system, the luma component is more sensitive than the chroma components [19], and generally, the chroma components are subsampled to reduce redundancy [20]. Consequently, we choose to only modify the block structure of the chroma components for embedding secret bits, which can effectively reduce the impact on visual quality and compression efficiency.

3. The Proposed Algorithm

3.1. The Chroma CU MTT Depth-Based Hierarchical Coding

The proposed hierarchical coding method is based on the MTT depth of the chroma CUs. This method includes two sub-bijective mapping rules that convert secret binary bits to particular block partition modes. For simplicity, a chroma CU with an MTT depth j is expressed as $CU_{depth_{MTT}=j}$.

The first bijective mapping rule is called the 4-bits mapping rule which can embed 4 secret binary bits to a 16×16 $CU_{depth_{MTT}=0}$, which also means its $depth_{QT}$ is 2 for the YUV420 format.

Step I: In VVC standard, a 16×16 $CU_{depth_{MTT}=0}$ can be split by 5 partition modes. However, considering the complexity, the 4-bits mapping rule removes the QT split. Hence, there are only 4 partition modes left to be chosen. According to Table 1, the partition modes of a 16×16 $CU_{depth_{MTT}=0}$ can be mapped to 2 secret binary bits.

Table 1. Mapping of 16×16 $CU_{depth_{MTT}=0}$ partition modes.

$depth_{MTT} = 0$ modes	VBT	VTT	HBT	HTT
Binary bits	00	01	10	11

Step II: We choose the first 8×16 or 16×8 $CU_{depth_{MTT}=1}$ in sub-CUs to embed secret information. In order to avoid redundant partitions, in $depth_{MTT} = 1$, we only choose 2 partition modes HBT and VBT. In our design, if CU is split by VBT or VTT in **Step I**, $CU_{depth_{MTT}=1}$ is only split by HBT. For the CU which is split by HBT or HTT in **Step I**, $CU_{depth_{MTT}=1}$ is only split by VBT. The mapping of the $CU_{depth_{MTT}=1}$ partition modes is shown in Table 2.

Table 2. Mapping of $CU_{depth_{MTT}=1}$ partition modes.

$depth_{MTT} = 0$ modes	VBT	VTT	HBT	HTT
$depth_{MTT} = 1$ modes	HBT	HBT	VBT	VBT

Step III: A $CU_{depth_{MTT}=1}$ can be parted into 2 sub-CUs, as illustrated in Figure 5, and we can embed 1 bit at $depth_{MTT} = 2$. It can be concluded that if the $CU_{depth_{MTT}=3}$ is located in the first $CU_{depth_{MTT}=2}$, the secret bit is 0, and if the $CU_{depth_{MTT}=3}$ is located in the second $CU_{depth_{MTT}=2}$, the secret bit is 1. The mapping rule is defined as

$$M = \begin{cases} 0, & i = 0 \\ 1, & i = 1' \end{cases} \tag{1}$$

where M denotes the binary coding for sub-CU with order i .

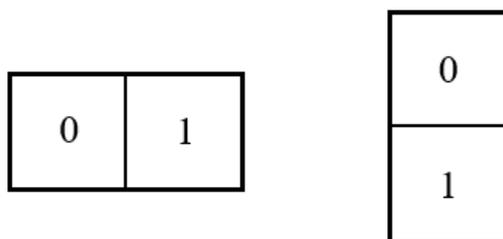


Figure 5. Illustration of sub-CUs of $CU_{depth_{MTT}=1}$.

Step IV: For a $CU_{depth_{MTT}=2}$, there are only 2 partition modes left, which can embed 1 bit. The mapping rule of $CU_{depth_{MTT}=3}$ partition modes is shown in Table 3. Algorithm 2 shows the process of the 4-bits mapping rule.

Table 3. Mapping of $CU_{depth_{MTT}=3}$ partition modes.

$depth_{MTT} = 3$ modes	VBT	HBT
Binary bits	0	1

Algorithm 2: 4-bits Mapping Rule.

```

depth = depthQT + depthMTT;
if Chroma CU's size = 16 × 16 and depthMTT = 0 then
    Step = Step I;
    Embedding information according to the Table 1;
    depthMTT = depthMTT + 1
else
    partition mode = QT
end
if depthMTT = 1 then
    Step = Step II;
    Embedding information according to the Table 2;
    depthMTT = depthMTT + 1
end
if depthMTT = 2 and Step = Step III then
    Step = Step IV;
    Embedding information according to the Table 3;
end
if depthMTT = 2 then
    Step = Step III;
    Embedding information according to the Equation (1);
end
    
```

The second bijective mapping rule is called the 2-bits mapping rule that can embed 2 secret binary bits to a 16 × 16 chroma CU, for which the $depth_{MTT}$ is 0. The difference between the 4-bits mapping rule and the 2-bits mapping rule is that **Step II**, **Step III** and **Step IV** are not forcible in the 2-bits rule. The block partitioning of the $CU_{depth_{MTT}=1}$ is dependent on the RD cost.

Figure 6 illustrates an example of the proposed hierarchical coding method. The number of corresponding bits is 1010.

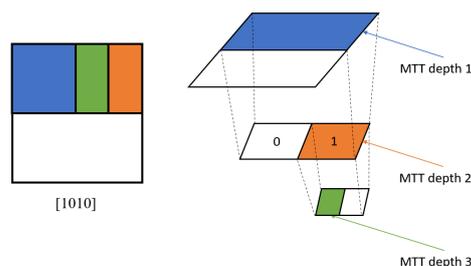


Figure 6. Illustration of the proposed hierarchical coding method.

By using the proposed bijective mapping, we can convert binary secret messages to CUs of different sizes. Therefore, there are 2 methods to obtain the $16 \times 16 CU_{depth_{MTT}=0}$. In the next part, the 2 methods used for the proposed hierarchical coding method are introduced.

3.2. Four Embedding Schemes

As shown in Figure 4, there are some chroma CUs for which not all the $depth_{QT}$ of chroma CUs equals 2. If we apply the proposed algorithm to the whole picture, the visual quality will be decreased. Therefore, we propose two methods to use the proposed hierarchical coding method. Method 1 is to forcibly modify all the $CU_{depth_{QT}=2}$ and then utilize mapping rules to embed secret information.

As for Method 2, we only select the $CU_{depth_{QT}=2}$ to embed secret information. Therefore, we first start the block partitioning of the chroma components to find $CU_{depth_{QT}=2}$.

Then, we start the block partitioning of the chroma components again, and this time we apply the proposed hierarchical coding method to the $CUs_{depth_{QT}=2}$ that we found the first time. As for the other chroma CUs, we utilize the structure from the first time as the final structure. Additionally, as shown in Figure 7, if the block partitioning structure has been modified, it will influence the following block partitioning.

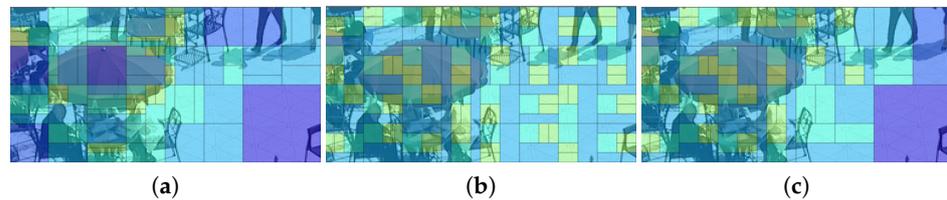


Figure 7. (a) The original frame; (b,c) data-hiding frames at Level 1 and Level 2, respectively.

Thus, there are four different embedding schemes which are shown in Table 4. To extract embedded information, we just need to calculate the corresponding coding bits of each QT depth 2 chroma CU in zigzag order.

Table 4. Four Embedding-Level Schemes.

Schemes	Mapping Rules	Embedding Method
Level 1	4-bits	Method 1
Level 2	4-bits	Method 2
Level 3	2-bits	Method 1
Level 4	2-bits	Method 2

3.3. The Additional In-Loop Filter MSRNN

The proposed steganographic algorithm will affect the visual quality and bitrate of the embedded video sequences. In order to improve the performance of the embedded video sequences, we utilize MSRNN as an additional in-loop filter. Figure 8 shows the steganography algorithm diagram. Firstly, the raw video sequence is compressed by a VVC AI encoder. In the process of encoding, the CUs partition modes are extracted, and then the selected chroma CUs are modified according to the secret data. The subsequent VVC encoding process is continued, where we utilize MSRNN as an additional in-loop filter. As shown in Figure 9, the MSRNN is located after the deblocking filter (DBF). In [15,16], the proposed CNN-based in-loop filter modules can improve the visual quality and bitrate effectively. The MSRNN structure is shown in Figure 10 and the details of each convolution kernel is shown in Table 5.

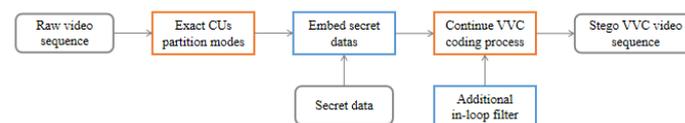


Figure 8. The proposed steganography algorithm diagram.



Figure 9. Intergration into the VVC diagram.

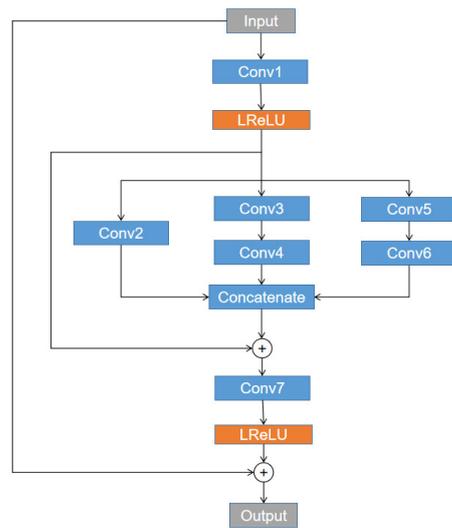


Figure 10. The architecture of MSRNN.

Table 5. The configuration of the MSRNN.

Layer	Filter Size	Filter Number
Conv1	3×3	64
Conv2	3×3	32
Conv3	3×3	16
Conv4	3×3	16
Conv5	5×5	16
Conv6	5×5	16
Conv7	3×3	1

Therefore, we proposed a super-resolution CNN called MSRNN as an additional in-loop filter module. The MSRNN structure is shown in Figure 10 and the details of each convolution kernel is shown in Table 5. In order to extract multiscale features, the convolutional layers we utilized are with different kernel size. Leaky ReLU (LReLU) activation function is aimed to get the shallow features (SFs) of the input. We also utilize zero-padding to make the size of output as same as the input and stride is set to 1. DIV2K dataset [21] is used for training and VTM16.2 AI encoder is used to compress original frames at QPs: 26, 32, 38 and 42. The network is trained individually for each QP. We also utilize luma component and chroma components datasets to train out the network separately. The original image I_{ori} is the target of the CNN. The input I_{input} is the compressed I_{ori} . The loss function we utilized is:

$$Loss = \frac{1}{N} \cdot \sum_{n=0}^N \cdot \|I_{ori} - I_{input}\|, \quad (2)$$

where N is the number of training images, I_{ori} means the original picture and I_{output} is the output of CNN. In order to test the effect of the MSRNN, we compared with improved VRCNN [22] on BD-rate. Under the same video quality, if the BD-rate is smaller, the bitrate savings is more. Table 6 shows the comparison results. From Table VI, the BD-rate of each algorithm is negative, therefore, we can know that using CNN as an additional in-loop filter can improve the reconstructed video quality effectively and the MSRNN is more effective as an additional in-loop filter.

Table 6. The Configuration of the CNN.

Class	Improved VRCNN [22]	Proposed
ClassA	−1.73%	−1.73%
ClassB	−0.63%	−0.72%
ClassC	−3.18%	−3.38%
ClassD	−3.60%	−3.76%
ClassE	−2.54%	−2.77%

4. Experimental Results

4.1. Setup

The proposed steganographic algorithm and the MSRNN are intergrated in the VVC reference software VTM16.2 AI encoder and tested on a database that includes 18 YUV sequences, which is shown in Table 7 in detail. In our experiment, the test sequences are encoded at a frame rate 30 fps with QPs 26, 32 and 38 and the temporal subsample ratio is 8, which means the sequences are encoded at intervals of 8 frames. Additionally, the final results are normalized with the encoded frame numbers. We utilize the DIV2K dataset [21] to train the MSRNN. The I_{ori} and I_{input} are cropped to 128×128 . The method proposed in [23] is utilized to initialize the weights, and the Adam optimizer [24] is also utilized for training.

4.2. Subjective Performance

The basic requirement of steganography is that human eyes cannot distinguish whether the videos are embedded with secret information. Figure 11 shows the original VVC compressed video and stego-video under four different hiding strategies and with MSRNN. As shown in Figure 11, stego-videos with the MSRNN produce better visual quality, especially for the grass, and it is difficult for human eyes to distinguish whether these videos are embedded with secret information. This observation verifies that the proposed steganography algorithm preserves visual quality well.

Table 7. The Video Database.

Class	Sequence Name	Resolution	Encoded Frames Numbers
ClassA	PeopleOnStreet Traffic	2560×1600	18
ClassB	BasketballDrive BQTerrace Cactus Kimono1 ParkScene	1920×1080	20
ClassC	BasketballDrill BQMall PartyScene RaceHorses	832×480	20
ClassD	BasketballPass BlowingBubbles BQSquare RaceHorses	416×240	20
ClassE	FourPeople Johnny KristenAndSara	1280×720	20

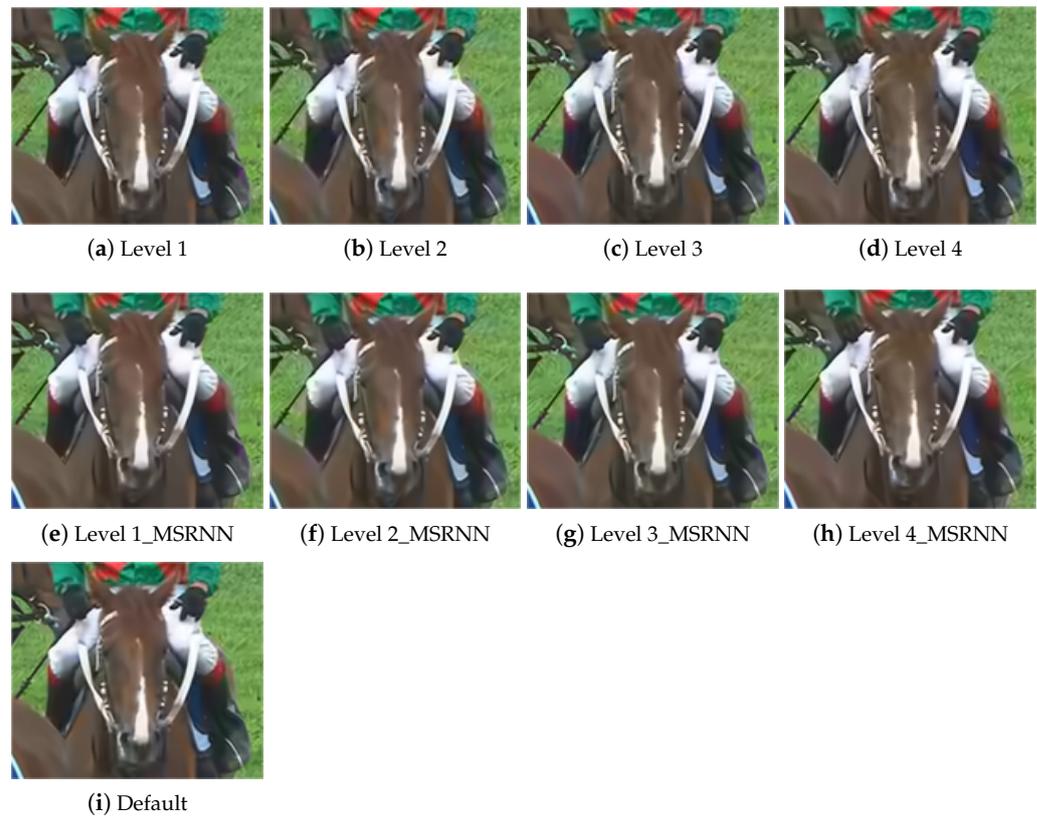


Figure 11. Visual quality of I frame in RaceHorses of ClassD.

4.3. Objective Performance

We utilize the following four evaluation methods: peak signal-to-noise ratio (PSNR), bit rate increasing (BRI), embedding capacity and anti-steganalysis to measure the performance of the proposed algorithm objectively.

The PSNR is used as a classical index to evaluate the objective quality of images. The PSNR between the 8-bit-depth original image I and 8-bit-depth reconstructed image I' can be calculated by (3) and (4), respectively:

$$MSE = \frac{1}{W \cdot H} \cdot \sum_{x=0}^{W-1} \cdot \sum_{y=0}^{H-1} \cdot (I(x, y) - I'(x, y))^2, \quad (3)$$

$$PSNR = -10 \cdot \log_{10} \left(\frac{MSE}{255^2} \right), \quad (4)$$

where W and H represent the width and height of the image, respectively. To measure the quality of YUV420 format videos, the PSNR is given by:

$$PSNR_{YUV} = \frac{6 \cdot PSNR_Y + PSNR_U + PSNR_V}{8}, \quad (5)$$

where $PSNR_Y$, $PSNR_U$ and $PSNR_V$ denote the average PSNR values of the Y component, U component and V component, respectively.

The video bitrate represents the number of transmitted bits per second. BRI represents the increase in the bitrate between the modified video and the original video and is defined as

$$BRI = \frac{BR_{steg} - BR_{ori}}{BR_{ori}}, \quad (6)$$

where BR_{steg} and BR_{ori} denote the bitrate of the modified video and the bitrate of the original video, respectively.

The embedding capacity is the number of embedded binary bits, and in our experiment, it is the average embedding capacity of each I slice.

Table 8 shows the results of the PSNR of different channels, the BRI and the capacity using different QPs. The results shown that for most test videos, the PSNR is decreased around 0.27 dB and the average BRI is 3.07%, which indicates that the proposed steganographic algorithm has just a little negative influence on visual quality and bitrate. The smaller QP represents the smaller quantization step, and during the rounding and truncation process, less information is lost. In addition, the lower capacity means that the distortion caused by the modification is smaller. Thus, the PSNR, BRI and capacity are decreased with the same trend, that is, with the increase in QP.

Table 8. The PSNR, BRI and capacity performance of different QPs.

Class	QP	$PSNR_Y$ (Level 2_MSRNN)	$PSNR_U$ (Level 2_MSRNN)	$PSNR_V$ (Level 2_MSRNN)	$PSNR_{YUV}$ (Level 2_MSRNN)	$PSNR_{YUV}$ (Default)	BRI (Level 2_MSRNN)	Capacity (Level 2_MSRNN)
ClassA	26	41.7852	41.2967	42.4510	41.7297	42.1655	2.68%	8768.00
	32	38.1119	39.5137	40.8508	38.6314	38.7845	1.24%	3068.00
	38	34.6532	37.7330	39.1266	35.5459	35.5106	0.74%	1096.00
ClassB	26	40.1954	40.8220	42.7717	40.6005	40.9229	2.27%	3540.80
	32	37.5342	39.2428	40.9064	38.1727	38.3461	1.72%	1595.20
	38	34.6550	37.9042	38.9339	35.5435	35.5857	1.36%	774.40
ClassC	26	39.8219	38.8454	39.6275	39.6237	40.2379	5.48%	1294.00
	32	35.7351	35.8652	36.5972	35.8856	36.3046	4.30%	1000.00
	38	31.7524	34.0027	34.5614	32.4303	32.7169	4.04%	640.00
ClassD	26	39.9070	38.7649	39.4814	39.5875	40.2220	6.55%	322.00
	32	35.3895	35.8083	36.1828	35.4840	35.9306	4.55%	218.00
	38	30.3488	33.9349	34.2280	31.9570	32.2150	4.04%	170.00
ClassE	26	43.3082	45.2538	45.9940	43.9427	44.1721	3.72%	1029.33
	32	40.5442	42.6368	43.3097	41.2098	41.3639	2.33%	528.00
	38	36.3138	40.8980	41.6887	38.2767	38.2435	1.06%	213.33
Mean				37.9081	38.1815	3.07%	1617.14	

Because VVC is the latest video coding standard, there are few steganographic algorithms for comparison. Therefore, we just compared the results of our algorithm when using MSRNN as an additional in-loop filter and four different schemes. Table 9 shows the comparative result in QP 26. The results show that the application of MSRNN performs well in improving the PSNR. Especially for $PSNR_U$ and $PSNR_V$, the MSRNN plays an important role in recovering the negative influence caused by the modification. As expected, the $PSNR_Y$ is almost not influenced by the steganography. Additionally, the proposed algorithm also performs well on the BRI and embedding capacity. Level 1 and Level 3, which utilize Method 1, have a better performance on embedding capacity at the expense of the PSNR and BRI . On the contrary, Level 2 and Level 4 perform better on the PSNR and BRI at the expense of a decrease in embedding capacity. Similarly, with the same embedding method, the schemes with the four-bits mapping rule (Level 1 and Level 2) normally have a better performance in capacity. In summary, according to the different needs, we can choose different schemes to embed secret information.

Table 9. The PSNR, BRI and capacity performance of QP = 26.

Class	Schemes	$PSNR_Y$	$PSNR_U$	$PSNR_V$	$PSNR_{YUV}$	BRI	Capacity
ClassA	Default	41.6335	42.9328	44.0051	42.3597		
	Level 1	41.7114	41.1107	42.3614	41.6399	3.53%	16,000.00
	Level 2	41.7161	41.1457	42.3382	41.6434	2.62%	8768.00
	Level 3	41.7120	41.2661	42.5203	41.6900	2.36%	9576.00
	Level 4	41.7030	41.3112	42.5623	41.6995	1.76%	5160.00
	Level 1_MSRNN	41.7797	41.2729	42.4788	41.7284	3.55%	16,000.00
	Level 2_MSRNN	41.7852	41.2967	42.4510	41.7297	2.68%	8768.00
	Level 3_MSRNN	41.7815	41.4306	42.6278	41.7783	2.39%	9576.00
	Level 4_MSRNN	41.7725	41.4630	42.6658	41.7844	1.80%	5160.00
ClassB	Default	39.8567	42.0612	44.0721	40.6242		
	Level 1	40.1853	40.6023	42.4676	40.5311	3.37%	7920.00
	Level 2	40.1845	40.7686	42.6871	40.5781	2.24%	3540.80
	Level 3	40.1832	40.6950	42.6181	40.5605	2.15%	4624.40
	Level 4	40.1834	40.8523	42.7661	40.5983	1.81%	2018.40
	Level 1_MSRNN	40.1950	40.6290	42.1957	40.4964	3.36%	7920.00
	Level 2_MSRNN	40.1954	40.8220	42.7717	40.6005	2.26%	3540.80
	Level 3_MSRNN	40.1950	40.7774	42.7470	40.5926	2.20%	4624.40
	Level 4_MSRNN	40.1949	40.9158	42.8657	40.6238	1.87%	2018.40
ClassC	Default	39.5904	41.2428	42.1792	40.1519		
	Level 1	39.7120	38.8564	39.7711	39.5500	5.57%	1560.00
	Level 2	39.6998	38.6593	39.5338	39.4646	5.56%	1294.00
	Level 3	39.7155	38.9484	39.8661	39.5830	5.01%	1099.50
	Level 4	39.7108	38.9120	39.8781	39.5727	5.44%	800.00
	Level 1_MSRNN	39.8250	39.0794	40.0634	39.7172	5.42%	1560.00
	Level 2_MSRNN	39.8219	38.8454	39.8077	39.6237	5.48%	1294.00
	Level 3_MSRNN	39.9359	39.1714	40.1578	39.7479	4.90%	1099.50
	Level 4_MSRNN	39.8343	39.1326	40.1796	39.7413	5.38%	800.00
ClassD	Default	39.5789	41.4094	42.1426	40.1426		
	Level 1	39.6644	38.8616	39.4872	39.4482	5.98%	364.00
	Level 2	39.6834	38.6192	39.1996	39.3675	6.76%	322.00
	Level 3	39.6768	38.9773	39.6241	39.5068	5.63%	260.00
	Level 4	39.6627	39.0237	39.6032	39.4957	6.37%	187.00
	Level 1_MSRNN	39.8883	39.0062	39.7214	39.6604	5.62%	364.00
	Level 2_MSRNN	39.9070	38.7649	39.4814	39.5875	6.55%	322.00
	Level 3_MSRNN	39.9036	39.0993	39.8522	39.7160	5.25%	260.00
	Level 4_MSRNN	39.8926	39.1727	39.8666	39.7185	6.05%	187.00
ClassE	Default	43.1490	46.6232	47.5673	44.1049		
	Level 1	43.2314	44.5351	45.4497	43.7308	7.49%	3520.00
	Level 2	43.2241	45.0728	45.7937	43.8349	3.72%	1029.33
	Level 3	43.2204	44.7645	45.6570	43.7776	5.16%	1945.33
	Level 4	43.2099	45.2183	46.0224	43.8641	3.15%	582.67
	Level 1_MSRNN	43.3161	44.7835	45.6571	43.8513	7.40%	3520.00
	Level 2_MSRNN	43.3082	44.2538	45.9940	43.9427	3.72%	1029.33
	Level 3_MSRNN	43.3091	44.9987	45.8640	43.8974	5.02%	1945.33
	Level 4_MSRNN	43.2946	45.3951	45.1864	43.9677	3.10%	582.67

4.4. Comparative Analysis

In this section, we compare our proposed algorithm with Shanableh [11], which is an HEVC steganography algorithm based on CU block partition. In order to display the results more intuitively, we utilize $\Delta PSNR$ to measure the change in $PSNR_{YUV}$ of the default VVC ($PSNR'_{YUV}$) compared to the proposed steganography algorithm ($PSNR''_{YUV}$):

$$\Delta PSNR = PSNR''_{YUV} - PSNR'_{YUV}. \quad (7)$$

We utilized three test sequence (BasketballPass 416×240 , BasketballDrill 832×480 and FourPeople 1280×720).

Table 10 shows the comparison results for $\Delta PSNR$ and capacity. With the increase in QP, the capacity is reduced. Furthermore, with the increase in resolution, the capacity also increases. The reason for this is that in higher-resolution videos, there will be more suitable CUs in which to embed secret information.

The values in Table 10 marked in bold indicate the best performance. As shown in the comparison results, the proposed steganography has great advantage in visual quality and capacity.

Table 10. Comparison results for Δ PSNR and capacity.

Method	QP	BasketballPass		BasketBallDrill		FourPeople	
		Δ PSNR	Capacity	Δ PSNR	Capacity	Δ PSNR	Capacity
Shanableh's [11]	26	-0.25	138.20	-0.22	548.20	-0.08	877.17
	32	-0.33	133.60	-0.43	454.20	-0.30	557.17
	38	-0.86	116.80	-0.65	404.60	-0.59	477.83
Level2	26	-1.00	284.00	-0.61	1288.00	-0.36	1184.00
	32	-0.80	196.00	-0.43	968.00	-0.36	544.00
	38	-0.70	156.00	-0.64	736.00	-0.14	208.00
Level2 _MSRNN	26	-0.74	284.00	-0.38	1288.00	-0.22	1184.00
	32	-0.58	196.00	-0.22	968.00	-0.13	544.00
	38	-0.50	156.00	-0.48	736.00	0.15	208.00

4.5. Security Performance

The security performance is also an important evaluation criterion for a steganographic algorithm. Nevertheless, few steganalytic approaches have been proposed for a VVC steganographic algorithm.

Thus, we utilize StegExpose [25] and the latest universal steganalytic algorithm [26] to evaluate the security of the proposed steganographic algorithm. StegExpose [25] is an open-source steganalysis tool. Figure 12 shows the ROC curve of the proposed method by setting thresholds in a wide range, indicating that the ROC curve of the proposed method is very close to the curve of random guesses. Because the input of [26] only includes grayscale information, the detection accuracy is only 49.81%. These results show that our steganographic algorithm is hard for steganalysis algorithms based on luminance components to detect. Almost all the steganalytic approaches for detecting stego-video with previous encoding standards, such as MPEG4, H.264 and HEVC, only utilize the statistical data in the luma component, which makes our algorithm have an advantage in terms of security performance. Different from previous standards, VVC is the only standard that has separate CU block structure rules for the chroma component, and this unique feature is used in the proposed algorithm, which has guaranteed both high visual quality and the security, as shown in the experimental results.

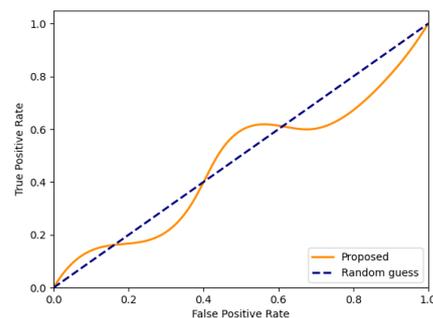


Figure 12. The ROC curve produced by StegExpose [25].

5. Conclusions

In this paper, we proposed a novel VVC steganographic algorithm based on chroma CUs and an additional in-loop filter based on CNN. Different from HEVC steganography, the VVC standard designs a new VVC technique, CST. Benefiting from this new technique, we only utilize chroma CUs to embed secret messages. To improve the distortion and reduce the video bitrate, a deep learning network called MSRNN is designed to replace the in-loop filter in the VVC codec. Our experimental results verify the efficiency of MSRNN

and show that the proposed algorithm has high embedding efficiency and strong security. In the future, we hope to widen our research on VVC steganography and utilize the characteristics of inter-frames to develop novel steganography schemes.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by The Scientific Research Common Program of the Beijing Municipal Commission of Education (KM202110015004).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, Y.; Liao, X. Improved CMD Adaptive Image Steganography Method. In *International Conference on Cloud Computing and Security*; Springer: Cham, Switzerland, 2017; pp. 74–84.
2. Al-Shatnawi, A.M. A new method in image steganography with improved image quality. *Appl. Math. Sci.* **2012**, *6*, 3907–3915.
3. Asad, M.; Gilani, J.; Khalid, A. An enhanced least significant bit modification technique for audio steganography. In Proceedings of the International Conference on Computer Networks and Information Technology, Abbottabad, Pakistan, 11–13 July 2011; pp. 143–147.
4. Mandal, K.K.; Jana, A.; Agarwal, V. A new approach of text Steganography based on mathematical model of number system. In Proceedings of the 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, India, 20–21 March 2014; pp. 1737–1741.
5. Chang, P.C.; Chung, K.L.; Chen, J.J.; Lin, C.H.; Lin, T.J. A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames. *J. Vis. Commun. Image Represent.* **2014**, *25*, 239–253. [[CrossRef](#)]
6. Rana, S.; Kamra, R.; Sur, A. Motion vector based video steganography using homogeneous block selection. *Multimed. Tools Appl.* **2020**, *79*, 1–16. [[CrossRef](#)]
7. Yang, Y.; Li, Z.; Xie, W.; Zhang, Z. High capacity and multilevel information hiding algorithm based on pu partition modes for HEVC videos. *Multimed. Tools Appl.* **2019**, *78*, 8423–8446. [[CrossRef](#)]
8. Li, Z.; Meng, L.; Jiang, X.; Li, Z. High Capacity HEVC Video Hiding Algorithm Based on EMD Coded PU Partition Modes. *Symmetry* **2019**, *11*, 1015. [[CrossRef](#)]
9. Wang, J.; Jia, X.; Kang, X.; Shi, Y.Q. A Cover Selection HEVC Video Steganography Based on Intra Prediction Mode. *IEEE Access* **2019**, *7*, 119393–119402. [[CrossRef](#)]
10. Tew, Y.; Wong, K. Information hiding in HEVC standard using adaptive coding block size decision. In Proceedings of the 2014 IEEE international conference on image processing (ICIP), Paris, France, 27–30 October 2014; pp. 5502–5506.
11. Shanableh, T. Data embedding in high efficiency video coding (HEVC) videos by modifying the partitioning of coding units. *Image Process. IET* **2019**, *13*, 1909–1913. [[CrossRef](#)]
12. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.R. Overview of the Versatile Video Coding VVC Standard and its Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [[CrossRef](#)]
13. Zhang, Q.; Wang, Y.; Huang, L.; Jiang, B. Fast CU Partition and Intra Mode Decision Method for H.266/VVC. *IEEE Access* **2020**, *8*, 117539–117550. [[CrossRef](#)]
14. Huang, Y.W.; An, J.; Huang, H.; Li, X.; Hsiang, S.T.; Zhang, K.; Gao, H.; Ma, J.; Chubach, O. Block Partitioning Structure in the VVC Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3818–3833. [[CrossRef](#)]
15. Huang, Z.; Sun, J.; Guo, X.; Shang, M. One-for-All: An Efficient Variable Convolution Neural Network for In-Loop Filter of VVC. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2342–2355. [[CrossRef](#)]
16. Chen, S.; Chen, Z.; Wang, Y.; Liu, S. In-Loop Filter with Dense Residual Convolutional Neural Network for VVC. In Proceedings of the 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Shenzhen, China, 6–8 August 2020; pp. 149–152. [[CrossRef](#)]
17. Sullivan, G.J.; Wiegand, T. Rate-distortion optimization for video compression. *IEEE Signal Process. Mag.* **1998**, *15*, 74–90. [[CrossRef](#)]
18. IENT. YUView. 2021. Available online: <https://github.com/IENT/YUView> (accessed on 26 October 2021).
19. Starosolski, R. New simple and efficient color space transformations for lossless image compression. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1056–1063. [[CrossRef](#)]
20. Chung, K.L.; Huang, C.C.; Hsu, T.C. Adaptive chroma subsampling-binding and luma-guided chroma reconstruction method for screen content images. *IEEE Trans. Image Process.* **2017**, *26*, 6034–6045. [[CrossRef](#)] [[PubMed](#)]
21. Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.H.; Zhang, L. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017.

22. Liu, J.; Li, Z.; Jiang, X.; Zhang, Z. A High-Performance CNN-Applied HEVC Steganography Based on Diamond-Coded PU Partition Modes. *IEEE Trans. Multimed.* **2022**, *24*, 2084–2097. [[CrossRef](#)]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
24. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
25. Boehm, B. StegExpose—A Tool for Detecting LSB Steganography. *arXiv* **2014**, arXiv:1410.6656.
26. Liu, P.; Li, S. Steganalysis of Intra Prediction Mode and Motion Vector-based Steganography by Noise Residual Convolutional Neural Network. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *719*, 012068. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.