electronics

MDPI

Article

Impact of Video Compression and Multimodal Embedding on Scene Description

Jin Young Lee

School of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea; jinyounglee@sejong.ac.kr; Tel.: +82-2-6935-2672

Received: 22 July 2019; Accepted: 28 August 2019; Published: 30 August 2019

Abstract: Scene description refers to the automatic generation of natural language descriptions from videos. In general, deep learning-based scene description networks utilize multimodalities, such as image, motion, audio, and label information, to improve the description quality. In particular, image information plays an important role in scene description. However, scene description has a potential issue, because it may handle images with severe compression artifacts. Hence, this paper analyzes the impact of video compression on scene description, and then proposes a simple network that is robust to compression artifacts. In addition, a network cascading more encoding layers for efficient multimodal embedding is also proposed. Experimental results show that the proposed network is more efficient than conventional networks.

Keywords: deep learning; video compression; multimodal embedding; scene description

1. Introduction

Scene description networks, which input a sequence of images and output a sequence of words, have been recently designed using long short-term memory (LSTM) models [1]. Because LSTM stores long-term temporal information, it is widely used for sequence-to-sequence modeling. LSTM was designed to overcome the vanishing gradient problem. Its memory cell can capture previous information until the current time, and then update the current feature under three gates, which consist of input, forget, and output gates. In addition, LSTM-based networks can embed various modalities, such as image, motion, audio, and label information, to model the richness of the language employed in human-generated descriptions. Figure 1 shows an example video with a human-annotated sentence "a group of swimmers are swimming fast". For example, the words "a group of swimmers" can be simply recognized from the image information. However, the words "are swimming fast" are strongly correlated with motion information. If the video contains the sound of swimming, audio information will be very useful in generating the sentence. If the video contains label information, such as sports, it will be easy to predict that the sentence that will be generated will be related to swimming. Hence, the various modalities can complement each other in translating videos to language.

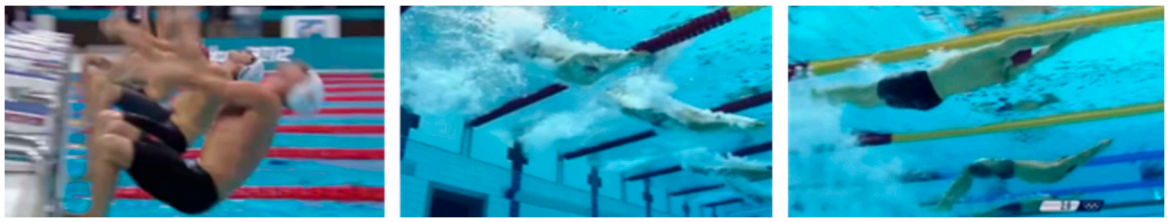

Figure 1. Example video with a human-annotated sentence—"a group of swimmers are swimming fast".

Early deep learning-based scene description networks mainly used image information. A mean pooling based network, proposed in [2], employs an average of features extracted from image information. Because the average feature cannot consider an overall temporal structure, sequence to sequence video to text (S2VT), introduced in [3], embeds the image features from each instance in an encoder–decoder framework to enhance accurate sentence generation. In order to achieve a better description performance, a multimodal video description network (MMVD) [4], which is an extended version of S2VT, considers multimodal information features, such as image, motion, audio, and label information. As shown in Figure 2, it uses LSTM encoding and decoding layers. An encoder simultaneously inputs a sequence of the multimodal features. For instance, the features are concatenated at each time step, to generate a single feature vector as an input of LSTM. This single vector is encoded to a hidden state vector, which is represented as a fixed-length vector. A decoder outputs a sequence of words from not only the encoded representation vector but also previous words, as shown by the dashed line in Figure 2. The previous words represent ground truth words in a training phase, but words with the maximum probability after a softmax during inference. The LSTM parameters in MMVD are updated between the encoder and the decoder in such a way that the log-likelihood of the predicted words can be maximized. Finally, a sentence is completed by collecting the words that are an output of each LSTM.

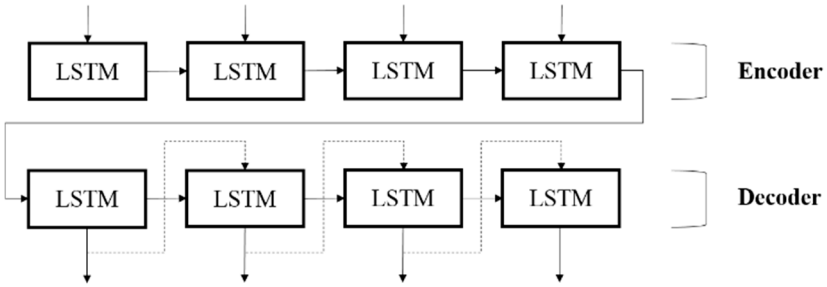

Figure 2. MMVD designed with the encoder and decoder.

In order to further improve MMVD, a multirate multimodal video network was introduced to encode images with different time intervals and consider motion speed differences to understand a temporal structure of the videos [5]. A network proposed in [6] mines multimodal topics in an encoder, guides a decoder with these topics, and then generates topic-aware descriptions. A multimodal attention network introduced in [7] employs a temporal attention to selectively focus on multimodal information. A category-aware ensemble model proposed in [8] employs efficient fusion of different description models.

However, the previous studies only focused on the improvement of the scene description network itself. They assume that the quality of input videos is always sufficient to generate accurate descriptions. In general, the video quality is degraded by distortions that commonly occur during capture, storage and transmission of videos. To overcome this problem, data augmentation methods were introduced to enrich the dataset used in network training [9–12]. Particularly, videos are compressed to satisfy the bitrate condition in a limited bandwidth. If the target bitrate is high enough, the video quality will have no problem in the scene description. On the other hand, if the target bitrate is low, the videos will be strongly compressed. This results in severe quality degradation, because of compression artifacts. However, if the network is already well-trained in a given training dataset and is ready to be used for the scene description, it is not practical to train the network with the data augmentation again. In this paper, the proposed network embeds additional features extracted from videos of various quality to deal with this problem during inference.

In addition, various features, such as image, motion, audio, and label information, are embedded in the same encoding layer in the previous multimodal networks [5–8]. Regardless of the number of features, they are embedded together. However, embedding each feature in the different layers, by cascading more encoding layers to the network, may be considered. When a new feature is embedded,

the proposed network simply stacks one more encoding layer. Experimental results indicate that the proposed network cascading the multiple layers for the efficient multimodal embedding can improve the description performance.

The remainder of this paper is organized as follows. Section 2 introduces video compression in detail. In Section 3, efficient multimodal embedding is discussed. Finally, experimental results are presented and the paper is concluded in Sections 4 and 5, respectively.

2. Video Compression

As images and videos are always transmitted in a limited bandwidth, image compression and video compression are one of the important issues in multimedia applications. To transmit a single image, image compression standards, such as JPEG (joint photographic experts group) [13], JPEG2000 [14], and JPEG-LS [15], were developed. In order to compress consecutive images in videos, H.264/AVC (advanced video coding) [16] and HEVC (high efficiency video coding) [17] are widely used as video compression standards. Both H.264/AVC and HEVC adopted a block-based video compression scheme with advanced technologies [18–21]. First, each block is predicted from already compressed images. Second, an original block is subtracted from the predicted block. Third, residuals within this subtracted block are transformed into frequency coefficients. Fourth, the transformed coefficients are quantized to reduce their magnitude. Finally, the quantized coefficients are losslessly compressed by entropy encoding. When a decoder receives the coefficients sent from an encoder, entropy decoding, inverse quantization, and inverse transform are performed in order. For reconstruction, the residuals generated after the inverse transform are added into a block predicted from previously reconstructed images. During the video compression process, quantization leads to data loss that is irreversible, so inverse quantization cannot recover the original data. Figure 3 shows a general video compression process.

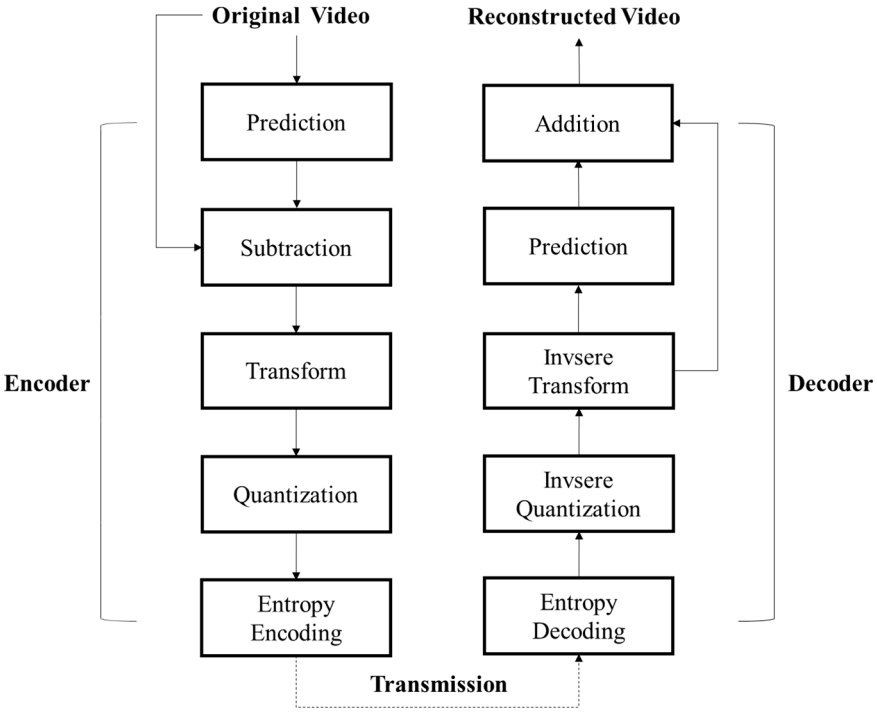

Figure 3. General video compression process.

In general, both H.264/AVC and HEVC encoders can determine compression types according to target applications. If videos are used in error sensitive applications, such as medical imaging, lossless compression is performed. Otherwise, lossy compression is performed in most cases. If an original video is losslessly compressed, there will be no quality degradation during the compression. Hence, the quality of the original and reconstructed videos will be same. However, if the lossy compression is

used, the quality degradation of the reconstructed video will depend on a quantization parameter (QP) value, which directly decides how much the original video is compressed. The QP can have values between 0 and 51. If a target bitrate is high, the QP should have low values. If a target bitrate is low, the QP should have high values. Usually, high QP values generate significant distortions.

Figure 4 indicates the first images in reconstructed video examples that were compressed with QPs of (a) 20, (b) 40, and (c) 50, and difference images between the original and reconstructed images. The brighter the pixel value in the difference images, the more severe the compression distortion. As shown in Figure 4, the quality of the reconstructed video is significantly low, when QP is equal to 40 or 50. In particular, compression artifacts, such as blocking artifact, are visible. However, when QP is equal to 20, the quality difference is negligible. Therefore, it can be concluded that the compression artifacts significantly deteriorate the quality of the reconstructed video, when the high QP values are used. It should be noted that other coding parameters can generate the different compression artifacts, such as temporal flickering, but only the blocking artifacts are considered in this paper. Meanwhile, since most of videos are stored in a compressed format, scene description networks will mainly use the reconstructed videos rather than the original videos. If the quality of these reconstructed videos is very low because of the compression artifacts, it will have a negative influence on the description quality of the scene description networks. The previous networks are vulnerable to the compression artifacts, because both training and testing datasets used in the experiments do not contain severe compression artifacts. Based on this observation, this paper proposes a simple network that is robust to compression artifacts.

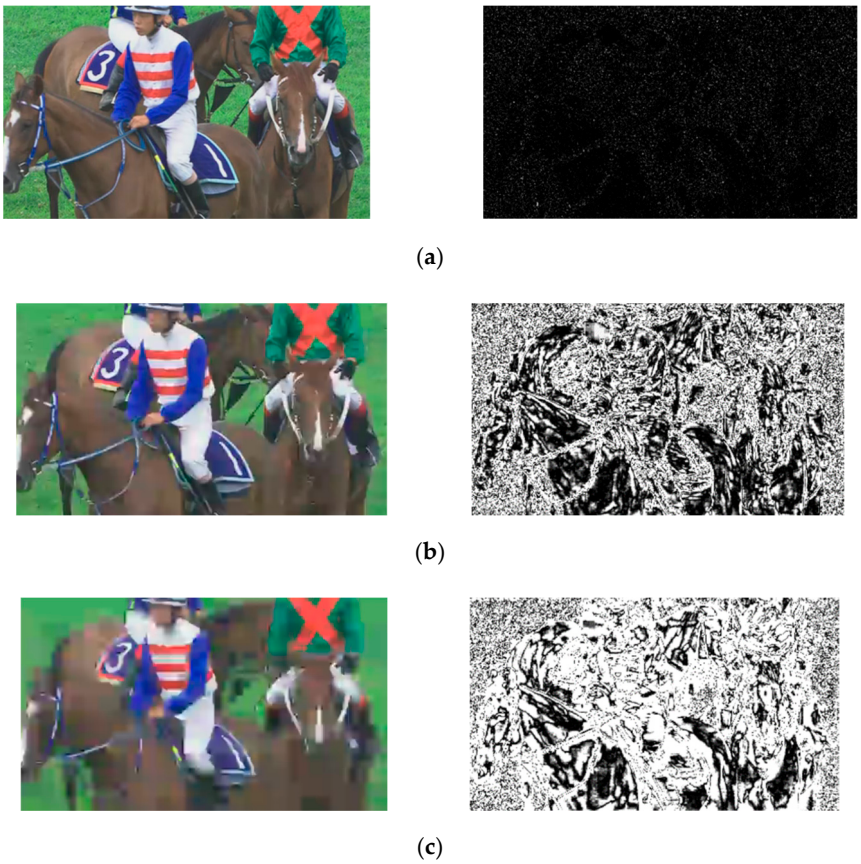

Figure 4. First images in reconstructed video examples compressed with various QPs (left) and difference images between the original and reconstructed images (right). (**a**) Reconstructed image with a QP of 20 and its difference image; (**b**) reconstructed image with a QP of 40 and its difference image; (**c**) reconstructed image with a QP of 50 and its difference image.

Figure 5 illustrates the proposed network considering videos with n qualities. It is assumed that the network is already trained with a given training dataset, and the quality of the training dataset is unknown. First, the proposed network compresses an original video with n different QPs, as shown in Figure 5. Therefore, n reconstructed videos from r_1 to r_n are generated. For example, if n is set to four, four different quality videos from r_1 to r_4 can be reconstructed with QPs of 0, 20, 40, and 50. Setting the QP to 0 indicates that lossless coding is being performed. Hence, r_1 will be the same as the original video. Second, these four features are embedded into the network encoder. If the video quality of the training dataset is high, the r_1 and r_2 features will be helpful for generating an accurate description. On the other hand, if the video quality is very low, the r_3 and r_4 features will be useful. Finally, an ensemble model selects the words with the maximum probability among the n candidate words from cw_1 to cw_n generated from the network decoder. Similarly, if the quality of the training dataset is high, it is highly possible that cw_1 or cw_2 is determined. On the other hand, if the quality is low, cw_3 or cw_4 will be chosen. To summarize, as the video quality of the dataset employed for the network training is unknown, the proposed network compresses the original video with n QPs, inputs the n reconstructed videos of various quality, and then outputs the optimum words among the n candidates. MMVD was employed for the experiments in this paper, as the scene description network. However, the proposed method can be easily applied to all previously developed networks by simply adding the video compression process that generates the n inputs right before the networks, and the ensemble model that finds the optimum sentence among the n outputs right after the networks.

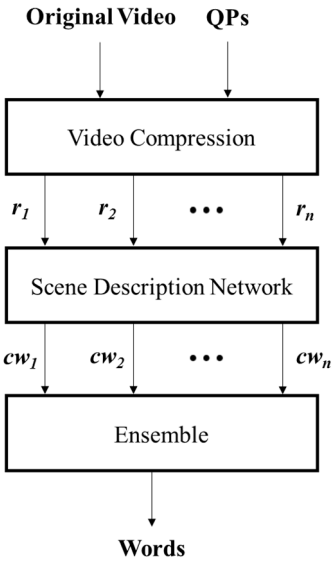

Figure 5. Consideration of the n reconstructed videos of various quality in the scene description.

3. Multimodal Embedding

As a video contains various information, such as image, motion, audio, and label information, it should be embedded to a scene description network with a predefined rule. In MMVD, as displayed in Figure 2, they are embedded in the same encoding layer, regardless of the number of features. The number of encoding and decoding layers is only one. Most previous studies have designed networks similar to MMVD, which embeds the various information features into the single encoding layer. However, through exhaustive experiments, it was observed that each information feature can be embedded in different encoding layers to improve the description performance. Therefore, the proposed network embeds each feature into the different layers by cascading the encoding layers in the encoder, while maintaining to use a single layer in the decoder to reduce the computational complexity. In addition, it was also observed that proper embedding of the multimodal features contributes to improvement of the description performance.

Figure 6 indicates the proposed network, when only one feature is available. The first encoding layer inputs the feature and encodes it to the hidden representation vector. The second layer encodes the vector again. The decoder performs the same as MMVD. The reason why the proposed network stacks one more encoding layer is to deep-encode an input information feature. When the multimodal features are embedded, it is very helpful if the last layer encodes them once more, right before the decoder starts to generate words. It should be noted that a convolutional neural network (CNN) layer is added, when the image information is embedded into the network. Each CNN extracts the feature from the image at each time instance. If the other information, such as audio and label information, is embedded, the CNN layer may be removed.

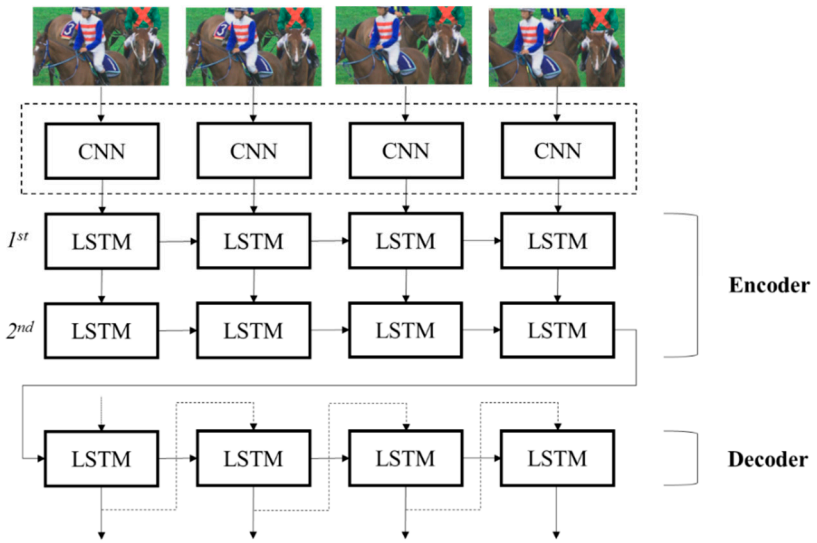

Figure 6. Proposed network when only one feature is available.

As displayed in Figure 2, MMVD embeds various information features into the single encoding layer. This makes it possible that some features are not fully encoded, but should be immediately ready to be decoded. Because each feature has different characteristics, it is better if they are embedded in the different encoding layers. Figure 7 illustrates the proposed network, when multimodal features are available. When the features are additionally embedded, the proposed network simply cascades the encoding layers. For example, if the number of multimodal features is two, one more encoding layer is added on top of the network encoder in Figure 6. If the number of features is three, four encoding layers are needed. If the number of features is $n-1$, n layers are required in the proposed network, as shown in Figure 7. Hence, the total number of cascaded encoding layers is proportional to the number of the multimodal features in the proposed network. However, the number of the decoding layers is one to maintain the computational complexity. The n^{th} encoding layer, which is the last layer in the encoder, plays a role that encodes all the embedded features once more, right before delivering them to the decoder.

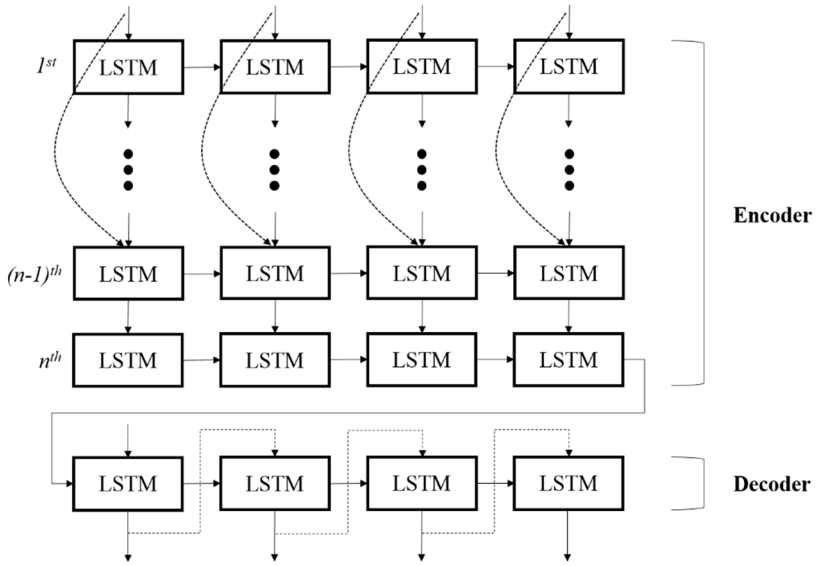

Figure 7. Proposed network when multimodal features are available.

4. Results

To analyze the impact of video compression and multimodal embedding on scene description, various datasets and information features were employed. First, the impact of the video compression on the scene description was analyzed with H.264/AVC [16], the S2VT network [3], image features extracted from a VGG model [22], and the Microsoft video description corpus (MSVD) dataset [23]. The MSVD dataset contains 1970 YouTube videos, which are split into a training dataset of 1200 videos, a validation dataset of 100 videos, and a testing dataset of 670 videos. Its average duration is about 9 s. Next, for the analysis of the multimodal embedding in the scene description, four different features, such as image, motion, audio, and label information, were used. For example, Inception-v4 (V4) was used for the image information feature [24], and the two-stream inflated 3D convolutional neural network (I3D) was employed as the motion information feature [25]. The mel-frequency cepstral coefficient (MFCC), which is derived from a type of cepstral representation of the sampled audio clips, was considered as the audio information feature [26]. The label feature was provided from the video metadata, and its feature was simply represented as a one-hot vector. In this experiment, the networks were trained on the Microsoft video-to-text dataset (MSR-VTT 2016) [27], which is a large-scale dataset consisting of 10,000 videos. It includes a training dataset of 6513 videos, a validation dataset of 497 videos, and a testing dataset of 2990 videos. Each video clip is annotated with 20 natural language descriptions, it is tagged with one of 20 labels, and its duration is between 10 and 30 s. In all experiments, initial values for the trainable weights were set to random values with uniform distribution between −0.1 and 0.1, and the Adam optimizer with a learning rate of 0.0001 was used to optimize the model. The size of the LSTM hidden state was set to 1024, and dropout regularization with a rate of 0.5 was employed. In order to evaluate the description performance of the proposed network, four different evaluation matrices, BLEU4 [28], METEOR [29], ROUGE-L [30], and CIDEr [31], were employed. For example, BLEU4 measures the word similarity between two sentences [28], whereas METEOR and LOUGE-L measure the semantic similarity [29] and sentence level structure similarity [30], respectively. CIDEr considers not only the sentence similarity but also grammaticality, saliency, importance, and accuracy [31].

Table 1 illustrates the description performance of the proposed network (Figure 5), when the original MSVD dataset and its low quality dataset were used for the network training. The low quality dataset with compression artifacts was generated with high compression. In a similar way, during inference, to generate a testing video with different quality from r_1 to r_n in Figure 5, video compression with various QPs was performed. Even though all possible QPs from 0 to 51 can be used in the

proposed network, only four different QPs, 0, 20, 40, and 50, were used in this experiment. Therefore, the r_1, r_2, r_3, and r_4 features were extracted from the reconstructed videos compressed with QPs of 0, 20, 40, and 50, respectively. As displayed in Table 1, the proposed network embedding the image features with various qualities improves the description performance in all scores, compared to the conventional S2VT network only considering the original quality.

Table 1. Description performance of the proposed network (Figure 5), when the original MSVD dataset and its low quality dataset were used for the network training.

Dataset Quality	Network	BLEU4	METEOR	ROUGE-L	CIDEr
Original	S2VT [3]	0.346	0.294	0.641	0.546
	Proposed	0.347	0.294	0.643	0.560
Low quality	S2VT [3]	0.297	0.268	0.619	0.413
	Proposed	0.310	0.273	0.629	0.456

Table 2 represents the description performance of the high (r_1 and r_2) and low (r_3 and r_4) quality features in the proposed method, respectively. When the original dataset was used for the training, the high quality features were much better than the low quality features. However, when the low quality dataset was used, the low quality features were more useful. It can also be observed that the description performance of the r_1 to r_4 features in Table 1 is exactly same as that of the high quality features, when the original dataset was used. When the low quality dataset was used, the performance of the low quality features was the same as that of the r_1 to r_4 features. This indicates that the improvement can only be achieved when the various quality of the reconstructed videos used during inference covers the quality of the dataset used for the network training. Sometimes, it is not known what kind of dataset is employed for the training. The dataset quality may be high enough, or the quality may be very low, because of the compression artifacts. For example, wireless multimedia sensor networks cannot guarantee that the video quality of the collected training dataset is sufficiently high, due to a limited bandwidth. Because the proposed network can utilize image information of various quality, it can achieve a relatively higher performance than the conventional network, regardless of the quality of the training dataset.

Table 2. Description performance of the high and low quality features in the proposed network, when the original MSVD dataset and its low quality dataset were used for the network training, respectively.

Dataset Quality	Features	BLEU4	METEOR	ROUGE-L	CIDEr
Original	r_1 and r_2	0.347	0.294	0.643	0.560
	r_3 and r_4	0.325	0.280	0.630	0.521
Low quality	r_1 and r_2	0.303	0.271	0.627	0.452
	r_3 and r_4	0.310	0.273	0.629	0.456

Table 3 illustrates the description performance of the proposed network (Figure 7), when the multimodal features of the image, motion, audio, and label information were used. The features were extracted from the original MSR-VTT 2016 dataset. The proposed network cascades the additional encoding layers according to the number of features. However, the single decoding layer is always used to reduce the computational complexity. In the result, *i*, *m*, *a*, and *l* stand for the image, motion, audio, and label information, respectively, and *ael* refers to the additional encoding layer to encode the features once more. If two features are used, *n* becomes three and they are embedded in the first and second encoding layers. If four features are all used, *n* becomes five and they are embedded in the first to fourth layers. As shown in Table 3, as additional features are added, the overall performance of the proposed network increases. For example, when the audio feature is embedded with the image feature, its performance is much better than when only the image feature is used. The performance can

be drastically boosted by incorporating the motion and label features. Based on these experiments, the highest description performance was obtained when MFCC for the audio, one-hot vector for the label, I3D for the motion, and V4 for the image were embedded in the first, second, third, and fourth layers, respectively. The last layer, which is the fifth layer, encodes these features once more, right before the decoding layer receives the encoded vectors.

Table 3. Description performance of the proposed network (Figure 7), according to the combination of the features, which are image (*i*), motion (*m*), audio (*a*), and label (*l*) features.

n	Layers					BLEU4	METEOR	ROUGE-L	CIDEr
	1	2	3	4	5				
1	*i*	-	-	-	-	0.367	0.271	0.588	0.430
2	*i*	*ael*	-	-	-	0.369	0.271	0.589	0.431
3	*a*	*i*	*ael*	-	-	0.380	0.273	0.597	0.439
4	*a*	*m*	*i*	*ael*	-	0.391	0.279	0.605	0.462
5	*a*	*l*	*i*	*ael*	-	0.392	0.279	0.605	0.463
5	*a*	*l*	*i*	*m*	*ael*	0.395	0.281	0.608	0.472
5	*a*	*m*	*l*	*i*	*ael*	0.397	0.282	0.604	0.474
5	*a*	*l*	*m*	*i*	*ael*	0.407	0.282	0.612	0.473

Table 4 compares the proposed network with MMVD, when the original MSR-VTT 2016 dataset was used. In the proposed network using the five encoding layers, based on Table 3, MFCC, label, I3D, and V4 features were embedded in the first, second, third, and fourth layers, respectively. In addition, a beam search method was applied to the proposed network to improve the quality of the generated descriptions. The main contribution of the beam search is to determine the most relevant translation from a set of possible candidate words. It can be observed that the proposed network cascading the multiple encoding layers outperformed MMVD using the single layer. As additional information, the proposed network was compared with CST [32]. CST indicates a consensus-based sequence-training network, which is the first network that practically applies a reinforcement learning (RL) algorithm to the scene description. RL employs the sentence score as a reward, and the network model tries to maximize this reward. As CST employs the CIDEr matrix to calculate this score, the corresponding score is drastically improved up to 0.542. To fairly compare the proposed network with CST, RL was applied to the proposed network. Similar to MMVD, CST concatenates the multimodal features, and then embeds them into the same encoding layer. The result shows that the proposed network is very competitive, in comparison with CST. For example, the scores of the proposed network and CST are similar, in terms of METEOR, ROUGE-L, and CIDEr, but the proposed network achieves much higher scores than CST in BLEU4.

Table 4. Description performance of MMVD [4], CST [31], and the proposed network.

RL	Network	BLEU4	METEOR	ROUGE-L	CIDEr
Not Applied	MMVD [4]	0.395	0.277	0.610	0.442
	Proposed	0.429	0.293	0.619	0.502
Applied	CST [32]	0.414	0.288	0.622	0.542
	Proposed	0.430	0.286	0.625	0.539

5. Conclusions

This paper investigated the impact of video compression and multimodal embedding on the scene description. The experimental results demonstrate that the description performance can be improved when the features extracted from the reconstructed images of various quality are used together during inference. It was also observed that multimodal embedding can be performed in the different encoding layers. The proposed network simply cascades the layers to efficiently perform

multimodal embedding, according to the number of the multimodal features. The results show that the proposed network is more efficient than the conventional networks.

Acknowledgments: This work was supported by Sejong University.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1998**, *9*, 1735–1780.
2. Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; Saenko, K. Translating videos to natural language using deep recurrent neural networks. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA, 31 May–5 June 2015; pp. 1494–1504.
3. Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrel, T.; Saenko, K. Sequence to Sequence—Video to Text. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015.
4. Ramanishka, V.; Das, A.; Park, D.H.; Venugopalan, S.; Hendricks, L.A.; Rohrbach, M.; Saenko, K. Multimodal video description. In Proceedings of the ACM Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 1092–1096.
5. Yang, Z.; Xu, Y.; Wang, H.; Wang, B.; Han, Y. Multirate Multimodal Video Captioning. In Proceedings of the ACM Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1877–1882.
6. Chen, S.; Chen, J.; Jin, Q.; Hauptmann, A. Video captioning with guidance of multimodal latent topics. In Proceedings of the ACM Multimedia, Mountain View, CA, USA, 23–27 October 2017.
7. Xu, J.; Yao, T.; Zhang, Y.; Mei, T. Learning Multimodal Attention LSTM Networks for Video Captioning. In Proceedings of the ACM Multimedia, Mountain View, CA, USA, 23–27 October 2017.
8. Jin, Q.; Chen, S.; Chen, J.; Hauptmann, A. Knowing Yourself: Improving Video Caption via In-depth Recap. In Proceedings of the ACM Multimedia, Mountain View, CA, USA, 23–27 October 2017.
9. Gunasekar, S.; Ghosh, J.; Bovik, A.C. Face detection on distorted images augmented by perceptual quality-aware features. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 2119–2131. [CrossRef]
10. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the devil in the details: Delving deep into convolutional net. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
11. Mash, R.; Borghetti, B.; Pecarina, J. Improved aircraft recognition for aerial refueling through data augmentation in convolutional neural networks. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 12–14 December 2016; Springer: Cham, Switzerland, 2016.
12. Taylor, L. Improving Deep Learning with Generic Data Augmentation. In Proceedings of the IEEE Symposium Series on Computational Intelligence, Bengaluru, India, 18–21 November 2018.
13. ISO/IEC 10918-1:1994—Information Technology—Digital compression and coding of continuous-tone still images–requirements and guidelines. 1994. [CrossRef]
14. ISO/IEC 15444-2:2004—Information Technology—JPEG 2000 Image Coding System: Extensions. 2009.
15. T.87: Information Technology—Lossless and Near-Lossless Compression of Continuous-Tone Still Images-Baseline. ISO-14495-1/ITU-T.87 (JPEG-LS). 2011.
16. ITU-T Rec. H.264 and ISO/IEC 14496-10. Advanced video coding for generic audiovisual services. 2003.
17. ITU-T Rec. H.265 and ISO/IEC 23008-2. High Efficiency Video Coding (HEVC). 2013.
18. Ortega, A.; Ranchandra, K. Rate-distortion methods for image and video compression. *IEEE Signal Process. Mag.* **1998**, *15*, 23–50. [CrossRef]
19. Wiegand, T.; Sullivan, G.J.; Bjntegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuit Syst. Video Technol.* **2003**, *13*, 560–576.
20. Sullivan, G.J.; Ohm, J.-R.; Han, W.-J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuit Systems Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
21. Ohm, J.-R.; Sullivan, G.J.; Schwarz, H.; Tan, T.K.; Wiegand, T. Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC). *IEEE Trans. Circuit Syst. Video Technol.* **2012**, *22*, 1669–1684.

22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
23. Chen, D.L.; Dolan, W.B. Collecting highly parallel data for paraphrase evaluation. In Proceedings of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011.
24. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
25. Carreira, J.; Zisserman, A. Quo vadis, action recognition? A new model and the kinetics dataset. *arXiv* **2018**, arXiv:1705.07750v3.
26. Davis, S.; Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [CrossRef]
27. Xu, J.; Mei, T.; Yao, Y.; Rui, Y. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5288–5296.
28. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
29. Denkowski, M.; Lavie, A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In Proceedings of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; pp. 376–380.
30. Lin, C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004; pp. 74–81.
31. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-based Image Description Evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4566–4575.
32. Phan, S.; Henter, G.E.; Miyao, T.; Satoh, S. Consensus-based sequence training for video captioning. *arXiv* **2017**, arXiv:1712.09532v1.