# A Novel Approach for Video Captioning Based on Semantic Cross Embedding and Skip-Connection

Rakesh Radarapu(✉), Nishanth Bandari(✉), Satwik Muthyam(✉), and Dinesh Naik(✉)

National Institute of Technology Karnataka, Surathkal, India

**Abstract.** Video Captioning is the task of describing the content of a video in simple natural language. Encoder-Decoder architecture is the most widely used architecture for this task. Recent works exploit the use of 3D Convolutional Neural Networks (CNNs), Transformers or by changing the structure of basic Long Short-Term Memory (LSTM) units used in Encoder-Decoder to improve the performance. In this paper, we propose the use of a sentence vector to improve the performance of the Encoder-Decoder model. This sentence vector acts as an intermediary between the video space and the text space. Thus, it is referred to as semantic cross embedding that bridges the two vector spaces, in this paper. The sentence vector is generated from the video and is used by the Decoder, along with previously generated words to generate a suitable description. We also employ the use of a skip-connection in the Encoder part of the model. Skip-connection is usually employed to tackle the vanishing gradients problem in deep neural networks. However, our experiments show that a two-layer LSTM with a skip-connection performs better than the Bidirectional LSTM, for our model. Also, the use of a sentence vector improves performance considerably. All our experiments are performed on the MSVD dataset.

**Keywords:** Video captioning · Skip-connection · Semantic cross embedding · Sentence vector

## 1 Introduction

Humans are quite capable of giving a proper caption for a video clip without many semantic errors or misconceptions. Figure 1 shows the description given by humans for the video which is quite accurate. However, for machines it is not quite easy to identify the objects involved, the interactions among them and generate a fitting description. With the introduction of CNNs reading an image has become a feasible task for machines. The progress in Natural Language Processing, helped machines to understand human language in the form of

S. K. Singh et al. (Eds.): CVIP 2020, CCIS 1378, pp. 465–477, 2021.
https://doi.org/10.1007/978-981-16-1103-2_39

word embeddings. Word embeddings are used in applications like neural machine translation, text generation and information retrieval.

Video Captioning is processing the visual information to generate textual information. Processing videos is a challenging task because unlike an image, a video has both spatial and temporal information. CNNs are capable of capturing the spatial information in the images and the temporal information of the video is captured by the use of one or more LSTM layers.

Attention is a mechanism used for tasks related to language translation, visual content description, question-answering and became important in the Encoder-Decoder architecture which is used for all sequence-to-sequence related tasks. Soft attention is used in particular for these where the context vector for the Decoder is computed as a weighted sum of the encoder output states.

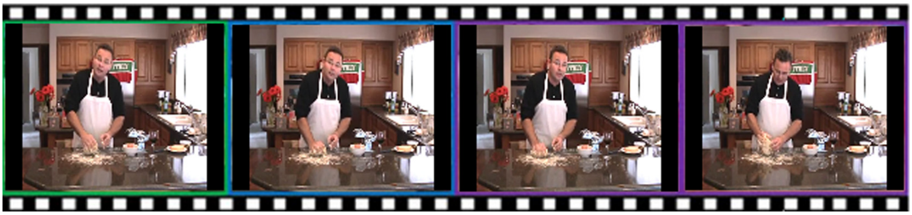


**Fig. 1.** Caption: a man is kneading a ball of dough

Thus the overview of our contribution is as follows:

I Word embeddings from pre-trained models like BERT base [3], GloVe, Elmo are used. Comparisons among these embedding approaches based on the performance of our model are made.

II Experiments are conducted with feature vectors for video frames taken from different CNNs like Inception-v3, VGG-16 and NASNet-Large.

III The use of Multi-Head attention for the Decoder to softly select the encoder states.

IV Employing the use of a skip-connection in the Encoder and comparing it to the Bidirectional model.

V The concept of a sentence vector to aid the Decoder. A sentence vector is a part of the caption space. We try to map the video information, which is in visual space, directly to its corresponding caption in the caption space.

## 2 Related Work

Initial works mostly concentrated on Image Captioning that were later extended to video captioning. Vinayals et al. [18] has proposed a deep-recurrent model that used latest advances in computer vision and machine translation to generate captions to images. His model was trained to maximize the likelihood of the target description sentence for a training image. You et al. [25] combines both top-down and bottom-up strategies to extract richer information from an image, and combines them with a RNN that can selectively attend semantic attributes detected from the image. Fang et al. [4] proposed a language model to generate captions and rank them using a dual-stream RNN model. The model learns to extract nouns, verbs, and adjectives from regions in the image. Using the above words the model generates a meaningful sentence describing the image.

Y. Pan et al. [9] proposed a novel deep learning architecture using Long Short-Term Memory with Transferred Semantic Attributes (LSTM-TSA). LSTM-TSA is used for transferring semantic attributes from the images and videos into the CNNs and RNNs in an end-to-end manner. In this, image and video semantics reinforce each other to boost video captioning. L. Gao et al. [6] proposed an Encoder-Decoder architecture with an attention LSTM in the Decoder to improve the context of caption and introduced a new loss function to improve the semantics of the caption. Song et al. [14] proposed a novel hLSTMat encoderdecoder framework, which integrates a hierarchical LSTMs, temporal attention and adaptive temporal attention. This model decides when to use visual information and semantic information on its own.

Xu et al. [22] used combination of convolutional neural networks and recurrent neural networks called RCN and combined it with a trainable vector of locally aggregated descriptor (VLAD) layer to develop a novel Sequential layer called SeqVLAD. They have tested this framework on video captioning and video action recognition task and proved its effectiveness. Ning Xu et al. [21] designed attention in attention network to hierarchically explore the attention fusion in an end-to-end manner. It specifically has multiple encoder attention modules and fusion attention modules.

Bin et al. [2] proposed a new LSTM called Bi-directional LSTM that processes videos in both forward and backward direction to gain information for decoding from future time steps as well. Also, a soft attention mechanism is proposed that focuses on targets with certain probabilities at every timestep. J. Song et al. [13] proposed a new end-to-end framework known as multi-model stochastic RNNs (MS-RNN) that takes uncertainty in the data into consideration by introducing stochastic variables. This approach combines BiLSTM and soft attention mechanism with a new LSTM called S-LSTM that introduces uncertainty in the training phase.

Y. Yang et al. [23] uses the concept of generative adversarial networks shortly known as GAN for captioning the videos. The generator is responsible for giving captions for videos where as the discriminator classifies the sentences into true or generated data thereby acting as an adversary to caption generator. Q. Zheng et al. [26] proposed a Syntax-Aware Action Targeting (SAAT) module which learns actions by simultaneously referring to the subject and video dynamics. First, they identify the subject by mapping global dependence among multiple objects and then decode action from a common space that fuses the embedding of the subject and the temporal feature of the video.

# 3 Proposed Approach

Our goal is to generate a natural language description for a given video. We propose a Semantic Cross Embedding (SCE) based Encoder-Decoder architecture with a skip-connection in the Encoder for this task. Our model takes feature vectors extracted from a 2D CNN as inputs and encodes them to capture the spatio-temporal information from the video. The overall architecture of our final model is shown in Fig. 2. We make use of the concept of a sentence vector which will be explained in Sect. 3.4.

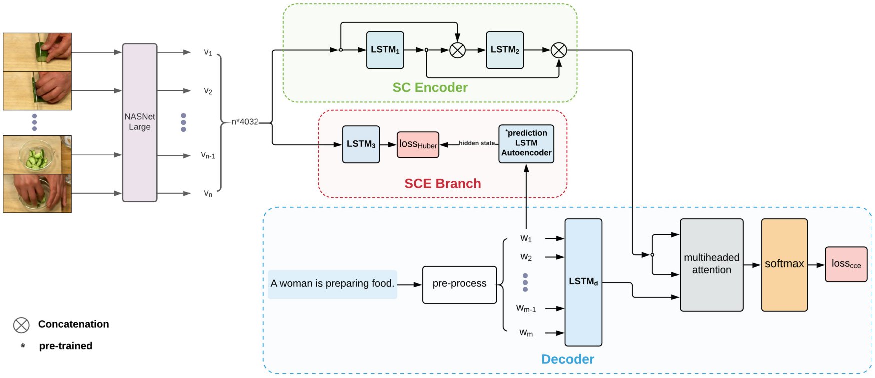


**Fig. 2.** The Architecture of our video captioning model with Semantic cross embedding and skip-connection (SCE + SC) has 3 main modules: a) Encoder with skip-connection b) Decoder with Multi-Head Attention c) Semantic Cross Embedding Branch.

## 3.1 Feature Extraction

The first phase consists of extracting feature vectors from the videos. The videos are broken down to frames, resized and re-scaled. Resize and re-scale are needed because the videos from the dataset are of varying frame sizes and also to suit the input needs of the pre-trained CNN model. The pre-processed frames are fed, in order, to a CNN model to extract feature vectors that are a high-level representation of videos. Three different models namely NASNet-Large, Inception-v3, VGG-16 are used to extract feature vectors.

**NASNet-Large** is a Convolutional Neural Network that takes an image of size $331 \times 331$ as input and outputs a feature vector of size 4032.

In NASNet-Large, though the overall architecture is predefined as in [27], the blocks or cells are not predefined by authors. Alternatively, they are explored by reinforcement learning search method i.e. the number of repetitions N and the number of initial convolutional filters are as free parameters and used for scaling. The variant used for our work is taken from the Tensorflow Hub and consists of 18 Normal cells, starting with 168 convolutional filters.

A Normal cell is a group of convolutional cells that return a feature map of the same dimension and a Reduction cell is a group of convolutional cells that return a feature map where the feature map height and width is decreased by a factor of two.

**InceptionV3** [15] is a Convolutional Neural Network developed by Google for image captioning that takes a 299 × 299 sized image as input and outputs a feature vector of size 2048.

The basic unit of this network is an ‘inception cell’, which consists of multiple convolutions running in parallel and ultimately concatenating the output. The input channel depth is adjusted using 1 × 1 convolutions. Each inception cell uses 1 × 1, 3 × 3 and 5 × 5 filters to learn features at various scales from input. Max pooling is used with ‘same’ padding to retain the dimension for concatenation.

**VGG 16** Visual Geometry Group [12] (VGG-16) was developed by Oxford that takes a 224 × 224 sized image as input and outputs a feature vector of size 4096.

VGG-16 uses only 3 × 3 filters with a stride of 1 in the convolution layers and 2 × 2 filters in the pooling layers with ‘same’ padding. All hidden layers use ReLU activation function and are followed by 3 fully-connected layers.

### 3.2 Terminology and Notation

Let the video for which a caption needs to be generated be denoted by V = $\{v_1, v_2, v_3 ...., v_n\}$, where n is the number of frames. Now the visual features extracted, from video V as described in Sect. 3.1, be X = $\{x_1, x_2, x_3 ...., x_n\}$ $\epsilon R^{d_i * n}$, where $d_i$ is the dimension of the feature vector of a single frame. Let the caption be denoted by $W \epsilon R^{d_w * c}$, where $d_w$ is the dimension of a word embedding and c is the number of words in the caption. The caption generated by the model be represented as $W' \epsilon R^{d_w * c}$.

### 3.3 Encoder-Decoder

The role of the Encoder is to capture the temporal features from the 2-D CNN features extracted at the frame level. We use Long Short-Term Memory Networks for this task. LSTMs are used to extract a fixed-dimensional vector representation for a series of frames. Our Encoder consists of a two-layer LSTM with a skip-connection.

The Decoder should learn to predict the caption given the video information. The caption is generated one word at a time. So given a current word and the output from the Encoder, the Decoder predicts the next word in the caption. We use multi-head attention for the model which takes the encoder output and the decoder state to attend to selective regions in the encoder output which determine the Decoder output. The loss that captures the translation from videos to words is computed as

$$loss_1 = -\sum_{t=1}^{N_w} \log P(w_t|E, w_1, w_2, ..., w_{t-1}) \tag{1}$$

where, $N_w$ represents the number of words in the caption and Eq. 2 represents the probability of the predicted word $w_t$ given the previously generated words $w_1$, $w_2$, ..., $w_{t-1}$ and the encoder output $E$.

$$P(w_t|E, w_1, w_2, ..., w_{t-1}) \tag{2}$$

### 3.4 Semantic Cross Embedding

The feature vector X from the video, as described in Sect. 3.2, is passed to an LSTM layer to generate a 768-D vector. During the training process, we compare it against the hidden state from an LSTM autoencoder, as shown in Fig. 3, trained on the caption set. Since this hidden state captures the content of the caption and is used in regenerating the caption, it gives extra information to the Decoder to predict the next word. Since this vector summarizes a caption we refer to it as a sentence vector.

Let the sentence vector from the language model be SV and the sentence vector generated by SCE branch be $SV'$ then the Huber loss is computed as

$$e_k = \begin{cases} \frac{1}{2}(sv_k - sv'_k)^2 & for\ |sv_k - sv'_k| \leq \delta \\ \delta|sv_k - sv'_k| - \frac{1}{2}\delta^2 & Otherwise \end{cases} \tag{3}$$

$$loss_2 = \sum_{k=0}^{d_w} e_k \tag{4}$$

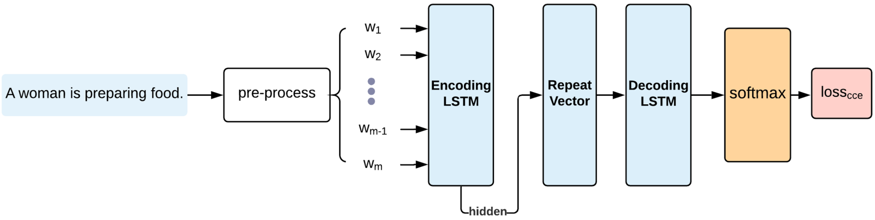


**Fig. 3.** LSTM autoencoder

### 3.5 Multi-head Attention

The encoder output contains information about the entire video. But for the Decoder to generate the next word given the current word it needs to select only a subset of the features. So scalar dot product attention [16] allows the Decoder to attend to only selective information from the Encoder. Consider Eq. 5, Query, Value represents the encoder output information, Key represents the decoder previous state information. Here the encoder output is weighted for different regions based on the current decoder states to produce a context vector for the next word. This constitutes one head of the multi-head attention. The use of multiple heads makes it possible for the Decoder to attend to the information from the Encoder at different positions from different representational spaces at the same time. Single head attention is computed as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_n}})V \tag{5}$$

$$(Q, K, V) = (R_d W^Q, R_e W^K, R_e W^V) \tag{6}$$

where $R_d$ is the decoder states, $R_e$ is the encoder states, $\{W^Q, W^K, W^V\}$ $\epsilon R^{d_i * d_n}$ are the weights for multiple attention heads, where $d_i$ is the dimension of the attention input and $d_n$ is the number of units in an attention head.

### 3.6 Training

We train the language model initially to get the sentence vectors of all the captions. Then we train the proposed architecture, loading the pre-trained language model, as shown in Fig. 2 with an objective loss function built by integrating two loss functions which simultaneously consider video translation to words and sentence vector generation. The experimental setup can be found in Sect. 4.2

$$loss = \lambda loss_1 + (1 - \lambda) loss_2 \tag{7}$$

where $\lambda$ is a hyper-parameter between 0 and 1.

## 4 Experimental Results and Discussion

### 4.1 Dataset

The Microsoft Video Description (MSVD) dataset has a total of 1,970 short video clips from YouTube, with attached human-generated descriptions. On average, there are around 41 descriptions per video. This dataset contains about 80,000 clip-description pairs, with each clip having descriptions in different languages. For our work we use the English descriptions only. We adopt the same data split as provided in [17], with 1,200 video clips for training, 100 video clips for validation and the remaining 670 clips for testing.

### 4.2 Implementation

**Data Preprocessing and Evaluation.** The average duration of the videos from the corpus is 10.2 s. So we sample only 28 frames per clip uniformly. These frames are pre-processed and passed to a CNN, here we consider NASNet-Large pre-trained model, to extract a 4032-D feature vector for each frame. So we have a 28*4032-D vector for each video from NASNet-Large. Every video has at least 28 frames and hence there is no need for padding here.

The captions provided for the videos are collected from different sources and are of varying lengths. Therefore we remove the punctuation from the captions, convert them to lower case and tokenize. The misspelt words and the least occurring words are filtered out from the vocabulary. Captions are adjusted to a length of 20. Any caption exceeding the size is truncated and a caption with less than 20 tokens is padded. Word Embedding techniques like BERT [3], GloVe, Elmo are used to obtain 768-D, 300-D or 1024-D vector for each token. We compare the performance of different word embeddings for our model in 4.4. The *bos*, *eos* tokens are used to mark the beginning and the end of the caption, *pad* as padding token and *unk* as unknown, for words not found in the vocabulary.

**Experimental Setup.** The 28*4032 feature vector represents the video to be processed. It is passed to the Encoder, which is a two-layer LSTM with a skip-connection. The number of units in the LSTM is taken as 512. So we get 28*512-D from layer 1 and 28*512-D from layer 2 that makes a 28*1024-D vector, from the Encoder. The SCE branch contains a single LSTM layer with 768 units and a pre-trained LSTM autoencoder. The number of units in the attention layers is set to 512. The number of units in the Decoder LSTM is taken as 1024. Adam Optimizer with a learning rate of 5e-4 is used for training and a batch-size of 64. Beam search with a beam-width of 3 is used during testing.

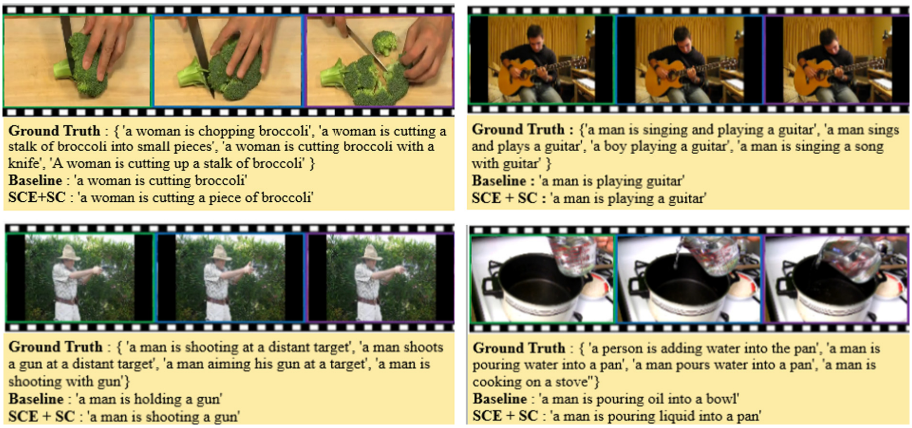
### 4.3 Evaluation Metrics

To evaluate the performance of the model we have considered 3 standard metrics: BLEU, ROUGE, CIDER and to compare the final results of our work with state-of-art papers on MSVD we have considered BLEU, METEOR and CIDER as the metrics.

- **BLEU (BiLingual Evaluation Understudy)** BLEU calculates the n-gram hit ratio of output caption against the ground truth. It is suitable for short sentences. We have used Bleu 1,2,3 and 4 scores for our performance analysis.
- **ROUGE (Recall-Oriented Understudy of Gisting Evaluation)** Rouge metric is based on the longest common subsequence. The longer the common subsequence the more similar the reference and candidate sentences.

- **METEOR (Metric for Evaluation of Translation with Explicit Ordering)** METEOR is a mean value of unigram-based recall and precision scores. The main difference between METEOR and BLEU is this metric combines both recall and precision.
- **CIDEr (Consensus-based Image Description Evaluation)** CIDER score for n-grams of length n is computed using the average cosine similarity between the generated sentence and the human annotated sentences, which accounts for both precision and recall.

## 4.4 Results and Analysis



**Ground Truth** : { 'a woman is chopping broccoli', 'a woman is cutting a stalk of broccoli into small pieces', 'a woman is cutting broccoli with a knife', 'A woman is cutting up a stalk of broccoli' }
**Baseline** : 'a woman is cutting broccoli'
**SCE+SC** : 'a woman is cutting a piece of broccoli'

**Ground Truth :** {'a man is singing and playing a guitar', 'a man sings and plays a guitar', 'a boy playing a guitar', 'a man is singing a song with guitar' }
**Baseline :** 'a man is playing guitar'
**SCE + SC :** 'a man is playing a guitar'

**Ground Truth** : { 'a man is shooting at a distant target', 'a man shoots a gun at a distant target', 'a man aiming his gun at a target', 'a man is shooting with gun'}
**Baseline** : 'a man is holding a gun'
**SCE + SC** : 'a man is shooting a gun'

**Ground Truth** : { 'a person is adding water into the pan', 'a man is pouring water into a pan', 'a man pours water into a pan', 'a man is cooking on a stove''}
**Baseline** : 'a man is pouring oil into a bowl'
**SCE + SC** : 'a man is pouring liquid into a pan'

**Fig. 4.** The qualitative comparison of our proposed SCE+SC model, baseline and ground truth captions

**Qualitative Analysis.** Figure 4 shows the qualitative comparison of our proposed SCE+SC model against ground truth and baseline model captions. The video frames and ground truth shown in the figure are taken from MSVD dataset. Both the baseline model and our proposed SCE+SC model have given a good set of captions for the videos.

But the baseline model fails to identify the portrayed action and objects in few cases. The SCE+SC model has performed well in object and action identification. For example, the baseline model has given *holding* whereas the action *shooting* identified by SCE+SC model is more precise according to the context. Also, the word *bowl* is given by the baseline model whereas the word *pan* given by SCE+SC model is more appropriate. Therefore, it is clear that the semantic cross embedding helped in getting proper words.

**Table 1.** Performance comparison with various methods on the MSVD test dataset

| Model | B@4 | METEOR | Rouge | CIDEr |
|---|---|---|---|---|
| S2VT [17] | – | 29.2 | – | – |
| h-RNN [24] | 49.9 | 32.6 | – | 65.8 |
| V-ShaWei-GA [7] | 47.9 | 30.9 | – | – |
| S2VT+RL [11] | 45.6 | 32.9 | 69.0 | 80.6 |
| SCN-LSTM [5] | 51.1 | 33.5 | – | 77.7 |
| MMVDN [20] | 37.6 | 29.0 | – | – |
| aLSTM [6] | 50.8 | 33.3 | – | 74.8 |
| BAE [1] | 42.5 | 32.4 | – | 63.5 |
| TA [24] | 41.9 | 29.6 | – | 51.7 |
| M3 [19] | **52.8** | 33.3 | – | – |
| MARN [10] | 48.6 | 35.1 | 71.9 | **92.2** |
| S-VC [8] | 35.1 | 29.3 | – | – |
| Baseline | 49.0 | 33.8 | 70.6 | 82.3 |
| SCE+SC | 52.1 | **35.5** | **72.1** | 85.7 |

**Quantitative Analysis.** We compare the performance of our method with state-of-the-art methods using BLEU@4, METEOR, ROUGE and CIDEr metrics on MSVD dataset. The methods include S2VT [17], h-RNN [24], V-ShaWei-GA [7], S2VT+RL [11], SCN-LSTM [5], MMVDN [20], aLSTM [6], BAE [1], TA [24], M3 [19], MARN [10], S-VC [8] including our baseline method. All the methods have followed the same train-validation-test split provided in [17].

Table 1 shows the metrics of various models. M3 [19] achieved the highest BLEU@4 score followed by our model which outperformed the rest in terms of BLEU@4 score. This shows that our proposed model has been able to identify the exact set of words in the dataset better than the other models which improved the BLEU@4 scores as it is measured by the n-gram hit ratio.

In terms of CIDEr metric, our proposed model ranked second. Our model achieved the highest Rouge and METEOR scores compared to others'. This trend shows that our SCE+SC model outperforms the state-of-the-art methods in video captioning. When compared to the baseline model, the proposed model achieved the highest scores. This shows that semantic cross embedding and skip-connection has boosted the performance of the model.

**Table 2.** Effect of various Word Embedding techniques on the model training

| Model | B@1 | B@2 | B@3 | B@4 | METEOR | Rouge | CIDEr |
|---|---|---|---|---|---|---|---|
| BERT | **81.0** | **68.8** | 59.1 | 49.0 | **33.8** | **70.6** | **82.3** |
| Elmo | 80.3 | 68.7 | **59.3** | **49.7** | 33.5 | 70.1 | 80.8 |
| GloVe | 80.6 | 67.0 | 55.7 | 47.8 | 33.1 | 70.3 | 80.3 |

Table 2 shows the effect of the use of different pre-trained word embeddings in our base model. With the use of BERT embeddings, the results are slightly better than Elmo and GloVe embeddings.

**Table 3.** Performance comparison of Bidirectional LSTM Encoder and Encoder with a skip-connection

| Model | B@1 | B@2 | B@3 | B@4 | METEOR | Rouge | CIDEr |
|---|---|---|---|---|---|---|---|
| BiLSTM | 81.1 | 69.5 | 59.8 | 49.1 | 33.8 | 69.7 | 79.1 |
| LSTMs with SC | **82.8** | **71.0** | **60.9** | **50.9** | **34.2** | **70.3** | **82.4** |

Table 3 compares the performance of Bidirectional LSTM Encoder and Encoder with a skip-connection. Encoder with a skip-connection performed slightly better for every metric.

**Table 4.** Comparison of different trained CNN models used for obtaining spatial features

| Model | B@1 | B@2 | B@3 | B@4 | METEOR | Rouge | CIDEr |
|---|---|---|---|---|---|---|---|
| NASNet-Large | **75.8** | **61.9** | **50.7** | **41.1** | **31.4** | **68.4** | **76.8** |
| Inception-v3 | 72.6 | 57.9 | 48.9 | 39.2 | 30.8 | 60.5 | 66.4 |
| VGG-16 | 69.8 | 54.5 | 43.1 | 31.4 | 29.3 | 64.6 | 55.4 |

Table 4 shows the comparison between the performances of a basic Encoder-Decoder model with a single-head attention using different 2-D CNN models. It is clear that NASNet-Large features have performed well when compared to VGG-16 and Inception-v3 based feature vectors. So, we have chosen to use NASNet-Large to extract feature vectors for the video/images.

Figure 5 shows BLEU@4 and METEOR scores for different values of the hyper-parameter $\lambda$ in Eq. 7. We can observe that the model performs best when $\lambda = 0.9$.

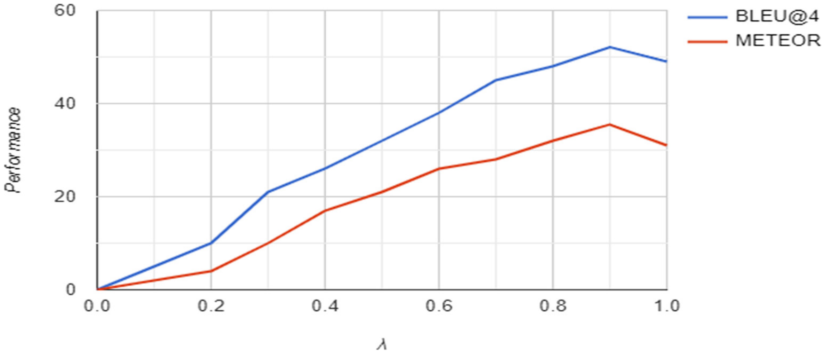


**Fig. 5.** BLEU@4 and METEOR scores for different values of $\lambda$

# 5 Conclusion

In this paper, we proposed semantic cross embedding and skip-connection in Encoder for Video Captioning. The SCE or sentence vector is a representation of the video in the caption space. It assists the Decoder to improve the performance of the model. The use of a skip-connection in the Encoder performs better than a Bidirectional Encoder. Experiments conducted on MSVD validate our approach and analysis. Our approach gave better results when compared to various other state-of-the-art techniques. Our future work would be to make use of 3-D CNNs along with 2-D CNNs to further boost the process of video captioning.

# References

1. Baraldi, L., Grana, C., Cucchiara, R.: Hierarchical boundary-aware neural encoder for video captioning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3185–3194 (2017)
2. Bin, Y., Yang, Y., Shen, F., Xie, N., Shen, H.T., Li, X.: Describing video with attention-based bidirectional LSTM. IEEE Trans. Cybern. **49**(7), 2631–2641 (2019)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
4. Fang, H., et al.: From captions to visual concepts and back. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1473–1482 (2015)
5. Gan, Z., et al.: Semantic compositional networks for visual captioning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1141–1150 (2017)
6. Gao, L., Guo, Z., Zhang, H., Xu, X., Shen, H.T.: Video captioning with attention-based LSTM and semantic consistency. IEEE Trans. Multimed. **19**(9), 2045–2055 (2017)
7. Hao, W., Zhang, Z., Guan, H., Zhu, G.: Integrating both visual and audio cues for enhanced video caption (2017)

8. Li, G., Ma, S., Han, Y.: Summarization-based video caption via deep neural networks. In: Proceedings of the 23rd ACM International Conference on Multimedia, MM 2015, pp. 1191–1194. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2733373.2806314
9. Pan, Y., Yao, T., Li, H., Mei, T.: Video captioning with transferred semantic attributes. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 984–992 (2017)
10. Pei, W., Zhang, J., Wang, X., Ke, L., Shen, X., Tai, Y.: Memory-attended recurrent network for video captioning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8339–8348 (2019)
11. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1179–1195 (2017)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
13. Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., Shen, H.T.: From deterministic to generative: multimodal stochastic RNNs for video captioning. IEEE Trans. Neural Netw. Learn. Syst. **30**(10), 3047–3058 (2019)
14. Song, J., Li, X., Gao, L., Shen, H.T.: Hierarchical LSTMs with adaptive attention for visual captioning. CoRR abs/1812.11004 (2018)
15. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision (2015)
16. Vaswani, A., et al.: Attention is all you need. CoRR abs/1706.03762 (2017)
17. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence - video to text. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4534–4542 (2015)
18. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. CoRR abs/1411.4555 (2014)
19. Wang, J., Wang, W., Huang, Y., Wang, L., Tan, T.: M3: multimodal memory modelling for video captioning. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7512–7520 (2018)
20. Xu, H., Venugopalan, S., Ramanishka, V., Rohrbach, M., Saenko, K.: A multi-scale multiple instance video description network (2015)
21. Xu, N., Liu, A., Nie, W., Su, Y.: Attention-in-attention networks for surveillance video understanding in internet of things. IEEE Internet Things J. **5**(5), 3419–3429 (2018)
22. Xu, Y., Han, Y., Hong, R., Tian, Q.: Sequential video VLAD: training the aggregation locally and temporally. IEEE Trans. Image Process. **27**(10), 4933–4944 (2018)
23. Yang, Y., et al.: Video captioning by adversarial LSTM. IEEE Trans. Image Process. **27**(11), 5600–5611 (2018)
24. Yao, L., et al.: Describing videos by exploiting temporal structure. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4507–4515 (2015)
25. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. CoRR abs/1603.03925 (2016)
26. Zheng, Q., Wang, C., Tao, D.: Syntax-aware action targeting for video captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020
27. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. CoRR abs/1707.07012 (2017)