Research on Video Compression Algorithm Based on Deep Learning

Xian Wang[1](✉) and Yu Chen[2]

[1] Xiamen Institute of Technology, Xiamen 362000, Fujian, China
sakurawx@163.com
[2] Xiamen Weiya Intelligent Technology Co., Ltd, Xiamen 362000, Fujian, China

Abstract. The storage of any type of data on the web requires the use of that particular type of data. The amount of pictures, movies, and other forms of content that are similar to one another that are circulating on the internet has skyrocketed. Even under the weight of many resource constraints, such as bandwidth bottlenecks and noisy channels, users of the Internet demand the data they access to be easily understood. As a direct result of this, data compression is rapidly becoming an increasingly important topic among the larger engineering community. Using deep neural networks for the purpose of data compression has been the subject of some previous research. Several different machine learning approaches are now being implemented into data compression strategies and put to the test in an effort to achieve improved lossy and lossless compression outcomes. The typical video compressive sensing reconstruction algorithm has an excessively lengthy delay throughout the reconstruction process. It is unable to make full use of the spatial and temporal correlation of video, which results in an improvement in the quality of the reconstruction poor. In this research, a video compression method that is based on deep learning (DL) is proposed, and it has the potential to handle these challenges effectively.

Keywords: Data compression · Video compression · Machine learning · Deep learning

1 Introduction

Video makes up the vast majority of the data that is being produced in the globe today [1–3]. The primary goal of compression techniques is to reduce the total number of bits necessary to code the data or information that is being provided, which in turn reduces the amount of memory that is necessary to retain the data. Traditional methods of data compression algorithms involve the creation of codec pairs by hand. These codecs are referred to as codecs. The users are unsure whether the data has been compressed or whether it has degraded gracefully. These methods were designed specifically for bitmap graphics (images organized into a grid of color dots called pixels). They are not adaptable to the many formats used by modern media.

Deep neural networks (DNN) have been responsible for a number of significant advancements in recent years, particularly in the area of image compressed sensing

Q. Liu et al. (Eds.): CENet 2022, LNEE 961, pp. 1371–1378, 2022.
https://doi.org/10.1007/978-981-19-6901-0_145

reconstruction. To realize image block compressive sensing reconstruction, Reference [4] used for the first time the method of deep neural networks to construct a reconstruction network composed of fully connected layers and convolutional layers. This resulted in an improvement in reconstruction quality and a reduction in reconstruction time by several stars. The literature [5] proposed a DL-based image compressed sensing algorithm (DCSNet), set up a learnable convolutional sampling network to retain more effective information, and used convolutional full image reconstruction at the reconstruction end. Structure, effectively reducing the block effect. a learnable convolutional sampling network to retain more effective information. The literature [6] combines the classic iterative threshold shrinkage technique (ISTA) [7]. This approach produces high-quality and quick image reconstruction, and it has a certain level of theoretical interpretability. The aforementioned algorithms not only demonstrate the superiority of DL in terms of image compressive sensing and reconstruction, but they also present ideas that may be used in the creation of algorithms for video compressive sensing and reconstruction. An end-to-end video compressed sensing reconstruction algorithm called CSVideoNet was first proposed in Reference [8]. This algorithm uses a multi-layer convolutional layer to perform a simple initial reconstruction of a single frame image, and then uses a long short-term memory network for synthetic motion estimation. CSVideoNet was developed by the authors of Reference [8]. The (LSTM) algorithm realizes the flow of temporal information by transferring the detailed information that is present in key frames to frames that are not key frames. However, it is challenging to describe pixel spatial correlation using LSTM, and the training process is also challenging. On the basis of CSNet1's reconstruction, the piece of literature [9] presents a multi-level feature compensation convolution network. However, it is challenging to mine the accurate motion information included inside video signals using a neural network that is based on convolution. The performance of the reconstruction is very low for sequences of quick and complex motion.

Some academics have recently released two papers on thumbnail and full-resolution picture compression [10, 11], using recurrent neural networks, inspired by recent developments in DL. These researchers used recurrent neural networks [12–15] (RNNs). It uses a network that has been trained to represent blocks with a specific bit depth. The residuals that are created between the input blocks and the representations that are based on the trained network will be supplied into the next stage utilizing the trained network that was used in the previous stage. This is what is known as the round route. The encoded values for the coefficients correspond to each step in the process. The higher the number of steps, the greater the number of bits that are needed for the block's representation, and the higher the quality of the reconstruction. We are able to see that a decomposition employing an adaptive transform basis is being used when the trained network-based signal block is being used at each stage.

2 Method

In this part, we will introduce each part of the proposed method in detail.

The flow chart of the video compression technique used in DL is displayed in Fig. 1.

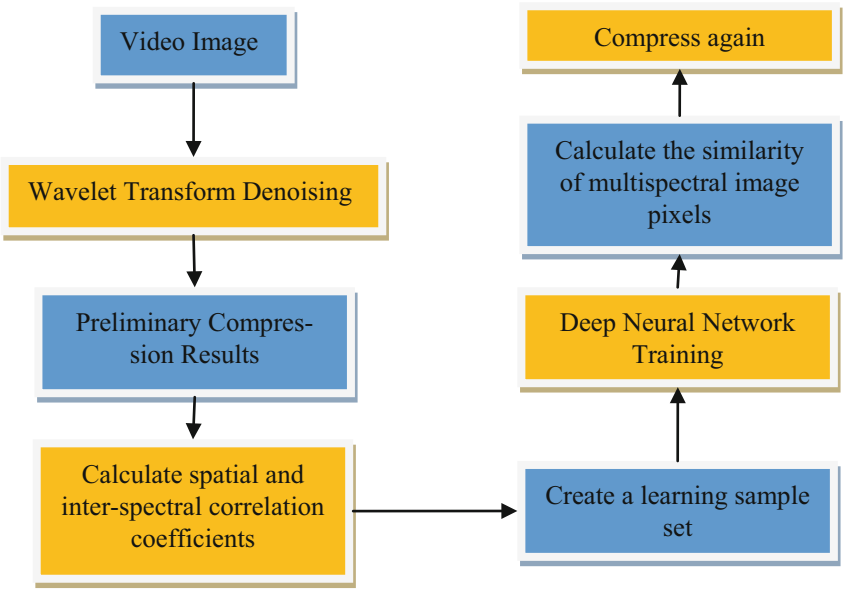

Fig. 1. DL video compression algorithm process

2.1 Multispectral Image Denoising Based on Wavelet Transform

The wavelet mother function is denoted by the symbol u(t). In the event that there is $u(t) \in F^2(R)$, then the following form must be satisfied by its Fourier transform.

$$H_u = \int_{-\infty}^{+\infty} |u(w)|^2 w^{-1} dw < \infty \tag{1}$$

We perform translation and expansion operations on the wavelet function in order to obtain a wavelet basis function set $\{u_{x,y}(t)\}$ whenever the positive and negative are alternated. This is necessary due to the wavelet function's volatility and oscillation, which we see when the positive and negative are switched.

$$u_{x,y}(t) = x^{-\frac{1}{2}} u(\frac{t-y}{x}) \tag{2}$$

where x and y each stand for the translation factor and the scaling factor, respectively

2.2 DL

At the moment, the convolutional neural network is one of the most used DL algorithms. There are some parallels to be seen with the work done by the BP neural network. It is similar to BP neural networks in that it has signal forward propagation and error back propagation, but its structure is entirely distinct from that of BP neural networks. Convolutional layers and sampling layers make up the majority of the levels, and there

are a great deal of convolutional layers and sampling layers. The level of difficulty of the issue is the primary factor that decides the precise number. In most cases, a sample layer is required to come immediately after a convolutional layer. When a convolutional neural network is forward propagating a signal, it will typically use the output of the currently active layer as the input of the layer below it. It will then use the activation function to convert the input, calculate the output of the layer below it, and continuously transmit the signal layer by layer. The formula for the layer's output computation can be represented as

$$a^l = f(w^l a^{l-1} + b^l) \tag{3}$$

The number of layers that the convolutional neural network has is represented by the letter l in the formula.

The error that exists between the output layer node output y_j^i of the convolutional neural network and the actual output value Y_j^i can eventually be measured thanks to the continuous transmission of the signal. The error is

$$e^n = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{k} (Y_j^i - y_j^i) \tag{4}$$

In the formula, the number of samples is denoted by the letter n, and the number of sample categories is denoted by the letter k.

According to the back-propagation of the mistake, the corresponding update operation is carried out on the convolutional layer, and the convolutional layer can be described as

$$a_j^l = f(\sum_{i \in P_j} a_i^{l-1} \times k_{ij}^l + b_j^l) \tag{5}$$

The size of the feature map can be altered by the use of downsampling, and the formula is as follows:

$$a_j^l = f(\eta_i^l d(a_i^{l-1}) + b_j^l) \tag{6}$$

where d refers to the function used for downsampling.

2.3 Data Sources

Our data comes from the freely available Open Images collection. This is a dataset consisting of over 9 million picture URLs that span hundreds of class image-level label bounding boxes and have annotations attached to them. A training set consisting of 9,011,219 photos, a validation set consisting of 41,260 images, and a test set consisting of 125,436 images are included in the dataset. Figure 3 depicts an example graph of the dataset that was provided (Fig. 2).

Fig. 2. Dataset example

3 Experiment

In this research, we refer to the method that we developed for DL video compression as the OUR algorithm. The algorithm for performing the K-L transform is renamed the K-L algorithm. The Multispectral Image Compression Algorithms are referred to by their acronym, MIC, which stands for algorithm.

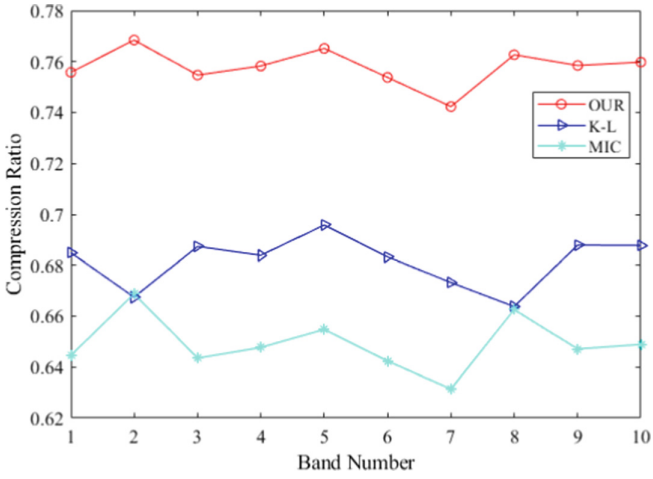

Fig. 3. Multispectral Image Compression Ratio Variation Curve

These three algorithms were evaluated side by side. Figure 3 presents a comparison of the compression ratios achieved by each of the three algorithms while using filters

with varying bands. Figure 3 reveals that the compression ratio of the OUR method is superior to that of the K-L algorithm and the MIC algorithm when applied to filters for a variety of bands.

The levels of precision and recall achieved by the test set when subjected to a variety of different algorithms are compared in Table 1. It is clear to see that the accuracy of the OUR algorithm is 85.27%, which is a modest improvement over the K-L algorithm by 0.09% and a significant improvement over the MIC algorithm by 0.75%. The outputs of the K-L algorithm and the MIC method have a lower recall rate than those produced by the OUR algorithm, which has a recall rate that is superior. This demonstrates that the OUR method is the best out of the three algorithms, and that the generalization performance of OUR model is superior than that of the other models.

Table 1. Accuracy and recall under different algorithms

Algorithm	OUR	K-L	MIC
Accuracy	85.27%	85.18%	84.52%
Recall	87.40%	82.94%	84.84%

The efficiency with which multi-spectral images can be compressed is an essential criterion to consider when assessing the effectiveness of various techniques for multi-spectral image compression. In this study, the average time required for multi-spectral picture compression was chosen as the metric to use when describing the effectiveness of the compression technique. Figure 4 presents the findings of this study. When the multi-spectral image compression time of the OUR algorithm is compared to the multi-spectral image compression time of the K-L algorithm and the MIC algorithm in Fig. 4, it can be seen that the multi-spectral image compression time of the OUR algorithm is greatly reduced, and the multi-spectral image compression efficiency is improved. This

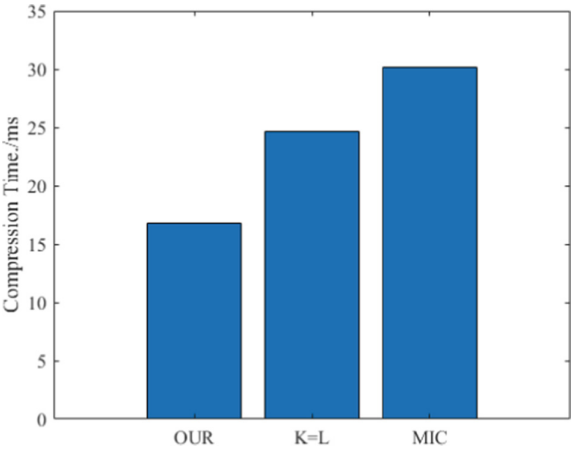

Fig. 4. Efficiency comparison of different algorithms

can be determined through comparison and analysis of the average time of multi-spectral image compression.

4 Conclusion

The application of approaches that are based on machine learning fulfills our requirements in this setting now that we live in the era of big data. Numerous machine learning algorithms are capable of carrying out a variety of tasks, including regression, classification, clustering, decision trees, extrapolation, and many others. In machine learning, algorithms are trained to extract information from data in order to carry out tasks that are data-dependent. During the process of developing these algorithms, a number of different machine learning methods, including supervised learning, unsupervised learning, reinforcement learning, and others, may be utilized. It is expected that researchers will concentrate on this problem while also offering new approaches. We propose a method for video compression that is based on DL.

An algorithm for DL video compression is proposed, and it makes use of both the DL adaptive optimization concept and the classic CVS multi-hypothesis motion compensation technique. The intra-frame image produced by this algorithm makes use of residual reconstruction blocks in order to compensate for detail information. This not only produces better initial reconstruction results for video compressive sensing reconstruction, but it can also be applied to video inter-frame reconstruction networks that make use of observations in order to perform reconstruction. Correction. The accuracy of the prediction is improved by the algorithm. The findings of the experiments indicate that the video compression method presented in this research that is based on DL has superior performance in reconstruction when compared to the great video compression sensing reconstruction algorithm that is currently in use.

Acknowledgements. 1. Middle-aged Education and Scientific Research project of Fujian Province. JAT1909561.

2. Computer Science and Information Engineering School's Fund for Scientific Research, Xiamen Institute of Technology.

References

1. Bulao, J.: How much data is created every day in 2021. techjury (2021)
2. Munson, B.: Video will account for 82% of all internet traffic by 2022, Cisco says (2018)
3. Cisco, U.: Cisco annual internet report (2018–2023) white paper. San Jose, CA, USA, Cisco (2020)
4. Kulkarni, K., Lohit, S., Turaga, P., et al.: Reconnet: non-iterative reconstruction of images from compressively sensed measurements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 449–458 (2016)
5. Shi, W., Jiang, F., Zhang, S., et al.: Deep networks for compressed image sensing. In: 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 877–882. IEEE (2017)
6. Zhang, J., Ghanem, B.: ISTA-Net: interpretable optimization-inspired deep network for image compressive sensing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1828–1837 (2018)

7. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. J. Issued Courant Inst. Math. Sci. **57**(11), 1413–1457 (2004)
8. Xu, K., Ren, F.: CSVideoNet: a real-time end-to-end learning framework for high-frame-rate video compressive sensing. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1680–1688. IEEE (2018)
9. Shi, W., Liu, S., Jiang, F., et al.: Video compressed sensing using a convolutional neural network. IEEE Trans. Circuits Syst. Video Technol. **31**(2), 425–438 (2020)
10. Toderici, G., O'Malley, S.M., Hwang, S.J., et al.: Variable rate image compression with recurrent neural networks. arXiv preprint arXiv:1511.06085 (2015)
11. Toderici, G., Vincent, D., Johnston, N., et al.: Full resolution image compression with recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5306–5314.(2017)
12. Medsker, L.R., Jain, L.C.: Recurrent neural networks. Des. Appl. **5**, 64–67 (2001)
13. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
14. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning. PMLR, pp. 1310–1318 (2013)
15. Pascanu, R., Gulcehre, C., Cho, K., et al.: How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026 (2013)