SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING

VOLUME 7, ISSUE 4
ISSN 2054-7412

A Framework: Region-Frame-Attention-Compact Bilinear Pooling Layer Based S2VT For Video Description

[1] **Haifeng Sang**, [2] **Ge Hai**
[1,2] College of Information Science and Engineering, Shenyang University of Technology, Shenyang, China;
jianglia_0119@qq.com

ABSTRACT

In the video description task, the temporal information and visual information of the video are very important for video understanding, and high-level semantic information contained in mixed features of text features and video features plays an important role in the generation of video caption.In order to generate accurate and appropriate video captions.Based on the S2VT (sequence to sequence: video to text)framework, we propose a video description neural network framework (RFAC-S2VT) with a two-level focus and compact linear pooling layer (CBP) fusion.We use visual information and category information from the dataset for class training, and then we use CNN to extract the trained visual features.In the encodering stage,this paper designs a regional attention mechanism to dynamically focus on each frame of video,and then the region-weighted 2D visual features and C3D visual features containing temporal information are then fused together. We use the characteristic of model to model the fusion visual features with temporal information.In the decodering stage, this paper designs a frame-level attention ,and then fine-grained the video features which has been focusd by frame-level attention and the text features in the dataset by using compact linear pooling layer (CBP),finally model generated relevant video caption.We validate the proposed network framework on the MSR-VTT dataset,the results show that our proposed neural network framework is competitive on this dataset and current state of the art.

Keywords: Video description,Region attention,Frame level attention,Fine-grained fusion.

1 Introduction

The video description is an automatic textual description of the video in natural language. The research[1] direction which is attributed to the field of video understanding is a comprehensive manifestation of computer vision and natural language processing, and it has always been regarded as one of the biggest challenges in the field of video understanding. Recently, video description technology has received more and more attention. It helps the blind to understand the content of the video, human-computer interaction and improve the quality of online video search.

A video description is the task of converting a sequence of video frames into a sequence of words, a sequence-to-sequence task. Due to the popularity of deep learning and the development of deep neural networks, the results[2] that people expect to see have been achieved.In the part of natural language processing,people are affected by machine translation, the Encoder-Decoder framework which is widely used in the field of machine translation was successfully applied to video

DOI: 10.14738/aivp.74.6717
Publication Date: 19st,August 2019
URL: http://dx.doi.org/10.14738/aivp.74.6717

descriptions.And it also achieved good results.First, the video feature is encoded into a fixed size vector that is used as the input to the RNN. The RNN is then used as a decoder to generate sentences.

In particular for the input features of the decoding stage's RNN,the input feature is very relevant for the generation of a correct video caption. Early adopting the method[3] of mean pooling to deal with the input feature to the RNN.Recently, researchers have found that the video contains a lot of irrelevant and redundant content which can influence on the accuracy of the video description.Especially each time a word is generated, the video feature is the same for different words, which is not consistent with the logic of the sentence. Therefore, [4]proposes to use the temporal attention to generate different video feature vectors for each word, and low-level attention to video frames that are unrelated to words at the current time, focusing on video frames associated with the target video caption.Later, people found that each frame of the video also has a irrelevant background that was unrelated to the word, so[5] and [6] proposed a region attention to foucs on the most relevant area of the generated words at each moment.

The input feature types of the decoding part's RNN are also different,[7]only used text features as the input of the decoder RNN,then the visual features is the input of RNN 's hidden state. This approach does not allow for deeper interactive text and visual features, so the input of the RNN of the decoding part is concatenate features of text features and visual features, which can interactive two types of features.

In the process of video description, it is also important to use the category information of the video in the dataset, [8]combine category features and visual features according to specific weight assignments.

Inspired by the above work, we designed a video description neural network framework (CRFAC-S2VT) which combines two-level attention and compact linear pooling layer (CBP)[9] layer based on the S2VT framework.First, we use the category labels from the dataset and the video features extracted by the Convolutional Neural Network (CNN) to train the feature extractor, which can make the visual features extracted by the feature extractor contain class characteristics.Inspired by [], we combine the regional attention mechanism with the S2VT neural network framework.Using the region attention to focus on each frame of the video, focusing only on the region of each frame which is related to the target caption. Then, the 2D features which has been weighted by the region attention and the C3D features extracted by the 3D convolutional neural network are fused, and the fusion feature is used as input to the encoder of S2VT . Such an encoding method can obtain higher-level semantic information and temporal information of the fusion feature, and then he encoded features are integrated into a video vector.The frame-level attention selects the video frame vector that is most relevant to the word at the current moment, reducing the attention of the irrelevant frame, and generating different video vectors when the word is generated at each moment.In the inspire of[10],we combine video vectors and text features of words by Compact Bilinear Pooling (CBP)Layer, which can fine-grained fusion of video and text features and make two types of features interact deeply.Finally the fused feature vector is input to the LSTM of the decoding stage to generate video caption.

To the best of our knowledge, this is the first time that S2VT, a two-level attention , and a compact linear pooling (CBP) layer have been combined. Such a combined framework can comprehensively deal with visual information, category information, and text information, and excavate higher-level semantic information to produce a more appropriate video description.The framework we propose is validated on the most popular datasets, Microsoft Research Video-to Text (MSR-VTT).The experimental results show that our proposed neural network framework shows the most advanced performance in the evaluation criteria.

2 Related Works

Early methods[1, 11-12] of video description were divided into two stages. The first was to identify the semantic content of the video which include the subject, the verb, the object, and then generate a sentence based on the template.These work usually require training a single classifier to identify candidate objects, actions, and scenes, and then use a probability map model which combine with visual information and language models to determine the most likely content in the video,finally the work can be generated a video caption.Although the method divides content extraction and sentence into simplified separations, it needs to select a set of related objects and actions, and the method that is template-based sentence generation is not sufficient to simulate the richness of human description language.

Recently,due to the development of deep learning, video caption have made great progress.The most popular method currently is the framework structure based on the CNN-RNN framework, where the CNN[13-15] is used to extract feature of video frames , that is the coding part; and the RNN is used to generate video caption, that is the decoding part.The CNN-RNN-based framework was first proposed by [3],which averages pooling simply each video frames feature extracted by CNN, and then the pooled visual features is used as the input of long and short time memories(LSTM). The network (LSTM)'s output is a video description.The main shortcoming of the method is that it cannot capture the temporal information of the video,so the CNN-RNN-based framework is only suitable for short video clips.In order to excavate the temporal information of the video's frame sequence, [7] proposed the S2VT framework. Firstly, the framework also uses the Convolutional Neural Network (CNN) to extract the features of each video's frame , and then the features of each video's frame is input to the LSTM. LSTM encodes the video's frame squence to get the temporal information, and then inputs the feature vector of each video's frame which has been encoded into the decoded stage's LSTM to generate a subtitle of the video caption.The main shortcoming of the framework is that the video feature vectors of the generated word at each moment are the same when the video caption are generated ,that is, there is no attention to the temporal information of the video's frame sequence.And there is no content screening for each video's each frame, because the content of each frame contains redundant visual content, it will interfere with the generation of related caption.In inspired of attention mechanism[16] successfully applied to the field of image caption, [17] proposed the use of frame-level attention in the field of video description.The model can focus on the video frames most relevant to the word at that moment based on the different words generated each time.Later [5-6] proposed a region attention , which can automatically focus on the important area of the video's each frame most relevant to the caption according to the difference characteristics of each frame.The proposed framework[18]is also improved on the basis of S2VT, It performs two encodings and two decodings according to different words generated each time, and finally filters the generated more accurate words according to two loss functions.However,the main shortcoming of the framework is applying just a single way of using vector concatenate when a word and video feature vector are combined,and it does not take into account the use of higher-level semantic feature fusion for video feature vectors and word feature vectors.The application of different kinds of features in the dataset is also very extensive. The framework proposed by [8] uses the audio, video and category information in the dataset, .which assigns weights to these different categories of information and then adds them to the fusion feature.The neural network framework designed in this paper solves all the above shortcomings .

3 Proposed Model

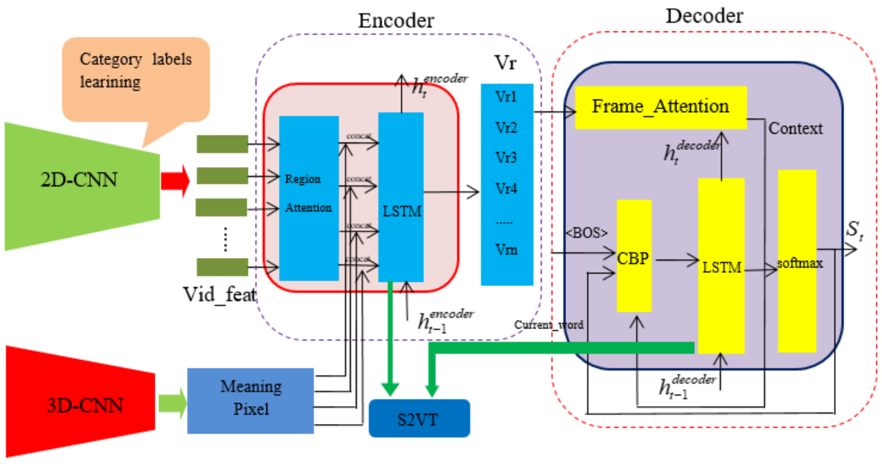


Figure 1:The Architecture Of RFAC-S2VT Framework

The framework we propose can be divided into three stages as shown in the Figure 1:the first stage is feature pre-processing and pre-training, which includes video processing into pictures, feature extraction of each frame of video and category pre-training of video features, and processing of text features;the second stage is the coding stage, which includes the fusion of the 2D features of the video frame and the 3D features of the video frame, the region attention, and the encoding of the fused video frame features by using the LSTM[19];the third stage is the decoding stage, which includes attention to useful frames of video frame features, fine-grained fusion of video features and text features, and text generation for video descriptions using decoded LSTM.For the sake of clarity, we will describe the combination framework in detail in the above three stages.

3.1 The Stage Of Video Feature Preprocessing And Pre-training

In this stage, we use the Resnet50 based on Convolutional Neural Network (CNN) to extract features from the video, and then classify the video frame features by using the category information in the dataset as a label.However, the dataset is video-level data, it cannot be directly applied to the neural network, so we need to divided the video into a picture database consisting of each frame of video.First, we divide the video into several frames at a rate of 30 frames per second, and the size of each frame is cropped to a size of 224×224. The reduced picture is subjected to regularization (normalization) processing, and the regularized parameters include the mean (R=0.485, G=0.456, B=0.406), and the variance (R=0.229, G=0.224, B=0.225).At the same time, in order to prevent the singularity of the data during training lead to that the neural network model is over-fitting, and we adopt the random flip processing method to increase the data diversity of the image.After the above processing, the pictures are input into the Resnet50 neural network to get the feature vector corresponding to the video.Then, the category information in the dataset is used as a label, and the video feature extractor is trained.The number of known categories is 20, and the training times is set to 500 epochs, which ensures that the value of the loss function is reduced to around 0. During training, we use a cross entropy loss function.

After the training is completed, we use the weight of the trained Resnet50 to extract the features of video's frame. When extracting features of video's frame , 50 frames are extracted from each video frame library at regular intervals and these sample frames are used as input to the feature extraction network. Then, the mean pooling layer of Resnet_50 and all layers after the pooling layeris are deleted , and the output of the feature extractor is used as input which is features of video's frame to our proposed framework.Vn={f_1,f_2,f_3,...,f_i} is a representation of the features of video ,where i=50, n represents the nth video sample.Since the framework we propose is a non-end-to-end training method, we need to save the extracted video frame features in a file in numpy format, which can be called directly during training.

Since video data is time-series, it contains a wealth of temporal information,in the above work the 2D feature of video's frame extracted by the convolutional neural network only contains the spatial information of the video content and the category information included after the pre-training of the category.Therefore, we are ready to fuse C3D features of video and 2D features of video's frame with temporal information, and the C3D features of video are extracted by 3D-CNN which has been trained on the Kinectics dataset.We also put it in a file in numpy format so that it can be called directly during training.

3.2 Processing Of Text Feature

We make a dictionary of words based on all video caption in the dataset.When the model is trained, we use the embeddings layer to extract the feature vectors of the words in the dictionary. All words in the caption are then made into words to ndexed, indexed to words, including starter <BOS> and terminator <EOS>, according to all video the label of caption.Since each word of the video caption passed in the model is represented by a vector, so indexing to words can transform the output of the model into words that we can understand.We will also all the words in subtitles into lowercase, remove the rare word and is represented by <UNK>.In order to make the decoding model easy to train and to adapt to different lengths of video caption, we set the length of each training caption to 28.If the number of words in the sentence is not enough 28, we do zero padding in other places. If the number of words in the sentence is greater than 28, then only the first 28 words are retained.

3.3 The Stage Of Encoding

3.3.1 Region Attention

In this stage, we will input the trained features of video into the coding part of our proposed framework, ie the input feature is $V_n=\{f_1,f_2,f_3,...,f_m\}$.The following is a detailed introduction to the coding process and the individual components.

Input video features $V_n=\{f_1,f_2,f_3,...,f_m\}$ to the encoding stage, and V_n represents the feature combination vector of C3D features of video and 2D features of video's frame. The feature vector of each frame of picture $f_i \in R^{h\times w\times d}$, $i\in\{1, 2,3,...,m\}$. Thus, for a certain frame i, we get a set of feature vector, $\{r_{i1}, r_{i2}, . . . , r_{im}\}$, where m denotes the number of regions.The feature combination vector of video is used as the input of the region attention (RA), that is, every region of each frame of the video is weighted and then summed.

$$f_i = \sum_{i=1}^{n} \alpha_{ij} r_{ij} \tag{1}$$

Where α_{ij} represents the scoring weight of the j-th region feature of the i-th frame feature of the video.The calculation of α_{ij} is the following formula:

$$n_{ij} = \omega^r \tanh\left(W_1^r r_{i,j} + W_2^r \alpha_{i-1,j} + b^r\right) \tag{2}$$

$$\alpha_{ij} = exp\{n_{ij}\} / \sum_{j=1}^{k} exp\{n_{ij}\} \tag{3}$$

Where W_1 and W_2 are optimized by training, and the obtained   emphasizes the most prominent area in each frame of the video, which can reduce the influence of redundant and irrelevant area features in the image.The area feature weight of the i-th frame is not only determined by the characteristics of the current frame, but also by the region attention weight score of the previous frame picture.The detailed structure diagram of the convolution attention mechanism is Figure 2.

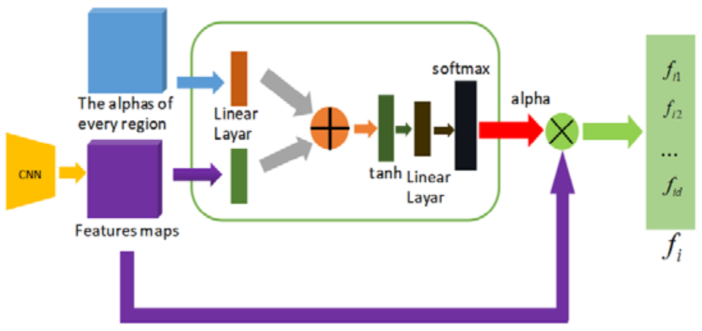


Figure 2:The Architecture Of Region Attention

3.3.2 Fusion Of 2D Visual Features And 3D Visual Features

Each region of the picture after the attention of the convolution area has been given different weights.However,the feature vector with the dimension of 7×7×2048 also reduces the dimension due to the weighted summation. $fi_{(2D)} \in R^D$ is the feature vector of each frame of the video weighted by the region attention mechanism, and it is linear vector whose the dimension is 2048.We take pre-fetched 3D video feature vectors for the mean of each pixel, so that each 3D video feature vector becomes a linear vector with a dimension of 2048.The specific details are based on the following formula:

$$V_{n(3D)} = \{f_1, f_2, \ldots, f_s\}$$
$$f_{mean} = \frac{\sum_1^s f_i}{s} \tag{4}$$
$$V_{n(3D)} \in R^{s \times D}, \quad f_i \in R^D$$

Where s is the length of the video sequence, and D is the dimension of the vector. ie D = 2048.$\sum_1^s f_i$ indicates that the pixel values of each column of $V_{n(3D)}$ are added, and f_{mean} represents the meaning feature after the mean of the sum of the pixel values of each column, ie $f_{mean} \in R^D$.

Then we perform the end-to-end splicing operation of the mean feature f_{mean} and the 2D feature vector $f_{i(2D)}$ of each frame of the video, and we get the hybrid feature is $V_k \in R^{b \times s \times 2D}$, where b is batch_size.The details are as follows:

$$V_k = F(f_{mean}, f_{i(2D)})_i \tag{5}$$

F is a vector concatenate operation.

3.3.3 LSTM For Encoding

In the above work, we obtained a mixed feature vector V_K containing spatial information and time information, which contains high-level semantic information about the video. We use the LSTM of the encoded part to encode the mixed video feature vector. The included high-level semantic information is mined.We input the mixed video feature vector V_K into the LSTM of the encoding part by per frame of the video, retain the output of each frame, and then only retain the hidden state hidden of the last frame output. Finally integrate each frame's output into a video feature vector $V_T \in R^{b \times s \times d}$,where b and s are the same size as V_K, and the dimension d is different, depending on the number of nodes of LSTM and the fully connected layer.The specific details are as follows:

$$\text{Encoder\_output, Encoder\_hidden} = \text{encoder\_LSTM(Encoder\_input, Encoder\_hidden)} \tag{6}$$

$$V_T = \text{concat}(\text{Encoder\_output})_i \tag{7}$$

Where the Encoder_input is the mixed feature vector V_{ki} for each frame.

3.4 The Stage Of Decoding

3.4.1 Frame-Level Attention

Since the description of the video is composed of almost every different word, and the meaning of each word is different, then when we generate the text, the video feature vectors corresponding to each word cannot be the same, which is not reasonable.At this time, we need to use the frame-level attention mechanism. Since it can filter the video frames related to words, the weight added to the video frames with high relevance to the word at the current moment will increase, and the weight of the unrelated video frames will be reduced, which reduces the use of video frames and makes full use of the video frames associated with the caption, so it will generate more accurate caption.The detailed frame-level attention mechanism is calculated as follows

$$h_\alpha^t = tanh(W_3 h^t + b_1) \tag{8}$$

$$V_\alpha = tanh(W_4 V_T + b_2) \tag{9}$$

$$\alpha_t = softmax\left(\left(h_\alpha^t \oplus V_\alpha\right) + b_3\right) \tag{10}$$

Where W_3, W_4 and b_1, b_2 are optimized through training and learning， α_t is weight score for each frame of image.Where $\left(h_\alpha^t \oplus V_\alpha\right)$ denotes adding each column of V_α and h_α^t.The initial state of h_t is the hidden state of the last frame output of the LSTM of the encoded part.The video features that are followed by the frame-level attention mechanism can be expressed as:

$$\phi^t(V) = \sum_{i=1}^{N} \alpha_{it} v_i \tag{11}$$

In the above work, the weighted video features can be called context vectors. That is, $C \in R^{b \times 1 \times dc}$, where b is batch_size, and d_c is the dimension size of the context vector.The specific structure of the frame-level attention mechanism model is as shown in Figure 3.

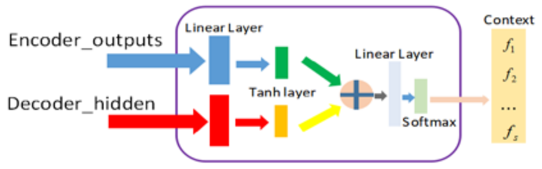


Figure 3 Internal structure of the frame-level attention mechanism

3.4.2 Compact Bilinear Pooling Layer

In the combination of the video feature vector and the current text word feature vector in [29], which is the traditional method . The traditional combination method is a little bit plus, dot multiplication, full connection, etc, but the vectors generated by these methods are not expressive enough, which make the feature vector of the video each element and each element of the text feature vector cannot interact.So inspired by [10], we used the Compact Bilinear Pooling layer to fine-grain the two types of feature vectors.As the Figure 4 shows, first we transform the context vector into $C_1 \in R^{b \times d \times 1 \times 1}$,where b and d are the same as the dc of the context vector C.We transform the text feature vector of the word into $T \in R^{b \times d \times 1 \times 1}$,equal to the above of b, d.We use these two feature vectors as input to the Compact Bilinear Pooling layer.

The context vector C_1 and the text vector T are extrapolated so that each element of the two feature vectors can interact in pairs, but it will cause the dimension to become the original squared, and the latter neural network layer will add many parameters. In order to make computing more efficient,we use the count sketch[28-29]to reduce the dimension to find the outer product.We use the count sketch to reduce the dimension to find the outer product.

$$\psi(C_1 \otimes T, h, s) = \psi(C_1, h, s) * \psi(T, h, s) \tag{12}$$

where * represents a convolution operation in the frequency domain,ψ represents the approximation using the count sketch matrix, $\psi(C_1, h, s) * \psi(T, h, s)$ represents FFT^{-1}(FFT($\psi(C_1, h, s)$) Θ FFT($\psi(T, h, s)$)).

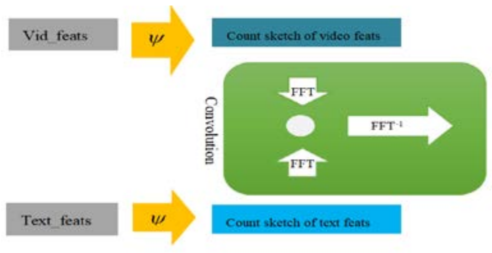


Figure 4 The Structure Of Compact Bilinear Pooling

So we use the CBP layer to complete the fine-grained fusion of the video feature vector and the text feature vector at the current moment.

3.4.3 LSTM For Decoding

The output we get with the CBP layer will be the input to the LSTM of the decoded part,i.e decoder_input$\in R^{b\times 1\times 2d}$.The word entered at the beginning is the start character <BOS>,and the hidden state decoder_hidden of the LSTM of the decoding part at the start time is the hidden state (encoder_hidden$\in R^{1\times b\times d}$) of the last moment reserved by the encoding part LSTM,which takes over the overall feature information of the encoded video sequence.When the start word <BOS> and the video feature together predict the first word, training and testing are different from here.

When the network framework is in the training state, we do not use the first word predicted by the model as the input of the second word prediction, but the first word of the real caption from the dataset as the input of the model, and we use this mode for training each moment.Until we train to meet the 28th cycle, stop training immediately, which will meet the output of different lengths.The total amount of lexicon words in the dataset is known during training,Z_t is the word with the highest probability of output from the softmax function.Follow this step to get the predicted word sequence in turn.The specific calculation is as follows :

$$Z_t\text{, Decoder\_hidden} = \text{Decoder\_LSTM(Decoder\_input, Decoder\_hidden)} \tag{13}$$

Z_t is the value predicted by the LSTM of the decoding part, and Z_t is input into the log_softmax function, and then the tag number corresponding to the value is found according to the total number of words in the word library, and the word corresponding to the current time is obtained.

$$\text{p}(y/Z_t) = log\frac{exp(W_y Z_t)}{\sum_{y' \in V} exp\left(W_{y'} Z_t\right)} \tag{14}$$

Finally, the predicted word sequence and the real caption in the dataset are used as the predictor and label input of the loss function, and the optimization training is started.In training time, our goal is to minimize the sum of the error values of the predicted 28 words. At each point in the test stage, we use the words predicted by the framework as input to the model at the next moment.When the model predicts <EOS>, the prediction of the video description is stopped.

4 Experimental Study

4.1 Dataset

MSR-VTT

MSR-VTT [20] is a large-scale benchmark data set for video understanding that covers 20 types of video and has a wide variety of video content. Specifically, it contains 10,000 video clips with a total duration of 41.2 hours and 20 sentences per clip. We follow the official evaluation agreement provided by [20].

4.2 Evaluation criteria

The task of video subtitles shares similar evaluation metrics with machine translation, such as BLEU, ROUGE-L, METEOR, which are widely used to evaluate the quality of video subtitles. In order to provide a comprehensive assessment, this paper uses all three of the above indicators.

There are four versions of BLEU, namely BLEU 1-4. It is a popular machine translation evaluation index, a precision-based similarity measure, which is used to analyze the degree of co-occurrence of n-gram ancestors in candidate translations and reference translations.
ROUGE-L is a similarity measure based on recall rate. Similar to BLEU, it mainly investigates the sufficiency and fidelity of translation.
The METEOR measure is based on a single-precision weighted harmonic mean and a single-word recall rate. The purpose is to solve some of the defects inherent in the BLEU standard. It also includes functions that are not available in other indicators, such as semantic matching. Calculating METEOR requires a predetermined set. Calibration, which is based on the WordNet synonym, is obtained by minimizing the consecutively ordered chunks in the corresponding statement.

4.3 Training details

4.3.1 Category training

Since all videos in the MSR-VTT data set are divided into 20 categories, we can take advantage of this a priori information.In the previous work, the category information and the video features were directly fused. In our work, the category is treated as a label of the video, and the mapping relationship between the video and the category is trained to achieve reasonable use of the category characteristics of the video.We use Resnet50[21] as a training model, call its weights pre-trained in Imagenet, and continue training on this basis.We use Resnet50 as a training model, call the weights pre-trained on Imagenet, and continue training on this weight.Before training, we first need to delete the last layer of the network's fully connected classification layer, replacing it with a fully connected layer with a node of 20. The loss function uses a cross entropy loss function and the optimizer is SGD.Regarding the parameter settings during training, our sample batch size batch_size is set to 16, the learning rate is 0.001, the weight decay rate is 5×10^{-4}, the momentum is set to 0.9, and the epoch is set to 500.

4.3.2 The training of our neural network framework

For the coding part of our framework, we set the number of hidden units of the LSTM for encoding to 512, and the decoding part of our frame, the number of hidden units of the LSTM for decoding is also 512.When converting words into vectors, we do not use the traditional one-hot vector method, but use the Embeding neural network. The number of input nodes of the network is the total number of dictionary's words, and the number of output nodes is 512.To prevent overfitting, we added a dropout[22] layer to the model where the dropout_ratio of the encoded portion is set to 0.2 and the dropout_ratio ratio of the decoded portion is 0.1. The sample batch size Batch_size is 32, the epoch is

300, the loss function is NLLLoss, the optimizer is Adam[30], the optimizer's parameter setting, the learning rate is 4×10^{-4}, and the weight attenuation rate is 5×10^{-4}.

Both of the above trainings were performed on Nvidia 1080 GPUs in 8G of RAM.

4.4 Experimental Result

Analysis of experimental results on MSR-VTT dataset

Table 1: Compare the test set in MSR-VTT with the top 5 methods

Model	METEOR	BLEU@4	ROUGE-L
ruc-uva	26.9	38.7	58.7
VideoLAB	27.7	39.1	60.6
Aalto	26.9	39.8	59.8
v2t navigator	28.2	40.8	60.9
WSDVC	28.3	41.8	61.1
Ours	**28.6**	**40.6**	**61.3**

TABLE 1 shows that our network framework compares with the top 5 methods in the MSR-VTT dataset. Overall, the proposed network framework is 28.6% Meteor, 61.3% Rouge_L and 40.6% BLEU@. 4 indicators have more competitive ability with the most advanced level.The ruc-uva[23] method is more complicated. After using CNN to extract the video features, the method uses the Video tagging method to extract the keywords of the video. These keywords are used together with the video CNN feature as the input of the decoder.In addition, the keywords generated by tagging are still used to generate the ordering of sentences. VideoLAB[24]uses the Encoder-Decoder framework, which uses a combination of video, image, sound, and category information. Both encoder and decoder use LSTM.V2t_navigator [26] also uses the Encoder-Decoder framework, using multi-modal feature fusion, the encoder uses a single-layer full-link layer with no activation function, and the decoder uses LSTM.Aalto[25]captures video content based on objects and attributes, then captures motion stream information, and uses these two functions to train two separate models.Finally, an evaluation model is trained to select the best title from the candidate pools generated by the specialized models in these field.WSDVC[27] focuses on the issue of the deck video captioning. The dense video captioning is to generate all possible descriptions of a video. It proposes a weakly supervised method, Multi-instance multi-label learning (MIMLL). MIMLL learns the vocabulary vector corresponding to each video image region directly from video-sentence data (such data is weakly supervised for this problem). These lexical description vectors are then combined as input to the encoder-decoder to implement video captioning.Our proposed neural network framework has improved performance in all three of the above indicators.

TABLE 2 Comparison of test sets in MSR-VTT using different structures and basic models

Model	METEOR	BLEU@4	ROUGE-L
S2VT(V+C)	22.3	32.4	50.5
Frame attention+S2VT(V+C)	26.9	35.1	57.8
Frame attention +CBP+S2VT(V+C)	27.5	37.3	59.0
Two level attention+CBP+S2VT(V+C)	**28.8**	**38.4**	**61.4**

TABLE 2 shows the comparison between our framework and S2VT. The S2VT model uses the stacked LSTM to encode the frame feature sequence and generate caption. That is, using one of the LSTM to encode video frame sequences, the hidden state of the encoded LSTM is input to the LSTM of the decoding part and the hidden state of the encoded LSTM is the initial hidden state of decoding LSTM, the input of decoding LSTM is <BOS> start character, and the words are generated step by step; three models improved based on this model are higher than the model in three evaluation criteria.This is because the framework we propose is based on the S2VT framework, and for the shortcomings of S2VT, a model with corresponding characteristics is added to make up for these shortcomings.The main reasons are as follows:

In S2VT, after the encoder encodes the video frame by frame, the obtained video representation is used as the input of the decoder. Then the word is predicted by the decoder each time, the corresponding video representations are the same,which is not logica due to a sentence is composed of different words.Therefore, we added a time domain attention mechanism to this shortcoming, which will weight the video representation feature vector according to each input word, so that the obtained video representation has a corresponding relationship with the input word at that moment, which will make The resulting video description is more logical.In the Figure 5 and Figure 6 shows a video, the weight score of each frame corresponding to each word, that is, the visualization of the frame-level attention mechanism, we sample 5 frames from 50 frames of the video for visual description.We find that the contribution of f_0 is the largest in the contribution of each frame corresponding to the word 'man' and the word 'blue', and f_{10} and f_{20} are almost flat.However, in the word ‘riding’, the contribution of f_{30} is the biggest, and we can clearly see the action of the 30th frame of the character riding.While other frames are less obvious or even not, the word 'road' is almost the same contribution for each frame. The results verify that our multi-functional combination framework has different visual information for different words.

Secondly, in the decoding stage, the input video representation is to be merged with the word input. In order to make the visual feature and the text feature deeper, rather than simple splicing, we use the Compact Bilinear Pooling layer to fuse the two types of features. The deep interaction of the two types of features is guaranteed, and the feature vector of the huge dimension is not generated, and the calculation parameters are reduced.

Furthermore, S2VT does not perform deeper processing on video features, ie, does not consider redundant visual information for each frame of redundant video. Excessive redundant visual redundancy information can interfere with subtitle generation, so our framework adds a convolutional area attention mechanism, through the combination of two attention mechanisms and the Compact Bilinear Pooling layer, which will make the generated video description more accurate.

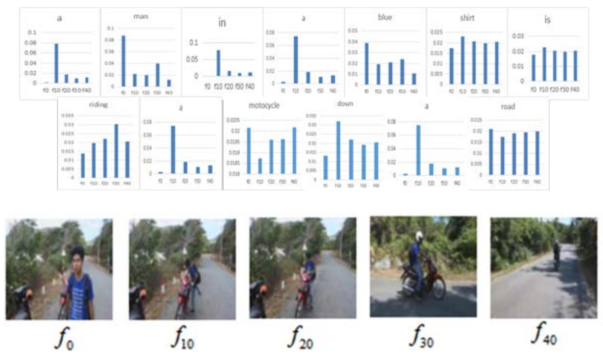


Figure 5 The caption generation of the motorcycle video and the contribution of each frame of the video corresponding to each word, where f_0,f_{10},f_{20},f_{30},f_{40} represent the position of the frame in the video.

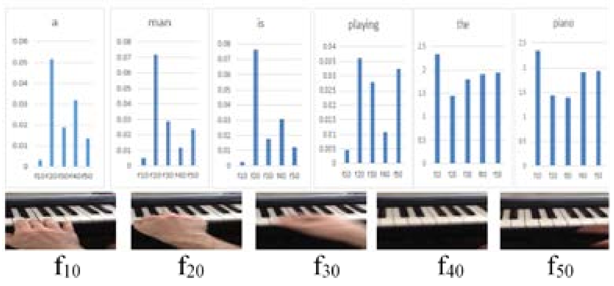
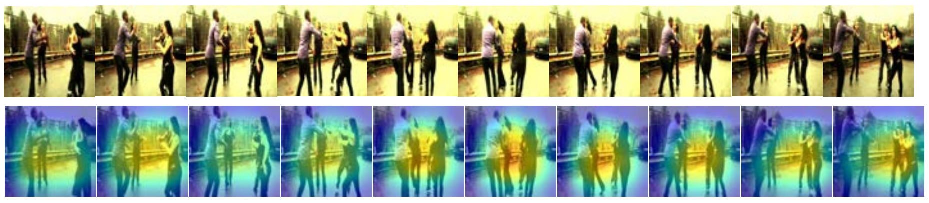


Figure 6 The caption generation of the video playing the piano and the contribution of each frame of the video corresponding to each word, where f_{10},f_{20},f_{30},f_{40},f_{50}represent the position of the frame in the video.

In Figure 7, we made a visualization of the video frame by the regional attention mechanism. The rendering is the thermal map, that is, the deeper the color, the higher the weight of the regional attention mechanism is. The shallower the area, the lower the full rating of the block area by the regional attention mechanism.

We find that the regional attention mechanism can focus on the important objects of each frame of the picture. If there is no attention mechanism to pay attention to everyone in the picture, then the subtitles generated by the video are not described enough, such as without regional attention. The subtitles generated by our web framework only describe one 'woman', while the attention produces a description of 'a group of people'.Thus reflecting the importance of regional attention mechanisms.



GT: a group of people are dancing outside
our framework： a woman is dancing
LRA+our framework:a **group** of **people** are dancing

Figure 7 Regional attention mechanism visualization heat map , Video descriptions generated by our neural network framework with and without attention.

5 Conclusion

In this article, we have improved S2VT in order to solve many shortcomings of framework based on the CNN-RNN and the S2VT framework. The two-level attention mechanism(Region attention and Frame attention), the fusion method of video features and text features and S2VT are concentrated in a network framework, and the category pre-training method is used to fuse the category information. A multi-functional composite network framework is proposed to realize the generation of video subtitles. The framework we propose can focus on the most relevant area of each frame of the video, and also generate the video feature vector that is most relevant to the current word when the text is generated. In the fusion of video features and text features, we use the Compact Bilinear Pooling layer for fine-grained fusion. We also have video features with category information on the input features. In summary, we can make our network framework in MSR-VTT data set has the ability to compete with advanced levels. However, the framework we proposed is not ideal when caption are generated for complex background video, especially when the content of the two types of video scenes is similar, there will be confusion, then for deeper video mining. Advanced semantic features are one of the issues we need to solve in the future.

REFERENCES

[1]. Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond J. Mooney, *Integrating language and vision to generate natural language descriptions of videos in the wild*, International Conference on Computational Linguistics, 2014, pp. 1218-1227.

[2]. Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun, *Title generation for user generated videos*, European Conference on Computer Vision, 2016, pp. 609-625.

[3]. Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J.Mooney, and Kate Saenko, *Translating videos to natural language using deep recurrent neural networks*,The 2015 Conference of the North American Chapter of the Association for Computational Linguistics, 2015, pp. 1494–1504.

[4]. Li Yao, Atousa Torabi, Kyunghyun Cho,Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, andAaron C. Courville, *Describing videos by exploiting temporal structure*,IEEE International Conference on Computer Vision, 2015, pp. 4507–4515.

[5]. Xuelong Li, Bin Zhao and Xiaoqiang Lu, *MAM-RNN: Multi-level Attention Model Based RNN for Video Captioning*, Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 2208–2214.

[6]. An-An Liu (Member, IEEE), YuRui Qiu, YongKang Wong, (Member, IEEE),Yu-Ting Su, and MoHan KanKanHai, *A Fine-grained Spatial-Temporal Attention Model for Video Captioning*, IEEE Access, 2018, pp.1-1.

[7]. Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney,Trevor Darrell, and Kate Saenko, *Sequence to sequence- video to text*, IEEE International Conference on Computer Vision, 2015, pp. 4534–4542.

[8]. ChunleiWu, YiweiWei, XiaoliangChu, SunWeichen, FeiSu and LeiquanWang, *Hierarchicalattention-basedmultimodalfusion for video captioning*, Neurocomputing, 2018, pp. 362-370.

[9]. Gao, Yang， Beijbom, Oscar， Zhang, and Ning, *Compact Bilinear Pooling*, IEEE Computer Society,2016.

[10]. Yin Cui, Guandao Yang, Andreas Veit, Xun Huang and Serge Belongie, *Learning to Evaluate Image Captioning*, CVPR, 2018.

[11]. Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond J. Mooney, Trevor Darrell, and Kate Saenko, *YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-Shot Recognition*, IEEE International Conference on Computer Vision, 2013, pp. 2712–2719.

[12]. Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama, *Generating natural-language video descriptions using text-mined knowledge*, Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.

[13]. Alex Krizhevsky, Ilya Sutskever,and Geoffrey E. Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems,2012, pp. 1106–1114.

[14]. Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, CoRR, 2014, abs/1409.1556.

[15]. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, *Going deeper with convolutions*,IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[16]. Oriol Vinyals, Alexander Toshev, Samy Bengio and Dumitru Erhan,*Show and Tell: A Neural Image Caption Generator*, CVPR, 2015.

[17]. Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas,Christopher Pal,Hugo Larochelle and Aaron Courville, *Video Description Generation Incorporating Spatio-Temporal Features and a Soft-Attention Mechanism*,ICCV, 2015.

[18]. HuiyunWang,ChongyangGao and YahongHan,*Sequence in sequencefor video captioning*,Pattern RecognitionLetters,2018, pp. 1-8.

[19]. Sepp Hochreiter and Jurgen Schmidhuber, *Long short-term memory*， Neural computation， 1997， pp. 1735 – 1780

[20]. J. Xu, T. Mei, T. Yao, and Y. Rui, *Msr-vtt: A large video description dataset for bridging video and language*,Proc. IEEE Conf. Comput.Vis. Pattern Recognit, 2016, pp. 52885296.

[21]. Akiba, Takuya， Suzuki, Shuji， Fukuda and Keisuke,*Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes*, 2017.

[22]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*,The Journal of Machine Learning Research, 2014,vol.15,no.1pp.1929-1958.

[23]. J. Dong, X. Li, and et al. *Early embedding and late reranking forvideo captioning*,ACM Multimedia Grand Challenge,2016, pp. 1082-1086.

[24]. V. Ramanishka, A. Das, D. H. Park, and et al. *Multimodal video description*, ACM Multimedia Grand Challenge, 2016,pp. 1092-1096.

[25]. R. Shetty and J. Laaksonen. *Frame-and segment-level features and candidate pool evaluation for video caption generation*,2016.

[26]. Q. Jin, J. Chen, and et al. *Describing videos using multi-modal fusion*, ACM Multimedia Grand Challenge, 2016.

[27]. Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li,Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue,*Weakly Supervised Dense Video Captioning*,IEEE, 2017.

[28]. Braverman V , Chestnut S R , Ivkin N , et al, *Beating CountSketch for heavy hitters in insertion streams* ,Proceedings of the 48th Annual ACM SIGACT Symposium On Theory of Computing - STOC, 2016,pp.740-753.2.

[29]. Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, and Trevor Darrell1 Marcus Rohrbach1,*Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding*,2016.

[30]. Kingma D , Ba J . *Adam: A Method for Stochastic Optimization[J]*. Computer Science, 2014.