DVC-Net: A deep neural network model for dense video captioning

Sujin Lee | Incheol Kim

Department of Computer Science, Kyonggi University, Suwon-si, Gyeonggi-do, Korea

Correspondence
Incheol Kim, Department of Computer Science, Kyonggi University, 154-42, Gwanggyuosan-ro, Yeongdong-gu, Suwon-si, Gyeonggi-do, Korea.
Email: kic@kgu.ac.kr

Funding information
Ministry of Science and ICT, Grant/Award Number: 10.13039/501100014188

Abstract
Dense video captioning (DVC) detects multiple events in an input video and generates natural language sentences to describe each event. Previous studies predominantly used convolutional neural networks to extract visual features from videos but failed to employ high-level semantics to effectively explain video content such as people, objects, actions, and places, and utilized only limited context information in generating natural language. To overcome these deficiencies, DVC-Net is proposed, a new deep neural network model that uses high-level semantics to efficiently represent important events as well as visual features. Additionally, DVC-Net uses a bidirectional long short-term memory network, a type of recurrent neural network, to detect events over time. Furthermore, DVC-Net applies an attention mechanism and context gating to effectively exploit context information in a caption generation step. In experiments conducted versus state-of-the-art models, DVC-Net presented relative gains of over 1.72% (BLEU@1 score increases from 12.22 to 13.94) and 3.19% (CIDEr score increases from 12.61 to 15.80) on the large-scale benchmark datasets, namely ActivityNet Captions and MSR-VTT, respectively.

1 INTRODUCTION

The popularity of video sharing websites such as YouTube, Netflix, and Dailymotion have caused a rapid increase in the size of large-scale video datasets and has spurred interest in video content analysis. The development of 2D and 3D convolutional neural networks (CNNs) has further inspired research into classifying, captioning, and answering questions about images and short videos [1–8]. However, performing these tasks on long and untrimmed videos remains a key challenge in computer vision. Of these tasks, dense video captioning (DVC, demonstrated in Figure 1) seeks to detect multiple events in long and untrimmed videos and convey natural sentences to describe each event. DVC requires more advanced intelligence than existing video captioning tasks that generate one natural language sentence about a video containing only one event or action [9]. The ability to localize the time domain that contains events needed for DVC and the ability to describe the event in a video with natural language can have extensive application to systems such as navigation for the visually impaired, video searching, and automated video subtitling.

DVC takes a long and untrimmed video that contains more than one action or event as the input. Effective DVC therefore requires three subtasks: (1) feature extraction to locate features that effectively illustrate an untrimmed video, (2) temporal event localization to find time domains that contain events or actions, and (3) caption generation to generate natural language sentences to describe the detected temporal events.

Previous studies on DVC have predominantly employed C3D [10], a type of CNN, to extract visual features depicted in videos, and long short-term memory (LSTM) [9,11,12], gated recurrent units (GRUs) [13], and recurrent neural networks (RNNs) to locate temporal events in the videos. However, these studies have not used high-level semantics to effectively explain video content such as people, objects, actions, and places [9,11,13], and have utilized only limited context information in generating natural language [9,12].

For DVC, it is critical to extract features that can most effectively illustrate a given video. Previous research utilized visual features extracted using CNNs such as C3D. However, distinguishing between video events solely based on visual features is difficult; for example, whether the event displayed in a

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. IET Computer Vision published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.
video is a baseball game or a cricket match. Therefore, high-level features are crucial for directly describing the content of a given video. To overcome such limitations, a DVC model called DVC-Net is proposed, which utilizes both semantic and visual features. Semantic features indicate words that convey a video's attributes, such as actions, people, objects, and places. Specifically, the proposed DVC-net model uses both dynamic semantic features—actions—and static semantic features—objects, people, and places—simultaneously. Using dynamic and static semantics to depict input videos allows us to place the various components of an event in a video into attribute vectors. Moreover, the proposed DVC-net model applies soft attention to high-level semantic features, whereas existing models apply soft attention to low-level visual features. At the caption decoding stage, the DVC-net model determines which semantic features should be more focused on to generate the following words with this soft attention mechanism.

Additionally, locating temporal events in a video is necessary for DVC. Previous models have implemented LSTM and GRUs—types of RNNs—to localize the time domain of candidate events. The key to detecting the time domain of candidate events is the proper use of context information. For instance, when considering 'an event where ingredients are taken out' and 'an event where chopped ingredients are put into glass containers', it is easy to infer that 'an event where the ingredients are chopped' would occur between the two events. However, 'an event where ingredients are taken out' does not easily lead one to infer the next event. Many earlier studies did not consider future context information from videos and focused only on past context information to locate the time domain of candidate events. This limitation can be overcome by using bidirectional LSTM (BLSTM). When BLSTM is used, it is possible to effectively utilize future context information, which allows more accurate localization of events in time in combination with past context information.

Existing models are limited by their focus on critical features when generating captions; that is, these models cannot focus on features that are relevant to each caption sentence attribute. For instance, when generating a caption sentence saying, ‘A woman is standing in her kitchen in front of a counter’, to generate the word ‘woman’, the model should focus on a video domain where a woman appears. To generate the word ‘kitchen’, the model should focus on a video domain where a kitchen appears. Despite the fact that each word should be derived by focusing on different domains of a video, existing models focus equally on all attribute domains. Furthermore, caption sentences may include articles, prepositions, and conjunctions as well as the nouns and verbs that act as the subjects, objects, and predicates. Although nouns and verbs in a caption sentence are usually determined by video content, in many cases, it may be more effective for articles, prepositions, and conjunctions to be determined by the sentence context. However, existing models do not fully use context information for caption sentences. To address this problem, DVC-Net applies an attention mechanism and context gating to generate more natural caption sentences.

2 | RELATED WORK

2.1 | Video action detection

The development of deep learning technology for computer vision, and the comparative ease of large-scale dataset acquisition, has enabled researchers to address increasingly complex problems. From analyses of single images, the field has expanded to analyzing videos, which are composed of sequences of images. Early video analyses focused on video action recognition and categorization in short video clips [14,15]. Most research extracted global features and representations,
classifying actions based on the extracted features. Some research used C3D and Optical Flow, which are effective for describing motions to recognize video actions.

Video action detection, which detects events over time in untrimmed videos and automatically classifies actions in each event, is also well studied [16–20]. Initial research on video action detection used sliding window techniques and applied action classifiers to all sections indicated in the windows. However, the sliding window technique creates too many time sections and duplicates the processing of some video frames. Recently, to overcome the limitations of the sliding window method and increase action detection efficiency, researchers have proposed various methods for generating a limited number of candidate temporal events. The proposed methods include the use of dictionary learning [16] or RNNs [17,18] and application of action classifiers to each section that might contain actions after generating candidate time sections. However, these methods have limitations. For example, in the method proposed by Escorcia et al. [17], some video frames continue to require duplicate processing. Moreover, accurate determination of candidate temporal events should utilize content that comes before and after the candidate event; in the method proposed by Buch et al. [18], candidate time sections are generated by relying only on the given content.

2.2 Video captioning

Recently, video captioning—describing actions or events in a video with natural language—has gained growing interest. This topic is not as easy as it may first appear, because it requires natural language processing technology to generate descriptions and computer vision technology to understand video content. Initial research on video captioning used human-designed features and template-based caption generation. That is, these methods aimed to extract human-designed features, such as a histogram of oriented gradients or a scale-invariant feature transform, from a video and generate natural language sentences according to predefined grammatical rules and structures. However, this video captioning method required manual effort to construct features that would describe videos effectively. Further, this system could only generate natural language sentences according to predefined grammatical rules or structures. With developments in deep learning technology, multiple deep learning-based video captioning techniques have been proposed. Research using an encoder–decoder network structure—which was extremely successful in machine translation—has been especially prevalent in video captioning models. This encoder–decoder network structure treats video captioning as delivering a sequence of words that make up description sentences, which are created by entering a sequence of video frames [21–25]. In general, in the encoder–decoder network structure, an encoder extracts key features in a video and a decoder delivers a sequence of words that describe the video content using the extracted features. The encoder modules are typically implemented using CNNs like VGG [26], ResNet [27], and C3D, which are trained in advance; the decoder modules are implemented using RNNs like LSTM. Encoder–decoder network structure-based video captioning research has shown fair results. Nevertheless, such methods are limited by the equal processing of all video frames, even though some frames play more critical roles.

As mentioned earlier, in many previous studies, CNN models such as VGG, ResNet, and C3D were used to extract visual features in a video. Later, attempts were made to extract semantic features to more effectively describe video content [28–32]. Semantic features refer to words that indicate actions, objects, people, and places in a video. In these studies, the researchers believed that more accurate natural language sentences could be generated by using high-level semantics describing the video content or the attributes that made up certain scenes. Additionally, approaches for using RNNs to generate caption sentences were attempted. The first approach was to use semantic features as input to an RNN without processing. Another approach calculated the probability of semantic features and used them as internal parameters in an RNN [29]. One approach applied attention to semantic features and used it in an RNN [32], whereas another applied a word embedding method to semantic features and used that as RNN input [31]. However, none of these studies attempted to differentiate between static and dynamic semantic features.

In previous research, the decoder modules for generating captions were implemented using RNNs [24,25,28,33–37]; one of the most typical modules was LSTM. LSTM solves the vanishing gradient problem found in standard RNNs and can also have a longer-term memory of past data. Song et al. [36] employed two LSTM layers to implement a temporal attention mechanism to the visual features.

2.3 Dense video captioning

Previous video captioning techniques aimed to generate one natural language sentence to describe a video, whereas DVC seeks to generate multiple natural language sentences to depict multiple actions and events in a single video [9]. DVC has more complications than simple video captioning; it requires a subtask for detecting multiple events in a video and a subtask for generating natural language sentences that successfully describe each detected event. Hence, DVC combines video action detection and video captioning.

To effectively perform DVC, features that describe a video should be extracted first. Then, candidate temporal events should be detected, and natural language sentences should be generated to describe each event. Recent DVC methods predominantly use C3D to extract visual features, a GRU and an LSTM to generate candidate temporal events, and an RNN to generate natural language sentences to describe each event. The seminal DVC study [9] adopted an LSTM-based multiscale event detection module to effectively locate temporal events of different lengths based on C3D visual features extracted from the video. The researchers proposed an LSTM-based captioning module to utilize context information between
evens. However, this model did not perform well on the ActivityNet Captions benchmark dataset. Wang et al. [11] subsequently developed a model to solve a problem that arose in the previous model—specifically, the inability to distinguish different events that end at the same time—by fusing each event with hidden states of the event proposal module and visual features. Further, Xu et al. [13] proposed a two-level hierarchical captioning module to maximize the use of context information. In addition, Li et al. [12] developed a temporal event proposal module, which combines three submodules—an event/background classifier to predict events, a temporal boundary regressor to process the temporal boundaries of each proposal, and a descriptive regressor to infer the descriptive complexity of each event—for predefined video sections of different lengths. Zhou et al. [38] proposed an end-to-end transformer model for dense captioning. The model consists of three parts: a video encoder, a proposal decoder, and a captioning decoder. The captioning decoder contains a mask prediction network to generate text description from a given proposal. And Shen et al. [39] proposed a weakly supervised learning approach for dense video captioning. Park et al. [40] proposed an adversarial technique including a hybrid discriminator for dense video captioning. And Mun et al. [41] developed a novel dense video captioning framework, modeling temporal dependency across events in a video with an event sequence generation network. However, a common limitation found in these DVC methods is that they fail to fully use semantic features for each event.

3 | DEEP NEURAL NETWORK MODEL FOR DVC

3.1 | Model review

An overview of the DVC-Net model is given in Figure 2. First, descriptive features in untrimmed videos are extracted using a visual feature extraction network (VFEN). Next, using extracted visual features, candidate temporal events are detected using a temporal event proposal network (TEPN). Then, semantic features are generated for each detected temporal event using a semantic feature extraction network (SFEN). Finally, captions are generated by a caption generation network (CGN).

The event proposal network encodes visual features in a video using BLSTM and extracts candidate temporal events through fully connected layers at each time step. After this process, events that scored higher than the threshold value serve as the video’s temporal event set. Later, using a CGN, captions for each event are generated.

For even more effective generation of video captions, we propose caption generation using semantic features as well as visual features. Adding high-level semantics that describe objects, backgrounds, and actions to low-level visual features can better explain the video, aiding caption generation. In this study, we distinguish static semantic features that indicate objects, backgrounds, and people from dynamic semantic features that are related to actions, clarifying the role of each attribute and allowing more effective extraction of these features. Further, to more optimally utilize input attributes for caption generation, we use a BLSTM-based gate CGN. Natural language caption sentences are made with a sequence of words.
These words are not only those that can be discovered by watching the video—like nouns, verbs, and adjectives—but also those for context, including prepositions or articles. Considering characteristics found in natural language caption sentences, we use gates to generate captions. We further elaborate on each network in the following sections.

3.2 | Visual feature extraction network

In the proposed model, to extract visual features to describe videos, as seen in Figure 3, we use C3D, a 3D CNN that can extract spatiotemporal features. To extract visual features, a video is divided into 16 frames, consisting of eight frames overlapping with later clips. The C3D used in the model is Sport-1M, a pretrained network. The extracted visual features are subsequently used for temporal event proposal.

3.3 | Temporal event proposal network

The TEPN illustrated in Figure 4 is an expanded version of the single-stream temporal action proposal (SST) model [18]. SST models use a CNN to extract visual features from a video and encode them with LSTM to obtain candidate temporal events for that video. Using this approach, it is possible to obtain a temporal event dataset for a video with just one search attempt, saving considerable time and calculation efforts.

Unlike previous SSTs, we used BLSTM instead of LSTM, which allows our model to use future information as well as past information from a video to obtain temporal events. The structure of the TEPN used in this study is as follows: First, visual features are extracted using C3D. The extracted visual features are encoded using BLSTM, which produces status updates at the current time based on the visual features in a video over time. That is, at current time $t$, the value at the hidden state in the LSTM $h_t$ contains visual information in the video at $\{1, 2, \ldots, t\}$. Next, from $h_t$, an output from the LSTM, one can calculate confidence scores $C_i = c_i, i \in \{1, 2, \ldots, K\}$ through fully connected layers $K$, as given in Equation (1). $a_i$ is a candidate event that starts at $t - \delta_i$ and finishes at $t$, where $\delta_i, i \in \{1, 2, \ldots, K\}$ refers to a predefined length of a candidate temporal event. Thus, candidate events have the same end time $t$.

$$C_i = \text{sigmoid}(W_ih_i + b_i) \tag{1}$$

Subsequently, events whose confidence scores are greater than a threshold are selected for caption generation.

3.4 | Semantic feature extraction network

We used both high-level semantics and visual features to describe a video. Previous video caption models ignored high-level features. Soft attention methods have been employed to tackle this problem, but they still lead to gaps between the language information to be generated and actual descriptions; soft attention uses high-level information indirectly.

Therefore, in the proposed model, we used semantic features to directly describe a given video. By using semantic features, which consist of words that directly describe a video, the CGN can better understand the video.

To generate captions using semantic features, these features must first be identified from a video. Dynamic semantic features are difficult to identify with just one scene or frame, and
they are usually discovered by watching the video for some time. Unlike dynamic semantic features, static semantic features are objects, people, and places in one video scene, so watching one frame can detect these features. Owing to these differences, we differentiated dynamic semantic features from static semantic features.

Furthermore, using visual features appropriate for each of the dynamic and static attributes of semantics, each semantic feature is efficiently extracted; this is considered a multilabel classification problem.

The structure of the SFEN is given in Figure 5. Based on the attributes that each semantic feature carries, dynamic semantics can be extracted using visual features that describe spatiotemporal features and static semantics can be extracted using visual features that describe spatial features.

In dynamic semantic networks, to first utilize visual features that effectively describe temporal and spatial characteristics of a video, in every set of 16 frames (one clip), visual features are extracted from a pretrained C3D CNN, as given in Equation (2); here, \( v_1, \ldots, v_t \) indicates the frame at the ith clip of the given video, and \( \frac{n_v}{16} \) indicates the total number of clips when a video is divided into 16 frames.

\[
F^d_{cis} = C3D(v_1, \ldots, v_{\frac{n_v}{16}}), i \in \{0, 1, \ldots, \frac{n_v}{16}\} \tag{2}
\]

Subsequently, the extracted visual features are encoded using the LSTM RNN model in Equation (3), where \( F^d_{cis} \) refers to the visual features in one clip to be encoded at the current time and \( b_{t-1} \) refers to the hidden state of the LSTM.

\[
e = LSTM(F^d_{cis}, b_{t-1}) \tag{3}
\]

Next, from encoded visual feature \( e \), with a fully connected layer and the sigmoid function, the model identifies the probability distribution for dynamic semantic \( p^d \), as given in Equation (4), where \( W_d \) is equal to the training weight and \( b_d \) indicates bias.

\[
p_d = \text{sigmoid}(W_d \cdot e + b_d) \tag{4}
\]

In static semantic feature networks, to successfully use visual features that capture spatial information in a video, visual features are extracted from a pretrained ResNet CNN. As shown in Equation (5), a video will be divided into 16 frames; from the middle (eighth) frame, visual features \( F^s_{res} \) are extracted and then encoded using LSTM, as in Equation (6). Then, using the fully connected layer and sigmoid function, the probability distribution for static feature \( p_s \) is identified.

\[
F^s_{res} = \text{ResNet}(v^*_i), i \in \{0, 1, \ldots, \frac{n_v}{16}\} \tag{5}
\]

\[
e = LSTM(F^s_{res}, b_{t-1}) \tag{6}
\]

\[
p_s = \text{sigmoid}(W_s \cdot e + b_s) \tag{7}
\]

### 3.5 Caption generation framework

To generate captions using an RNN, input features should be effectively utilized, and contextual information should be considered to train sentence structures. When input features that illustrate a given video vary excessively, it becomes more difficult to determine which information is critical at the moment, even though an RNN can obtain a considerable amount of video information. Therefore, to effectively employ these features, we must identify those features that are critical. For this purpose, we developed an approach for applying soft attention.

Additionally, to generate smooth and fluent natural language sentences, it is important to identify which words are

---

**Figure 5** Semantic feature extraction network

**Figure 6** Caption generation network
being generated from the sentence context information. RNNs use input features and context information to create natural language sentences. In these sentences, the words that can be understood only from the video content (i.e., nouns, verbs, and adjectives) use input features, but the words intended to meet grammatical requirements (e.g., prepositions and articles) should be created using contextual information. Specifically, when generating words for sentence structures, content information is more important than input features. Therefore, we developed an approach using gates to prioritize input features or context information at a given time. Thus, soft attention was used in the proposed CGN (Figure 6) to input features or context information at a given time. Thus, verbs, and adjectives) use input features, but the words can be understood only from the video content (i.e., nouns, action, news/events/politics, education, TV shows, movies/comedy, animation, vehicles/autos, how-to, travel, science/technology, animals/pets, kids/family, documentary, food/drink, cooking, beauty/fashion, and advertisement. Each video included approximately 20,000 untrimmed YouTube videos with average lengths of approximately 120 s. Each video has, on average, 3.65 temporal events and captions. Each caption sentence comprises an average of 13.48 words, and the training, validation, and test datasets consist of 10,024, 4926, and 5044 videos, respectively. In this dataset, the ground-truths to the test set were kept confidential for a competition, so we used a validation set as our test set.

To learn the semantic feature network proposed, we used the ActivityNet Captions dataset to collect semantic feature data. The caption sentences were categorized into nouns and verbs using the part-of-speech tag functionality of the Natural Language Toolkit (NLTK). The plural forms of nouns and past tenses or present participles of verbs were converted into base forms using the lemmatizing functionality of NLTK. Then, among the reformed verbs, we chose 500 verbs with the highest frequencies to form dynamic semantic feature label data; of the reformed nouns, 1500 nouns with the highest frequencies were used as static semantic feature label data.

Additionally, to evaluate the performance of our CGN, we used the MSR-VTT dataset, which consists of 10,000 web video clips [42]. The MSR-VTT data were categorized into 20 sub-categories: music, people, gaming, sports/action, news/events/politics, education, TV shows, movies/comedy, animation, vehicles/autos, how-to, travel, science/technology, animals/pets, kids/family, documentary, food/drink, cooking, beauty/fashion, and advertisement. Each video included approximately 20 caption sentences, and training, validation, and test datasets consisted of 6513, 497, and 2990 videos, respectively.

4.2 | Model implementation

To run our experiments, the model was implemented in an Ubuntu 14.04 LTS environment using the TensorFlow Python deep learning library. The maximum length of generated captions was limited to 20 words, and it was assumed that they were captions generated before the <EOS> token. Model learning was not done in an end-to-end manner; after extracting the input features, they were used to learn the model. For the TEPN, Adam was used as the optimal algorithm and multilabel cross-entropy was used as a loss function, as given in Equation (14):

\[ L(c, t, X, y) = -\sum_{j=1}^{K} \left[ a_j g_j \log \hat{c}_j + a_j (1 - \hat{y}_j) \log \left(1 - \hat{c}_j \right) \right] \] (14)
For the semantic feature network, we used the optimal algorithm Adam and binary cross-entropy as a loss function for training, as given in Equation (15), where $Y$ refers to the actual ground truth and $\hat{y}$ indicates an expected value.

$$L_{\text{binary}} = -\left[y_n \log \hat{y}_n + (1 - y_n) \log \left(1 - \hat{y}_n\right)\right]$$  \hspace{1cm} (15)

The model optimization algorithm used for the CGN was RMSprop. Categorical cross-entropy was used as a loss function, as shown in Equation (16).

$$L_{\text{categorical}} = -\frac{1}{N} \sum_{n=1}^{N} \left[y_n \log \hat{y}_n + (1 - y_n) \log \left(1 - \hat{y}_n\right)\right]$$  \hspace{1cm} (16)

5 | EXPERIMENTS

5.1 | Performance comparison for various input features

Our first experiment was to identify the effects of input features on caption generation performance. To evaluate performance, the BLEU@N [43] and CIDEr-D [44]—usually used as metrics for evaluating caption generation—were used as indexes. Specifically, BLEU scores measure the proportion of common N-grams between the hypothesis and reference groups. CIDEr is an agreement-based evaluation protocol to reward sentences that are similar to human-generated ground truths. The code provided by Krishna et al. [9] was used to calculate the metrics. Tables 1 and 2 compare the caption generation performance based on input features. Table 1 presents the ActivityNet Captions dataset and Table 2 presents the MSR-VTT dataset. There were four comparison models: those only using visual features, using visual and dynamic semantic features, using visual features and static semantic feature, and using visual features and both semantic features.

Table 1 shows that for the models implemented on the ActivityNet Captions datasets, using semantic features improves performance. Specifically, compared to using just dynamic semantic features, using only static features showed better outcomes. Static semantic features better describe a video, as they comprehensively cover people, objects, and places, whereas dynamic semantic features only display actions. Furthermore, models that used both types of semantic features performed the best, which proves that the two types of semantic features independently contributed to caption generation.

In Table 2, which presents the models implemented for MSR-VTT, the results are similar to those in Table 1. The model that used both semantic features performed the best. For the MSR-VTT dataset, the static semantic features contributed more to the models compared with the ActivityNet Captions dataset, perhaps because the MSR-VTT dataset comprised more diverse videos with different categories.

### Table 1 Comparison of performance between features on the ActivityNet Captions dataset

| Feature                                      | BLEU@1 | BLEU@2 | BLEU@3 | BLEU@4 | CIDEr |
|----------------------------------------------|--------|--------|--------|--------|-------|
| Visual feature                               | 19.42  | 10.67  | 5.21   | 2.30   | 19.96 |
| Visual + dynamic semantic feature            | 21.57  | 11.26  | 5.73   | 2.78   | 22.14 |
| Visual + static semantic feature             | 26.34  | 13.71  | 6.37   | 3.04   | 24.92 |
| Visual + dynamic + static semantic feature   | 28.30  | 14.06  | 6.97   | 3.21   | 25.69 |

Note: To indicate the best performance, we used bold fonts.

### Table 2 Comparison of performance between features on the MSR-VTT dataset

| Feature                                      | BLEU@1 | BLEU@2 | BLEU@3 | BLEU@4 | CIDEr |
|----------------------------------------------|--------|--------|--------|--------|-------|
| Visual feature                               | 68.5   | 44.1   | 29.3   | 14.7   | 7.9   |
| Visual + dynamic semantic feature            | 68.2   | 43.6   | 28.6   | 16.8   | 16.9  |
| Visual + static semantic feature             | 70.7   | 48.6   | 32.8   | 21.6   | 34.1  |
| Visual + dynamic + static semantic feature   | 77.7   | 62.0   | 50.4   | 41.8   | 60.1  |

Note: To indicate the best performance, we used bold fonts.

### Table 3 Performance comparison of temporal event detection on the ActivityNet Captions dataset

| Method            | Precision@1000 | Recall@1000 | F1@1000 |
|-------------------|----------------|-------------|---------|
| Random            | 0.272          | 0.956       | 0.424   |
| DAPs [17]         | 0.367          | 0.774       | 0.498   |
| SST [18]          | 0.411          | 0.910       | 0.566   |
| TEPN (our method) | 0.459          | 0.875       | 0.602   |

Note: To indicate the best performance, we used bold fonts.

Abbreviations: SST, single-stream temporal action proposal; TEPN, temporal event proposal network.
5.2 Performance comparison using temporal event data

In a second experiment conducted to evaluate the performance of temporal event detection in DVC, we used the ActivityNet Captions dataset. We compared the performance of DAPs [17], SST [18], and randomly generated event start and end times. DAPs and SST extract visual features in a video and encode them with LSTM to detect temporal events. Because temporal events are detected within a CNN-LSTM structure in DAPs, the approach can be viewed as similar to SST. However, SST utilizes hidden states at every time step in LSTM to detect temporal events, whereas DAPs only utilize hidden states at the end of LSTM. Moreover, SST differs from TEPN, as SST uses LSTM whereas TEPN uses BLSTM. To evaluate performance, we calculated the average precision and recall rates of the top 1000 candidate temporal events when the threshold values of the temporal intersection of the union tIoU were 0.3, 0.5, 0.7, and 0.9. Moreover, using F1 scores, evaluation scores considering the precision and recall rate were calculated. tIoU refers to the minimum degree to which the actual time event and predicted time were aligned and it can be calculated using Equation (17). All metrics were calculated using the code provided by Krishna et al. [9].

\[
tIoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{(A_g \cap A_p)}{(A_g \cup A_p)} \quad (17)
\]

The results in Table 3 indicate that the TEPN has the best accuracy and F1 scores. However, the random generation method had the best recall rates because the lengths of the ground truths from the ActivityNet Captions dataset were relatively long. However, the random generation method has a very low accuracy and low confidence in the generated events, lowering its overall performance. Therefore, to evaluate the performance of generated events, F1 scores—which consider both metrics—should be used.

5.3 Performance comparison with previous models

In our third experiment, we evaluated the temporal event detection and caption generation performance of DVC-Net. To analyze both subtasks, we used the measure proposed by Krishna et al. [9]. This measure calculates the average captioning accuracy of the top 1000 candidate temporal events when the tIoU values are equal to 0.3, 0.5, and 0.7.

Table 4 presents the results of DVC-Net and previous models—developed by Krishna et al. [9], Wang et al. [11], Li et al. [12], and Zhou et al. [38]—using the ActivityNet Captions dataset. The model proposed by Krishna et al. [9] uses visual features encoded with LSTM when generating captions. The model proposed by Li et al. [12] extracts visual features with C3D and applies mean pooling. The model proposed differs from these models in that it uses semantic as well as visual features for video captioning. The model developed by Wang et al. [11] uses visual features encoded through LSTM and applies context gating in generating captions, whereas the model provided by Zhou et al. [38] uses masking network and applies self-attention.

According to Table 4, the proposed model is the best among all those compared, suggesting that the use of semantic features, BLSTM-based temporal event detection, and the

| Model       | BLEU@1 | BLEU@2 | BLEU@3 | BLEU@4 | CIDEr |
|-------------|--------|--------|--------|--------|-------|
| Krishna et al. [9] | 10.81  | 4.57   | 1.90   | 0.71   | 12.43 |
| Wang et al. [11]  | 10.75  | 5.06   | 2.55   | 1.31   | 7.99  |
| Zhou et al. [38]  | 9.96   | 4.81   | 2.42   | 1.15   | 9.25  |
| Li et al. [12]    | 12.22  | 5.72   | 2.27   | 0.73   | 12.61 |
| DVC-net           | 13.94  | 5.67   | 2.52   | 1.26   | 15.80 |

Note: To indicate the best performance, we used bold fonts.

| Model       | BLEU@4 |
|-------------|--------|
| MP-LSTM(V)  [21]  | 34.8   |
| MP-LSTM(C) [21]  | 35.4   |
| MP-LSTM(V+C) [21] | 35.8   |
| SA(V) [23]     | 35.6   |
| SA(C) [23]     | 36.1   |
| SA(V + C) [23] | 36.6   |
| hLSTMt [36]   | 37.4   |
| hLSTMt [36]   | 38.3   |
| Weakly supervised [39] | 41.4   |
| StATS(R152) [45] | 39.2   |
| StATS(R152+C3D) [45] | 41.6   |
| StATS(I3D) [45] | 40.1   |
| StATS(I3D + FL) [45] | 41.0   |
| StATS(I3D + FL + C) [45] | 41.7   |
| StATS(I3D + FL + C) [45] | 41.8   |
| DVC-Net       | 41.8   |

Note: To indicate the best performance, we used bold fonts.
attention gate CGN proposed effectively contributes to DVC. In the BLEU@N metric table, performance decreased by 50% as \( N \) increased because the BLEU metric table compares sentences with the ground truths by \( N \)-gram unit.

Table 5 compares the performance of DVC-Net with previous models using the MSR-VTT dataset; in this table, MP-LSTM is the model introduced by Venugopalan et al. [21], SA is the model proposed by Yao et al. [23], hLSTMat is the
model proposed by Song et al. [36], Weakly Supervised is the model by Shen et al. [39], and STaTS is the model proposed by Cherian et al. [45]. (V) indicates cases where VGG was used as the CNN model for video encoding, (C) indicates cases in which C3D was used, and (V + C) indicates cases where both were used. R152, C3D indicates cases in which ResNet152 and C3D was used, respectively, and I3D and FL stand for the I3D RGB and optical flow models, respectively, while C stands for using the class annotations supplied with the dataset during training. The results given in Table 5 prove that the proposed model outperforms all the comparison models with a BLEU@4 score of 41.8%. These findings suggest that caption generation is excellent when using semantic feature networks.

5.4 | Qualitative analysis

Next, to validate whether the proposed model effectively performs DVC, we conducted a qualitative experiment using the ActivityNet Captions dataset (see Figure 7). Figure 7a demonstrates that the proposed model properly extracts semantic features and generates relevant captions using extracted features. Moreover, because of the BLSTM, as temporal events were proposed using sufficient context information, these events were better classified and more accurate than the ground truths.

The second caption in Figure 7b shows an error occurring owing to incorrect semantic feature extraction, where a semantic feature was extracted as ‘white’. However, because of this semantic feature (the verb ‘miss’), predicted proposals managed to better describe the situation in the video than the ground truth proposal. Based on these findings, we can conclude that the model proposed helps to effectively generate captions; however, if the extracted features are inaccurate, these incorrect extractions can impact caption generation performance. The second and third sentences of Figure 7c show that words and phrases are used repetitively (‘he’/’man’, ‘along the water’/’in the water’). In the second caption, a conjunction (‘then’) is added. Such results show that the proposed model is still limited in terms of creating natural sentences, although it generates nouns and verbs readily owing to semantic features.

6 | CONCLUSION

A deep neural network for effective DVC, called DVC-Net, is proposed, which adopts both semantic and visual features. Furthermore, the model uses BLSTM, instead of unidirectional LSTM, to fully exploit context information when detecting temporal events. An attention mechanism and context gating methods are also applied to generate more natural caption sentences. Using the large-scale benchmark datasets, namely ActivityNet Captions and MSR-VTT in a series of experiments, the proposed model was shown to perform well in generating video captions.

However, the proposed model became less accurate as the descriptive complexity of its sentences increased. To overcome this limitation, in future research, we plan to implement more refined temporal event mechanisms, which should improve video captioning, and explore different approaches for generating more natural sentences.

ACKNOWLEDGEMENTS

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-0-01642) supervised by the IITP (Institute for Information & Communications Technology Promotion).

REFERENCES

1. Ogawa, T., et al.: Favorite video classification based on multimodal bidirectional LSTM. IEEE Access. 6, 61401–61409 (2018)
2. Chadha, A., Abbas, A., Andreopoulos, Y.: Video classification with CNNs: using the codec as a spatio-temporal activity sensor. IEEE Trans. Circuits Syst. Video Technol. 29(2), 475–485 (2019)
3. Xue, H., Zhao, Z., Cai, D.: Unifying the video and question attentions for open-ended video question answering. IEEE Trans. Image Process. 26(12), 5656–5666 (2017)
4. Xue, H., et al.: A better way to attend: attention with trees for video question answering. IEEE Trans. Image Process. 27(11), 5563–5574 (2018)
5. Gao, L., et al.: Video captioning with attention-based LSTM and semantic consistency. IEEE Trans. Multimed. 19(9), 2045–2055 (2017)
6. Long, X., et al.: Attention Clusters: purely attention based local feature integration for video classification. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 7834–7843 (2018)
7. Long, X., et al.: Multimodal keyless attention fusion for video classification. Proc. AAAI Conference on Artificial Intelligence, pp. 7202–7209 (2018)
8. Gan, C., et al.: DevNet: a deep event network for multimedia event detection and evidence recounting. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 2568–2577 (2015)
9. Krishna, R., et al.: Dense-captioning events in videos. Proc. IEEE Int. Conf. Comput.Vis. (ICCV), pp. 706–715 (2017)
10. Tran, D., et al.: Learning spatiotemporal features with 3D convolutional networks. Proc. IEEE Int. Conf. Comput.Vis. (ICCV), pp. 4489–4497 (2015)
11. Wang, J., et al.: Bidirectional attention fusion with context gating for dense captioning. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 7190–7198 (2018)
12. Li, Y., et al.: Jointly localizing and describing events for dense video captioning. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 7492–7500 (2018)
13. Xu, H., et al.: Joint event detection and description in continuous video streams. arXiv preprint. arXiv:1802.10250 (2018)
14. Wang, X., et al.: Beyond frame-level CNN: saliency-aware 3D CNN with LSTM for video action recognition. IEEE Signal Process Lett. 24(4), 510–514 (2017)
15. Hou, J., et al.: Content–attention representation by factorized action–scene network for action recognition. IEEE Trans. Multimed. 20(6), 1537–1547 (2018)
16. Heilbron, FC., Nièbles, JC., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1914–1923 (2016)
17. Escorea, V., et al.: DAPs: ‘Deep action proposals for action understanding’. In: Liebe, B., et al. (eds.) Proc. European Conf. Comput. Vis. (ECCV), pp. 768–784. Berlin, Germany: Springer. (2016)
18. Buch, S., et al.: SST: single-stream temporal action proposals. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 6373–6382 (2017)
19. Wu, Y., et al.: Temporal action detection based on action temporal semantic continuity. IEEE Access, 6, 31677–31684 (2018)
20. Murtaza, F., et al.: End-to-end temporal action detection using bag of discriminant snippets. IEEE Signal Process. Lett. 26(2), 272–276 (2019)
21. Venugopalan, S., et al.: Sequence to sequence: video to text. Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 4534–4542 (2015)
22. Pan, Y., et al.: Jointly modeling embedding and translation to bridge video and language. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4594–4602 (2016)
23. Yao, Y., et al.: Describing videos by exploiting temporal structure. Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 4507–4515 (2015)
24. Pei, W., et al.: Memory-attended recurrent network for video captioning. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 8347–8356 (2019)
25. Gan, X., et al.: StyleNet: Generating attractive visual captions with styles. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 3137–3146 (2017)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Proc. Int. Conf. Learning Representations (ICLR) (2015)
27. He, K., et al.: Deep residual learning for image recognition. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 770–778 (2016)
28. Long, X., Gan, C., Melo, G.: Joint event detection and description in continuous video streams. Transactions of the Association for Computational Linguistics (TACL), pp. 173–184 (2018)
29. Gan, Z., et al.: Semantic compositional networks for visual captioning. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1141–1150 (2017)
30. Pan, Y., et al.: Video captioning with transferred semantic attributes. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 6504–6512 (2017)
31. Yu, Y., et al.: End-to-end concept word detection for video captioning, retrieval, and question answering. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 3165–3173 (2017)
32. Nian, F., et al.: Learning explicit video attributes from mid-level representation for video captioning. Comput. Vis. Image Underst. 163, 126–138 (2017)
33. Liu, A., et al.: A fine-grained spatial–temporal attention model for video captioning. IEEE Access, 6, 68463–68471 (2018)
34. Yang, Y., et al.: Video captioning by adversarial LSTM. IEEE Trans. Image Process. 27(11), 5600–5611 (2018)
35. Bin, Y., et al.: Describing video with attention-based bidirectional LSTM. IEEE Trans. Cybern. 49(7), 2631–2641 (2019)
36. Song, J., et al.: Hierarchical LSTM with adjusted temporal attention for video captioning. Proc. Twenty-Sixth Int. Joint Conf. Artificial Intelligence (IJCAI), Melbourne, Australia, pp. 2737–2743 (2017)
37. Liu, A., et al.: Hierarchical & multimodal video captioning: Discovering and transferring multimodal knowledge for vision to language. Comput. Vis. Image Underst. 163, 113–125 (2017)
38. Zhou, L., et al.: End-to-end dense video captioning with masked transformer. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 8739–8748 (2018)
39. Shen, Z., et al.: Weakly supervised dense video captioning. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1916–1924 (2017)
40. Park, J., et al.: Adversarial inference for multi-Sentence video description. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 6591–6601 (2019)
41. Mun, J., et al.: Streamlined dense video captioning. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 6581–6590 (2019)
42. Xu, J., et al.: MSR-VTT: A large video description dataset for bridging video and language. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 5288–5296 (2016)
43. Papineni, K., et al.: BLEU: A method for automatic evaluation of machine translation. Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, USA, pp. 311–318 (2002)
44. Vedantam, R., Zitnick, C.L., Parikh, D.: CIDEr: Consensus-based image description evaluation. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4566–4575 (2015)
45. Cherian, A., et al.: Spatio-temporal ranked-attention networks for video captioning. Proc. IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1617–1626 (2020)

How to cite this article: Lee S, Kim I. DVC-Net: A deep neural network model for dense video captioning. IET Comput. Vis. 2021;15:12–23. https://doi.org/10.1049/cvi2.12013