sensors

MDPI

Article

Video Compressive Sensing Reconstruction Using Unfolded LSTM

Kaiguo Xia [1], Zhisong Pan [2,*] and Pengqiang Mao [2]

[1] College of Communication Engineering, Army Engineering University of PLA, Nanjing 210001, China
[2] College of Command and Control, Army Engineering University of PLA, Nanjing 210001, China
* Correspondence: panzhisong@aeu.edu.cn

Abstract: Video compression sensing can use a few measurements to obtain the original video by reconstruction algorithms. There is a natural correlation between video frames, and how to exploit this feature becomes the key to improving the reconstruction quality. More and more deep learning-based video compression sensing (VCS) methods are proposed. Some methods overlook interframe information, so they fail to achieve satisfactory reconstruction quality. Some use complex network structures to exploit the interframe information, but it increases the parameters and makes the training process more complicated. To overcome the limitations of existing VCS methods, we propose an efficient end-to-end VCS network, which integrates the measurement and reconstruction into one whole framework. In the measurement part, we train a measurement matrix rather than a pre-prepared random matrix, which fits the video reconstruction task better. An unfolded LSTM network is utilized in the reconstruction part, deeply fusing the intra- and interframe spatial–temporal information. The proposed method has higher reconstruction accuracy than existing video compression sensing networks and even performs well at measurement ratios as low as 0.01.

Keywords: video compressing sensing; end-to-end deep learning network; unfolded LSTM; measurement matrix training

Citation: Xia, K.; Pan, Z.; Mao, P. Video Compressive Sensing Reconstruction Using Unfolded LSTM. *Sensors* **2022**, *22*, 7172. https://doi.org/10.3390/s22197172

Academic Editor: Lixiang Li

Received: 25 July 2022
Accepted: 17 September 2022
Published: 21 September 2022

1. Introduction

It is a dynamic world, and the video captures this world of objects and movement. The video is composed of continuous pictures. Generally speaking, the human eye can distinguish the frame rate is 15 frames per second [1]. More than 15 frames will be considered a motion video. The ordinary camera frame rate is generally between 30 and 60 fps, which can meet most cases of content recording, while high-speed cameras need to reach 120 fps or even higher. Although high-speed video offers rich details, it comes with higher storage space and transmission bandwidth usage.

Compressive sensing (CS) [2] can acquire the measurements of the original signal at a rate lower than the Nyquist sampling rate and use an algorithm to reconstruct the original signal. This feature makes it widely used in the video field. On the one hand, it can be applied to construct high-speed cameras. By compressing consecutive frames into one frame at one measurement, it is possible to use a plain low-speed camera sensor to achieve high-speed cameras, e.g., single-pixel camera [3], single-coded exposure camera [4], and coded strobing camera [5]. On the other hand, the VCS algorithm can alleviate the enormous demand for massive storage and transmission bandwidth required for video. VCS allows transmitting the video under 100× compression, significantly improving transmission efficiency and quickly reconstructing it at the receiving end.

According to the measurement of video frames, the existing VCS methods can be divided into temporal multiplexing VCS (TVCS) and spatial multiplexing VCS (SVCS). TVCS obtains a 2D measurement frame from sampling across the temporal axis, which superimposes k measurement frames into one frame. Its measurement ratio is $1/k$. The method proposed in [4,6–10] belongs to this category. These methods can obtain a high spatial resolution, which is generally implemented on sensors with low frame rates.

The SVCS is derived from single-pixel cameras [3]. It uses a programmable high-speed photodetector to measure the image. Only one measurement value is output for each measurement. A set of measurement values is obtained after multiple measurements, which is used to reconstruct the original video frame. Figure 1 shows the SVCS process. The size of the input video frame is $W \times H = N$, and it can be vectorized as $x \in R^N$. The measurement matrix (MM) is expressed as $\Phi \in R^{M \times N}$, which can be viewed as a set of M patterns, and each row of the MM corresponds to a vectorized pattern. The measurement process is to make the inner product of each pattern and the original signal, output a single measurement value, and obtain the measurement vector $y \in R^M$ after M times calculation. The SVCS is formulated as follows:

$$y = \Phi x \tag{1}$$

Compared with TVCS, the SVCS can reconstruct the original frames with fewer measurements, thus achieving a relatively low measurement ratio. As a result, it causes severe loss of image information in the measurement, leading to decreased reconstruction quality. In this paper, our proposed method belongs to SVCS. Therefore, making a tradeoff between measurement ratio and reconstruction quality is one problem to be solved.

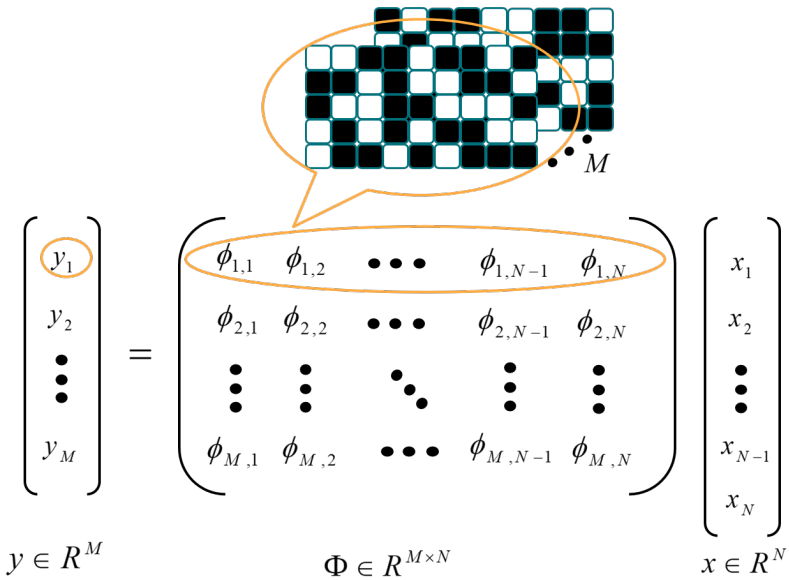


Figure 1. Principle of SVCS measurement process.

In the past ten years, many researchers have proposed VCS reconstruction algorithms based on optimization algorithms in the image field, e.g., [6,8,11,12]. Usually, these algorithms are based on the sparse prior of the image signal and use convex optimization or greedy algorithms to solve the reconstruction problem iteratively. These methods inevitably bring the problem of high computational complexity. As the resolution of the image increases, the time consumption for computation increases exponentially, making it challenging to meet real-time requirements. In recent years, with the wide application of deep learning, more and more methods based on deep learning have been proposed to solve the problem of VCS, e.g., [9,13–17]. These methods learn the inverse mapping directly from the measurements to the original signal through the neural networks. The reconstructed signal can be calculated through a feedforward network, which is less computationally complex than the iterative optimization algorithm. The real-time performance is improved, and the reconstruction quality is substantially improved.

The video consists of continuous frames. The video frame sequence contains the motion information of the objects in the scene. Due to the natural correlation between frames, it is possible to use spatial–temporal information, which is the key to improving the quality of reconstructed frames in the case of SVCS. In [16,18,19], the spatial–temporal information of video is fully considered when studying the VCS problem, and the spatial–

temporal features of video frames are extracted using deep networks to enhance the reconstruction quality. The CSVideoNet [16] uses a classical LSTM network to extract motion features of continuous frames. In [18], Hybrid-3DNet is proposed to extract spatial information by convolving video segments using a 3D convolutional network, while VCSNet in [19] uses CNN residual connections to transfer interframe information and achieve inter-frame multi-level compensation.

This paper uses an unfolded LSTM structure to model the spatial–temporal feature in the video frame sequence. This structure is first proposed in [20] for solving sparse Bayesian learning optimization problems by mapping the traditional process of iterative SBL to an LSTM structure. The model shows good convergence performance using the unfolded LSTM for sparse optimization, which can greatly accelerate the SBL optimization solution process. Inspired by this method, we try to use the unfolded LSTM model in the VCS problem, and it provides surprisingly good performance. In the experiment, we found that the structure can not only take advantage of the LSTM's property of long-time memory of sequences to efficiently fuse the spatial–temporal information from intra- and interframes but also fix the LSTM, which originally iterates according to the length of sequences, to a finite length. In that case, it essentially forms a feedforward network, which converges rapidly in the training process compared with the classical LSTM network. It can greatly reduce training time consumption and make the reconstruction process faster.

In addition, compared with CSVideoNet and VCSNet, since the proposed method does not adopt the multi-layer CNN structure, we use Xavier [21] to initialize the network parameters without pre-training, which makes our proposed network more efficient.

The contributions of this paper are summarized as follows:

- A unified end-to-end deep learning VCS reconstruction framework is proposed, including the measurement and reconstruction part, both of which consist of deep neural networks. We train the framework using a single loss function in a data-driven method, which makes the training process more efficient and faster;
- The spatial–temporal information fusion reconstruction with multiple sampling rates is accomplished by using the unfolded LSTM network, which obtains better reconstruction effects and improves the convergence performance significantly;
- Compared with the existing VCS methods, we demonstrate that the proposed network provides better reconstruction results under a wide range of measurement ratios, even under the measurement ratio as low as 0.01. We also experimentally demonstrate that the network has good convergence without pre-training and converges faster than the comparison methods.

2. Related Work

2.1. Conventional VCS Algorithm

From Eqution (1), the compressive sensing problem is reconstructing the original signal by solving an underdetermined equation, which is an NP-hard problem. There are some designed video signal priors for conventional VCS algorithms, e.g., Gaussian prior, sparse prior and TV prior, as regularization terms to the equations to constrain the possible solutions in a particular range. In [4], the Gaussian mixture model (GMM) is extended to CS territory. The method assumes that the pixels in video blocks obey the GMM distribution and use GMM to model the spatial–temporal video blocks. Video frames are reconstructed by E-M iteration. In [22], the author considers the total variational minimum can be used as the regularization. The generalized alternating projection (GAP) algorithm is used to solve the optimization problem. In [8], the author obtained better reconstruction accuracy by using low-speed image frames as side information to assist video reconstruction.

2.2. Deep Neural Network-Based VCS Algorithms

The DFC network in [9] is the first VCS reconstruction algorithm using a deep neural network. It uses a fully connected network to learn the direct mapping from the measurements to the original signal, solving the underdetermined problem using a data-driven

approach. Compared to conventional VCS algorithms, it reduces computational complexity and improves reconstruction quality. In [23], the author demonstrates that the learned measurement matrix is superior to the random matrix. It can achieve better reconstruction results by integrating the measurement matrix into a unified training framework and learning it through network training. However, using a simple ,fully connected network makes it difficult to exploit spatial–temporal information efficiently.

In order to solve the problems above, CSVideoNet [16] uses a CNN+LSTM structure to mine spatial–temporal information, enhancing the reconstruction quality. In order to balance the high compression rate and reconstruction accuracy, the video is grouped in fixed length, taking the first frame of each group as the keyframe with a low measurement ratio. The LSTM network is used to extract spatial–temporal features of the frame group to add motion estimation to the reconstruction process. CSVideoNet significantly improves the reconstruction quality. However, a pre-defined random measurement matrix in its measurement phase may not be adapted to the video task and limit performance improvement.

VCSNet [19] is a well-performing network composed entirely of a CNN structure. The training process contains multi-stages, with the measurement matrix being trained first, followed by the reconstruction part. The reconstruction part is further divided into two stages: keyframe training and non-keyframe training. A multi-level feature compensation method is used in these two parts, which compensates for the reconstruction of non-key frames by using high-quality keyframes. Using CNN networks with multi-level compensation can only compensate for the loss of spatial information in non-key frames. In contrast, the temporal information contained in the video is hard to be learned. In addition, up to 2K + 2 loss functions are used in the training process, and such a multi-loss network is difficult to train.

3. Video Compressive Sensing Reconstruction Using Unfolded LSTM

3.1. Overview of the Proposed Framework

This section will give the details of the proposed end-to-end VCS reconstruction network. The network structure is shown in Figure 2. It can be divided into encoding and decoding parts.

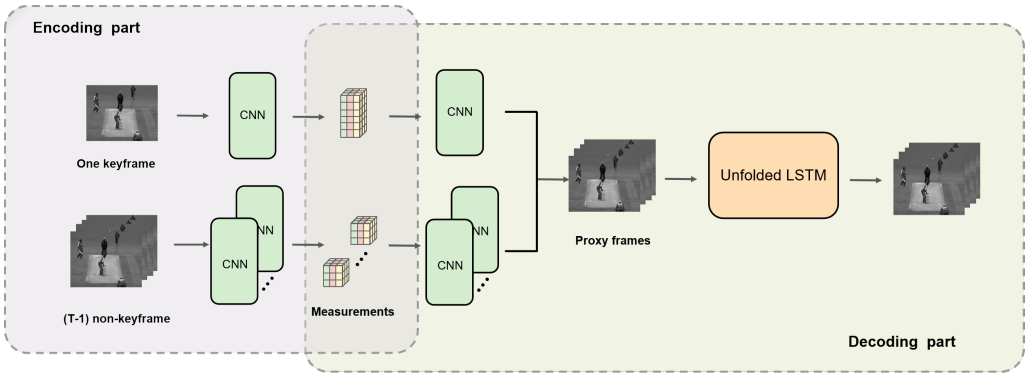


Figure 2. The architecture of the proposed framework.

The encoding part provides a learned measurement matrix, which is equivalent to a CNN's convolution process. Since the proposed network belongs to SVCS and will lose more spatial resolution in the measurement process due to the low measurement ratio, the keyframe technique is introduced. The input video is split into multiple groups of pictures (GOPs), with the first frame of each GOP as the keyframe and the rest as non-key frames. A higher measurement ratio is applied to the keyframe to retain more spatial information. In comparison, a lower measurement ratio for non-key frames reduces the overall sampling rate of the GOP.

The decoding part includes the initial reconstruction and the deep spatial–temporal information fusion reconstruction. Initial reconstruction is used in many VCS methods,

allowing a smoother transition to the subsequent enhanced reconstruction part. In the first stage of the decoding part, there is another CNN to perform an inverse convolution on the measurements to obtain an initial reconstructed frame, which has the same dimensions as the original frame. The proxy keyframes and non-key frames are input into the next stage, the unfolded LSTM network, for the dynamic information fusion reconstruction. The network can preserve the motion information in frame sequence and fully extract the spatial–temporal features, achieving high-fidelity reconstructed video.

For the training, we use a single mean square error (MSE) as the loss function, directly comparing the difference between the reconstructed and the original video segments. It makes the training process simple and efficient, avoiding the problems of parameter balance and poor convergence in multi-loss training.

3.2. Encoding Part

In many VCS algorithms, the measurement matrix uses a pre-defined random matrix for measurement. However, the study in [23] indicates that the measurement matrix learned from data has better performance in the reconstruction. The details of the encoding part are shown in Figure 3.

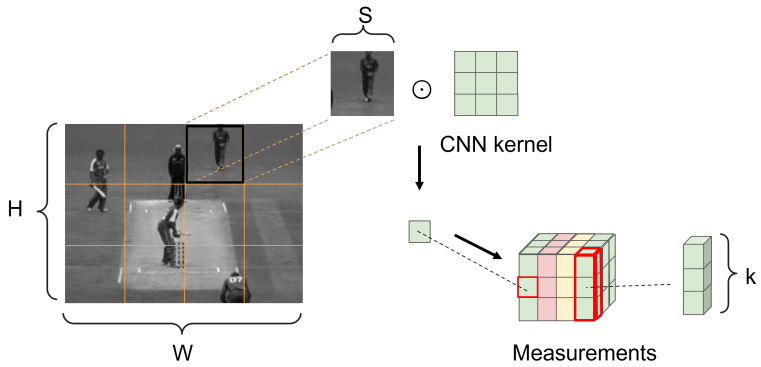


Figure 3. The measurement process based on CNN structure.

Each video frame size $W \times H$ can be divided into $w_p \times h_p = P$ image blocks of size without overlapping. We denote the ith patch as $x_i = [x_i^1, x_i^2, \cdots, x_i^M]$, where $M = s \times s$ denotes the pixels of each small block. As illustrated in Figure 2, for the ith patch, the measurement process can be formulated as:

$$y_i = \Phi x_i \tag{2}$$

where $\Phi = [\phi_1, \cdots, \phi_k]^T$, ϕ_j denotes the column vector, which is vectorized from the jth pattern. $y_i = [y_i^1, \cdots, y_i^k]$, y_i^j is the inner product of ϕ_j and x_i, which denotes as $y_i^j = \phi_j^T \bullet x_i$. The measurement process for each patch can be equated to convolution operation by CNN, which has k kernels of size $s \times s$ with step length s. We arranged the format in order (number of image channels, image height, image width); the size of the input is denoted as $(1, H, W)$ and output is denoted as (k, h_p, w_p) .

For a GOP of length T, the process from each video frame $X_i = [x_1, \cdots, x_P]$ to the measurements $Y_i = [y_1, \cdots, y_P]$ can be formulated as:

$$Y_i = \Phi X_i \tag{3}$$

where Y_i is of size (k, h_p, w_p) and X_i is of size $H \times W$; thus, the measurement ratio (MR) is calculated as:

$$MR = \frac{k \times h_p \times w_p}{H \times W} = \frac{k \times \frac{H}{s} \times \frac{W}{s}}{H \times W} = \frac{k}{s \times s} \tag{4}$$

From Eqution (4), it can be concluded that the size of the measurements can be easily controlled by the number of convolution kernels k. For each GOP containing T frames,

denoted k_1, k_2 be the number of convolutional kernels of the keyframe and non-keyframe, respectively; then, the global measurement ratio can be calculated as:

$$MR_{gop} = \frac{k_1 + k_2 \times (T-1)}{s \times s \times T} \tag{5}$$

3.3. Decoding Part

The decoding part can be divided into the initial reconstruction and the deep spatial–temporal information fusion reconstruction.

3.3.1. Initial Reconstruction

The initial reconstruction is used in many VCS algorithms. Firstly, a simple inverse transformation of the measurements is completed to obtain a proxy frame with the same dimensions as the target frame and then further refine the proxy frame in the next stage. Figure 4 shows the details. The initial reconstruction process can be seen as the inverse of the encoding part, converting measurements Y_i with dim $k \times h_p \times w_p$ to proxy frame $\tilde{X}_i$ with dim $1 \times H \times W$. Here, we focus on the inverse process. For each patch x_i, we can also use a CNN to obtain the proxy frame; just set the kernel of size $(s \times s) \times 1 \times 1$ and step length to 1. By convolving the measurements y_i, we obtain a column vector $\tilde{x}_i$ with dimension $1 \times 1 \times (s \times s)$. Reshaping $\tilde{x}_i$ to $1 \times s \times s$, we obtain the initial reconstruction of the patch. Extending the convolution operation to the whole frame, we can obtain the convolutional output of size $(s \times s) \times h_p \times w_p$. After reshaping, the proxy frame of size $1 \times H \times W$ is finally recovered.

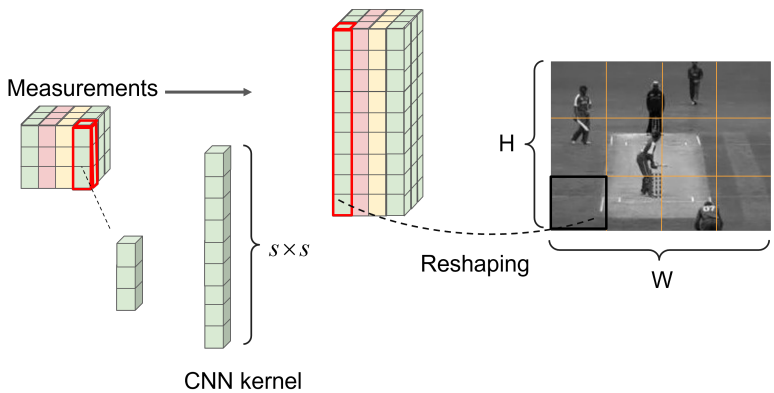


Figure 4. The process of the initial reconstruction.

3.3.2. Deep Spatial–Temporal Information Fusion

For a GOP of length T, which has undergone a measurement and initial reconstruction phase, we put one proxy keyframe $\tilde{X}_{key}$ and $(T-1)$ proxy non-keyframes $\tilde{X}^i_{\text{nonkey}}$ $(i = 1, 2, \cdots, T-1)$ together in sequential order and denote them as $\tilde{X}_{GOP}$. For continuous video frames, the interframes contain part of the scene invariance and the motion information of the objects. To fully use the spatial–temporal information, we propose a spatial–temporal information fusion module based on unfolded LSTM, which helps aggregate motion and spatial features for each frame.

Figure 5 shows the full details of the network. The unfolded LSTM network structure consists of LSTM cells. The update process is as follows:

$$
\begin{aligned}
c_j^{(t)} &= f_j^{(t)} \odot c_j^{(t-1)} + i_j^{(t)} \odot \tilde{c}_j^{(t-1)} \\
h_j^{(t)} &= o_j^{(t)} \odot \mathrm{Tanh}(c_j^{(t)}) \\
i_j^{(t)} &= \sigma(W_{i_j}[h_j^{(t-1)}, I_j^{(t)}]) \\
f_j^{(t)} &= \sigma(W_{f_j}[h_j^{(t-1)}, I_j^{(t)}]) \\
o_j^{(t)} &= \sigma(W_{o_j}[h_j^{(t-1)}, I_j^{(t)}]) \\
\tilde{c}_j^{(t)} &= \tanh(W_{c_j}[h_j^{(t-1)}, I_j^{(t)}])
\end{aligned}
\tag{6}
$$

where the subscript j indicates the index of the stacked layers of the LSTM, $I_j^{(t)}$ denotes the input of the first LSTM cell, $I_1^{(t)} = X_{GOP}^{(t)}$, $I_j^{(t)} = h_{j-1}^{(t)} (j > 1)$. $h_j^{(t)}$ and $c_j^{(t)}$ denote, respectively, the hidden state and memory cell of the LSTM cell of the jth layer at the t moment. $i_j^{(t)}, f_j^{(t)}, o_j^{(t)}$ correspond to the input gate, forget gate and output gate. We stack r layers of the LSTM cell to construct the LSTM stack module, denoted as $\mathcal{F}_{LSTM-Stacked}$, and the process of update can be formulated as:

$$
\begin{aligned}
&= \mathcal{F}_{LSTM-\text{ Stacked }}(H^{(t-1)}, \tilde{X}_{GOP}^{(t)}; \theta) \\
q^{(t)} &= h_r^{(t)}
\end{aligned}
\tag{7}
$$

where $H^{(t-1)} = [h_1^{(t-1)}, \cdots, h_r^{(t-1)}]$ contains all the hidden states of each LSTM cell in the stacking module, θ denotes the network parameters, and $q^{(t)}$ indicates the output of the stacking module, which is the hidden state of the last LSTM cell in the stack.

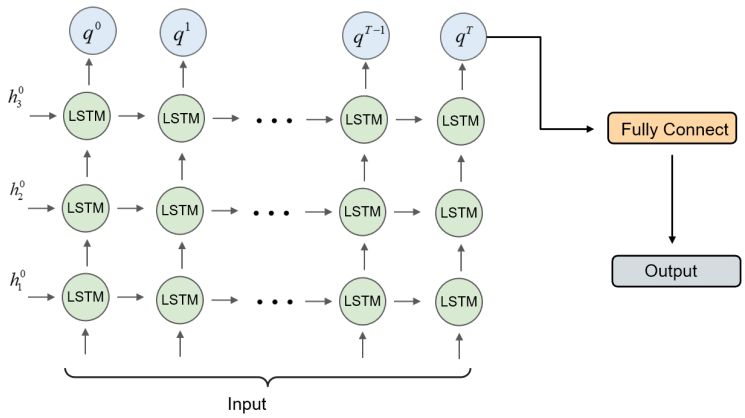


Figure 5. The process of the unfolded LSTM.

Unfolding the LSTM stack in l steps, we can obtain a feedforward network. During the forward stage, the input $\tilde{X}_{GOP}^{(t)}$ is broadcast to the lowest RNN cell at each unrolled step. The input of the hidden state of the unfolding step comes from the hidden state of the dth unfolding step. The output of $(d-1)^{th}$ the last unfolding step is passed through a fully connected network to obtain the final reconstructed video $\hat{X}_{GOP}$. Denoting the dth unfolding step as $\mathcal{F}_{LSTM-Stacked}^{d}$ and fully connected network as $\mathcal{F}_{FC}(\bullet)$, the process can be formulated as:

$$
\begin{aligned}
&= \mathcal{F}_{LSTM-Stacked}^{d}(H_{d-1}^{(t)}, \tilde{X}_{GOP}^{(t)}; \theta) \\
\hat{X}_{GOP} &= \mathcal{F}_{FC}(q^{(\text{ all })}) \\
q^{(\text{ all })} &= [q^{(1)}, q^{(2)}, \cdots, q^{(T)}]
\end{aligned}
\tag{8}
$$

3.4. Training

The VCS algorithm proposed is an end-to-end network architecture. The mean square error (MSE) is used as the loss function to train the whole network, which is defined as:

$$\text{LOSS} = \frac{1}{T}\sum_{i}^{T}\|\hat{X}_{GOP}^{i} - X_{GOP}^{i}\|_2^2 \tag{9}$$

Adam [24] optimizer is used for training to optimize the network parameters. Compared to the multi-stage training strategy VCSNet, the whole network is trained under a unified framework with only one global loss function, making the whole network training process concise and efficient. We use Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity (SSIM) [25] as the evaluation index of image reconstruction quality. PSNR, as a widely used metric for image reconstruction quality in engineering, is calculated based on the error between the corresponding pixels. The larger the value, the smaller the image distortion, which can reflect the final video reconstruction quality to a certain extent. SSIM is an evaluation metric based on the structural information of the scene, comparing the brightness, contrast and structural information in the image block, which to a certain extent can reflect the human eye's perception of picture similarity. The reconstruction quality can be better evaluated with these two metrics.

4. Experiment and Analysis

4.1. Dataset

UCF101 [26] is used as the dataset to benchmark our proposed network. The dataset collects and organizes 101 kinds of human action videos and contains up to 13,000 video clips with a duration of more than 27 h. Videos in the dataset have a resolution of 320 × 240 and are sampled at 25 fps. We selected 20 types of action videos as the training set and five types of entirely different action videos as the validation set. For comparison with the existing VCS algorithms, we crop the video frame into two different sizes of 96 × 96, 160 × 160 with the center part and split them into GOPs of length 10. We obtain 271,930 and 38,000 GOPs for training and testing, respectively.

4.2. Implementation Details

We set the keyframe measurement ratio MR_{key} to 0.5, 0.2, 0.1 for the different comparison experiments. According to Eqution (5), the corresponding non-key frame ratio MR_{nonkey} can be calculated based on the given MR_{GOP}. We set $s = 32$ and k_1, k_2 can be calculated according to Eqution (4). Our network is trained for 400 epochs with a batch size of 200. The network parameters are initialized using Xavier [21], and no pre-training is required. Adam is applied as the optimizer, and the learning rate is set to 0.001. Our network is trained for 400 epochs with a mini-batch size of 200.

4.3. Compared with Existing VCS Algorithms

In order to fully validate the effectiveness of the proposed method, we compare the proposed method with the existing VCS algorithms based on a deep neural network, including CSVideoNet [16], VCSNet [19], DFC [9], and C2B [27]. PSNR and SSIM are applied for performance evaluation.

In order to objectively compare the performance of the proposed algorithm with the existing methods, we set the same experimental condition as the comparison algorithms and compare the reconstruction quality separately. Since CSVideoNet and VCSNet all belong to SVCS, the proposed algorithm is compared with these two first.

The comparison with VCSNet is shown in Table 1. We use the parameter settings of VCSNet. Since keyframe MR is set fixed to 0.5, the length of GOP is 8 and non-key frame MR is set to 0.1, 0.01; the corresponding global MR of GOPs is 0.15, 0.07. The reconstruction quality of the proposed method is 2.52 db and 3.08 db higher than that of VCSNet under two different ratios, respectively.

Table 1. Comparison of the proposed method with VCSNet.

Name	Ratio	Frame Size	GOP	MR_{key}	MR_{nonkey}	PSNR	SSIM
VCSNet	0.15	96 × 96	8	0.5	0.1	34.29	0.90
Proposed		96 × 96	8	0.5	0.1	**36.91**	**0.96**
VCSNet	0.07	96 × 96	8	0.5	0.01	29.58	0.82
Proposed		96 × 96	8	0.5	0.01	**32.66**	**0.91**

The comparison with CSVideoNet is shown in Table 2. The experiment of CSVideoNet challenges the ultimate performance of VCS by setting three very low global MR values of 0.04, 0.02 and 0.01. We set the same experimental parameters, and it can be seen that the proposed method outperforms CSVideoNet by nearly 5 db on average in the three comparison groups.

Table 2. Comparison of the proposed method with CSVideoNet.

Name	Ratio	Frame Size	GOP	MR_{key}	MR_{nonkey}	PSNR	SSIM
CSVideoNet	0.04	160 × 160	10	0.2	0.022	26.87	0.81
Proposed		160 × 160	10	0.2	0.022	**32.64**	**0.88**
CSVideoNet	0.02	160 × 160	10	0.1	0.011	25.09	0.77
Proposed		160 × 160	10	0.1	0.011	**31.11**	**0.84**
CSVideoNet	0.01	160 × 160	10	0.06	0.004	24.23	0.74
Proposed		160 × 160	10	0.06	0.004	**28.64**	**0.81**

The VCS methods belonging to TVCS usually set the experiment to compress 16 original video frames into one measurement frame, where $MR = 1/16$ (0.0625). We compare the proposed algorithm with the existing algorithms under $MR = 1/16$, so we use $MR_{key} = 0.2$, $MR_{nonkey} = 0.047$, and correspondingly, $k_1 = 51$, $k_2 = 12$. The results are shown in Table 3, and our proposed method achieves the best performance. We also give some reconstructed frames as a visual comparison in Figure 6.

Table 3. Comparison of the proposed method with TVCS methods under ratio 1/16 (0.0625).

Name	Ratio	Frame Size	GOP	MR_{key}	MR_{nonkey}	PSNR	SSIM
DCF	1/16	160 × 160	16	-	-	24.67	0.71
C2B		160 × 160	16	-	-	32.23	0.93
CSVideoNet		160 × 160	10	0.2	0.022	28.08	0.84
VCSNet		160 × 160	10	0.2	0.022	28.57	0.86
Proposed		160 × 160	10	0.2	0.022	**35.02**	**0.95**

Obviously, through the above comparison experiments, we can conclude that the proposed algorithm performs better than the existing VCS methods under various MRs.

4.4. The Effectiveness of the Unfolded LSTM Structure

To demonstrate the effectiveness of this unfolded LSTM structure in the algorithm, we design comparative experiments where unfolded LSTM in the proposed algorithm is replaced with a classic LSTM structure. The experimental results are shown in Table 4. It can be seen that the average reconstruction accuracy is higher with the unfolded LSTM approach than with the classic LSTM, which indicates that the unfolded LSTM structure can improve the reconstruction quality better than the classic LSTM structure.

Table 4. Comparison of the proposed method with classical LSTM structure.

Name	Ratio	Frame Size	GOP	MR_{key}	MR_{nonkey}	PSNR	SSIM
LSTM	0.04	160 × 160	10	0.2	0.047	34.11	0.91
Proposed		160 × 160	10	0.2	0.047	**35.02**	**0.95**
LSTM	0.02	160 × 160	10	0.2	0.022	31.50	0.81
Proposed		160 × 160	10	0.2	0.022	**32.64**	**0.91**
LSTM	0.01	160 × 160	10	0.1	0.011	30.32	0.77
Proposed		160 × 160	10	0.1	0.011	**31.11**	**0.88**

4.5. Optimal Measurement Ratio Allocation for Keyframes and Non-Keyframes

We also conduct additional experiments to explore the measurement ratio allocation for keyframes and non-keyframes at the fixed MR_{GOP}. The experimental results are shown in Table 5. MR_{GOP} is set to 1/16. We found that appropriately reducing the measurement ratio of keyframes and increasing the measurement ratio of non-keyframes can improve the reconstruction results. We speculate that this phenomenon is due to a large amount of redundancy in the video frame information. A relatively low measurement ratio may exist for the keyframe to retain most of the spatial information.

Table 5. Comparison of the proposed method performance in different MR_{key} under the same ratio.

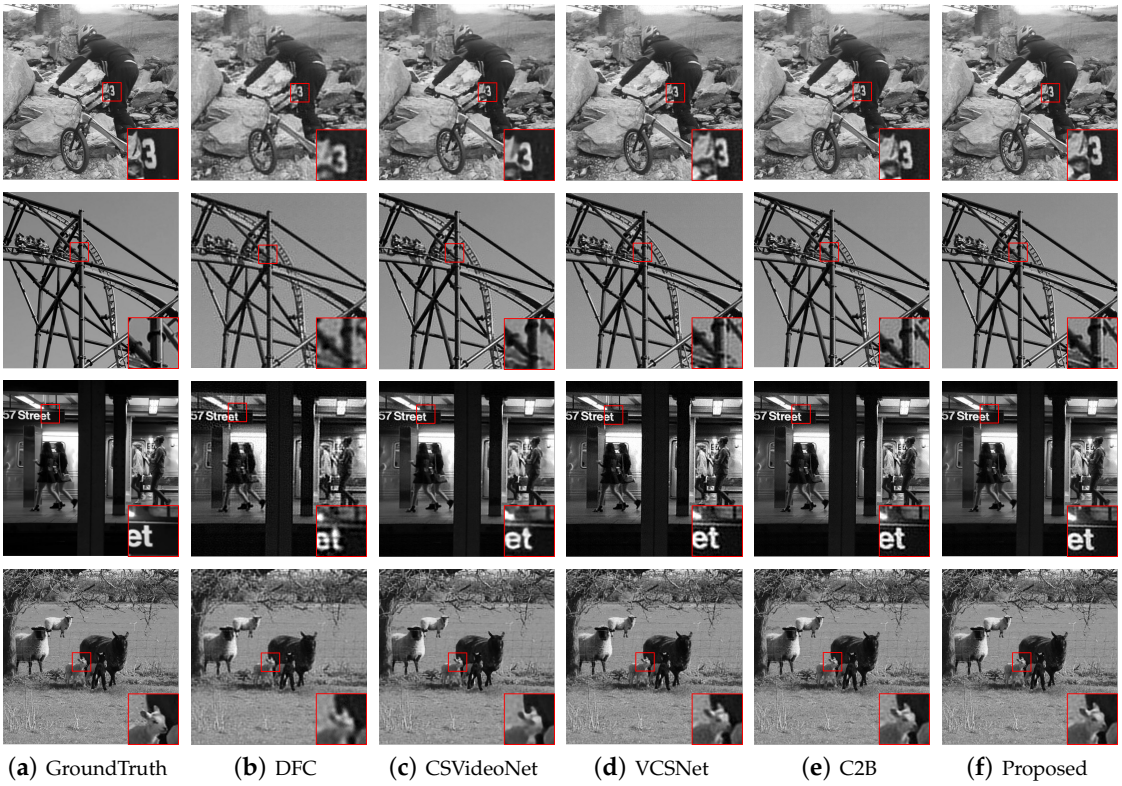
Frame Size	Patch Size	GOP	MR_{key}	MR_{nonkey}	k_1	k_2	PSNR	SSIM
96 × 96	32 × 32	10	0.5	0.014	512	14	31.88	0.81
96 × 96	32 × 32	10	0.4	0.025	409	25	33.59	0.83
96 × 96	32 × 32	10	0.3	0.036	307	37	34.58	0.86
96 × 96	32 × 32	10	0.2	0.047	204	48	**35.02**	**0.91**

4.6. Convergence Performance and Time Complexity

We designed experiments to compare the convergence performance of different methods. The same experimental conditions are set for each method, and then, we obtain the experimental results shown in Figure 7. The curves given are the variation of the reconstruction accuracy of the network on the validation set in each training epoch. From this figure, we can see that the reconstruction accuracy of CSVideoNet and VCSNet on the validation is are not higher than 20 db, indicating that the networks are easily trapped in local minima and challenging to converge to the optimum. The proposed method in this paper can converge to the appropriate interval after one epoch and achieves an accuracy of more than 30 db on the test set. Table 6 shows the time spent for each training epoch of the network during the training process. It takes much less time for the proposed network to train than CSVideoNet and VCSNet. From the above, the proposed network not only has better convergence performance in the training process but also has lower time complexity and higher overall efficiency.

Table 6. Comparison of the time complexity of the existing models and the proposed model.

Model	CSVideoNet	VCSNet	Proposed
Time(s)	587.70	359.61	**194.53**



(**a**) GroundTruth (**b**) DFC (**c**) CSVideoNet (**d**) VCSNet (**e**) C2B (**f**) Proposed

Figure 6. The comparison of reconstructed frames between the proposed and DFC, CSVideo, VCSNet and C2B methods.

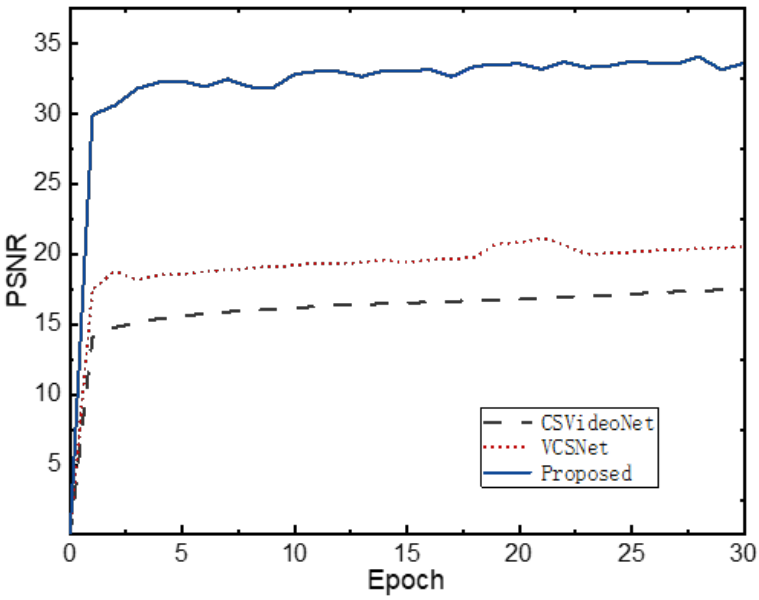


Figure 7. Variation curve of model reconstruction performance with the number of training epochs.

5. Conclusions

In this paper, we propose a novel unified end-to-end VCS reconstruction network. In the measurement sampling stage, the CNN network integrates the measurement process into the network and learns a better measurement matrix. In the reconstruction stage, the unfolded LSTM network is applied to fuse the spatial–temporal feature at the frames with different measurement ratios. Compared with the existing deep neural network-based VCS methods, the proposed network can achieve better reconstruction accuracy under the same

conditions and even obtain satisfactory reconstruction results at compression rates as low as 0.02 and 0.01. In future work, we will build on this and continue exploring computer vision downstream tasks such as target tracking, re-identification, etc.

Author Contributions: Conceptualization, K.X., Z.P. and P.M.; methodology, K.X. and Z.P.; software, K.X.; validation, K.X. and P.M.; writing—original draft preparation, K.X.; writing—review and editing, K.X., Z.P. and P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The database used in this article is UCF101. For details, please refer to [26].

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

References

1. Ou, Y.F.; Liu, T.; Zhao, Z.; Ma, Z.; Wang, Y. Modeling the impact of frame rate on perceptual quality of video. In Proceedings of the 15th IEEE International Conference on Image Processing (ICIP 2008), San Diego, CA, USA, 12–15 October 2008; pp. 689–692.
2. Candes, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [CrossRef]
3. Duarte, M.F.; Davenport, M.A.; Takhar, D.; Laska, J.N.; Sun, T.; Kelly, K.F.; Baraniuk, R.G. Single-pixel imaging via compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 83–91. [CrossRef]
4. Hitomi, Y.; Gu, J.W.; Gupta, M.; Mitsunaga, T.; Nayar, S.K. Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 287–294.
5. Veeraraghavan, A.; Reddy, D.; Raskar, R. Coded Strobing Photography: Compressive Sensing of High Speed Periodic Videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 671–686. [CrossRef] [PubMed]
6. Yang, J.B.; Yuan, X.; Liao, X.J.; Llull, P.; Brady, D.J.; Sapiro, G.; Carin, L. Video Compressive Sensing Using Gaussian Mixture Models. *IEEE Trans. Image Process.* **2014**, *23*, 4863–4878. [CrossRef] [PubMed]
7. Koller, R.; Schmid, L.; Matsuda, N.; Niederberger, T.; Spinoulas, L.; Cossairt, O.; Schuster, G.; Katsaggelos, A.K. High spatio-temporal resolution video with compressed sensing. *Opt. Express* **2015**, *23*, 15992–16007. [CrossRef]
8. Yuan, X.; Sun, Y.Y.; Pang, S. Compressive video sensing with side information. *Appl. Opt.* **2017**, *56*, 2697–2704. [CrossRef]
9. Iliadis, M.; Spinoulas, L.; Katsaggelos, A.K. Deep fully-connected networks for video compressive sensing. *Digit. Signal Process.* **2018**, *72*, 9–18. [CrossRef]
10. Qiao, M.; Meng, Z.; Ma, J.; Yuan, X. Deep learning for video compressive sensing. *Apl Photonics* **2020**, *5*, 030801. [CrossRef]
11. Zheng, J.; Jacobs, E. Video compressive sensing using spatial domain sparsity. *Opt. Eng.* **2009**, *48*, 087006. [CrossRef]
12. Dong, W.S.; Shi, G.M.; Li, X.; Ma, Y.; Huang, F. Compressive Sensing via Nonlocal Low-Rank Regularization. *IEEE Trans. Image Process.* **2014**, *23*, 3618–3632. [CrossRef] [PubMed]
13. Kulkarni, K.; Lohit, S.; Turaga, P.; Kerviche, R.; Ashok, A. ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 449–458.
14. Mousavi, A.; Dasarathy, G.; Baraniuk, R.G. DeepCodec: Adaptive Sensing and Recovery via Deep Convolutional Neural Networks. In Proceedings of the 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 3–6 October 2017; p. 744.
15. Shi, W.Z.; Jiang, F.; Zhang, S.P.; Zhao, D.B. Deep networks for compressed image sensing. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 877–882.
16. Xu, K.; Ren, F.B. CSVideoNet: A Real-time End-to-end Learning Framework for High-frame-rate Video Compressive Sensing. In Proceedings of the 18th IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1680–1688.
17. Chen, C.; Wu, Y.T.; Zhou, C.; Zhang, D.Y. JsrNet: A Joint Sampling-Reconstruction Framework for Distributed Compressive Video Sensing. *Sensors* **2020**, *20*, 206. [CrossRef] [PubMed]
18. Zhao, Z.F.; Xie, X.M.; Liu, W.; Pan, Q.Z. A Hybrid-3D Convolutional Network for Video Compressive Sensing. *IEEE Access* **2020**, *8*, 20503–20513. [CrossRef]
19. Shi, W.Z.; Liu, S.H.; Jiang, F.; Zhao, D.B. Video Compressed Sensing Using a Convolutional Neural Network. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 425–438. [CrossRef]
20. He, H.; Xin, B.; Ikehata, S.; Wipf, D. From Bayesian Sparsity to Gated Recurrent Nets. In Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 5560–5570.

21. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
22. Yuan, X. Generalized alternating projection based total variation minimization for compressive sensing. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2539–2543.
23. Iliadis, M.; Spinoulas, L.; Katsaggelos, A.K. DeepBinaryMask: Learning a binary mask for video compressive sensing. *Digit. Signal Process.* **2020**, *96*, 102591. [CrossRef]
24. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, 7–9 May 2015.
25. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
26. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv* **2012**, arXiv:1212.0402.
27. Shedligeri, P.; Anupama, S.; Mitra, K. A Unified Framework for Compressive Video Recovery from Coded Exposure Techniques. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 1599–1608.