Journal of Big Data

RESEARCH Open Access

Semantic context driven language descriptions of videos using deep neural network

Dinesh Naik* and C. D. Jaidhar

*Correspondence:
din_nk@nitk.edu.in
Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India

Abstract

The massive addition of data to the internet in text, images, and videos made computer vision-based tasks challenging in the big data domain. Recent exploration of video data and progress in visual information captioning has been an arduous task in computer vision. Visual captioning is attributable to integrating visual information with natural language descriptions. This paper proposes an encoder-decoder framework with a 2D-Convolutional Neural Network (CNN) model and layered Long Short Term Memory (LSTM) as the encoder and an LSTM model integrated with an attention mechanism working as the decoder with a hybrid loss function. Visual feature vectors extracted from the video frames using a 2D-CNN model capture spatial features. Specifically, the visual feature vectors are fed into the layered LSTM to capture the temporal information. The attention mechanism enables the decoder to perceive and focus on relevant objects and correlate the visual context and language content for producing semantically correct captions. The visual features and GloVe word embeddings are input into the decoder to generate natural semantic descriptions for the videos. The performance of the proposed framework is evaluated on the video captioning benchmark dataset Microsoft Video Description (MSVD) using various well-known evaluation metrics. The experimental findings indicate that the suggested framework outperforms state-of-the-art techniques. Compared to the state-of-the-art research methods, the proposed model significantly increased all measures, B@1, B@2, B@3, B@4, METEOR, and CIDEr, with the score of 78.4, 64.8, 54.2, and 43.7, 32.3, and 70.7, respectively. The progression in all scores indicates a more excellent grasp of the context of the inputs, which results in more accurate caption prediction.

Keywords: Attention, Computer vision, Convolutional neural network, LSTM, Video captioning

Introduction

Due to the rapid improvement in low-cost camera technology, there is explosive growth in the number of images/videos. For humans, it is easy to understand the visual content and give a description of it. But for the machines, it is a complex task until it learns completely. So recently, research on image and video captioning grabbed the attention of the research community. The computer vision-based automation of

the big data domain with advanced deep learnings is essential. The works in [1–3] have shown those data domains can also be processed with the help of different deep learning-based approaches.

Image/video captioning aims to automatically provide a meaningful and appropriate natural language description of the target image/video's visual content. The challenges in this are manifold. It needs to recognize the objects and their interaction in the sequence of images and arrange words from natural language sequentially.

Video captioning [4–6] is a task of automatically describing the video content using natural language. It is easy for humans to watch a brief video clip and give an appropriate caption to it. Nevertheless, it is quite difficult for machines to do it as they have to read raw pixel data, process, and generate a fitting caption to the video clip without semantic errors. It has various applications, such as video comprehension, text-based video retrieval accessibility for blind users, and multimedia recommendation. The exponential growth and research advances triggered the quick shift in the deep CNN domain, and other optimal variants prominently contributed to image captioning. However, due to the complexity of understanding the diverse sets of objects, and their relations, video captioning is usually a different task as compared to image captioning. In spite of different research issues in visual captioning, few attempts are mainly supported by recent technology like LSTM for more extended word prediction, Recurrent Neural Network (RNN), Gated Recurrent Units (GRU) [7, 8]. All natural language processing tools are very much efficient in some aspects. The LSTM model overcomes the problem of vanishing gradients and exploding problem by allowing the model to learn and update hidden states. The recent advancement in Natural Language Processing (NLP) has made machines understand that very efficiently with different word representations and used in many related areas.

Though the basic LSTM and its variants have been adapted to various other applications, the efficacy is always explorable and debatable in conjunction with a visual neural network. Thus, this work proposed to use stacked LSTM to generate a visual sentence. The past attempt [9] involves a deep visual model as a layered LSTM network.

The explicit non-usage of semantic attributes of different frames in visual attention models [6] for video captioning may be non addressed challenging area. For example, some words (i.e., "boy," "is," and "running"), here "boy" and "running" belong to visual words. In contrast, another word (i.e., "is",) a non-visual word, which requires no visuals but language context. The shortfall of current visual attention models [10] that generate non-semantic and non-contextual captions mislead visual understanding. The video captioning model requires a semantic correlation between visual contents and generated words, which has not been simultaneously considered in present models.

The widespread application of the LSTM based network with different contextual attention seen in many sequence generation tasks [4]. The success of any standard neural networks' is attributed to layering approach and each layer leverages contextual features. Each layer learns some essential features of the visual frames and passes this contextual information to the next higher-level layers. However, most of the necessary and existing visual information captioning methods utilize various LSTMs with a single layer. The very well developed and efficient deep CNN is able to capture the contextual spatial features in the image frames of the video clips on any scale.

The research community's recent trend is to find and experiment with the different visual attention mechanisms [11, 12] with different decoder architecture. This method succeeded in recent years. Specifically, this visual attention method is first used in visual captioning effectively [13]. It explains that the specific objects in an image are highlighted and extracted with visual contexts. These visual features are fused with weights giving more attention to the specific regions that are interested. The process of attention confronts the human behaviour of focus on the most exciting features of any visuals.

The sequence-based attention mechanism [14] prepares a fixed dimensional relevant feature vector. The method has become the most featured method on deep neural networks used for machine language translation, visual content identification, and question answering recommender-based systems. In this approach, we have used attention as a hidden layer that determines an unambiguous feature distribution to make a soft attention [15] selection over a fixed source feature vector.

The emphasis of attention is mostly on resolving the many difficulties outlined. To begin, close the gap between video frame visuals and summarize them to obtain a linguistic description suitable for encoding and decoding. By honing a well-tuned attention mechanism on the decoding model, one can forecast the most accurate and closely related language descriptions. Second, the semantic context of words and image regions is efficiently extracted and assigned to the feature vector production process. The suggested approach thoroughly investigates the temporal information contained in the entire sequence of video frames. Additionally, an effort is made at the decoding step to create an efficient joint model by merging stacked LSTM, CNN features, and an attention mechanism.

To the extent, the proposed framework's primary contributions are: Firstly, using GloVe word embedding [16], specifically used 100-dimensional GloVe vectors depending on the size of the vocabulary in the dataset. Secondly, a layered LSTM encoder experiment combined with visual Feature Extractor networks is used to extract temporal information from videos in order to comprehend their actions. Thirdly, a hybrid loss function is applied to bridge the gap between the semantic context of the video and word prediction. Finally, evaluate the proposed framework's efficacy using eight widely used performance evaluation measures.

The paper's remaining part is organized as follows: Section "Related works" explores existing works on captioning videos. Section "Proposed methodology" explains various modules involved in the proposed framework. The detailed explanations about experiments conducted and the results for video captioning are in Section "Experimental results and analysis". Section "Conclusions" concludes the paper with future thinking.

Related works

Generating captions for videos and images gained momentum in recent years. The advances in this area are manifold, and the current works in this direction have shown promising results. This trend has become widespread and a hot research topic. This section discusses related work on image and video captioning.

Several approaches [17] proposed to interpret the image's visual contents and to generate natural language descriptions. However, all of them lack attention mechanisms. In [18] an end-to-end deep neural network automatically learns the visual content of an

image and generates a corresponding sentence using RNN. Because RNN exhibits vanishing gradients, a still better alternative like LSTM would increase the model efficiency. Giving image description with semantic context [19] combines two strategies to abstract the image's deeper level information and combines with a decoder that can appropriately select rich semantic associations. The work in [20] suggests a method that learns to extract specific information in the image and guides a decoder model to generate text descriptions. However, these approaches utilized only available image content for captioning, which does not include any temporal information.

A video is a sequence of frames, so it includes temporal information too. The success of several image captioning approaches allured researchers to focus on temporal information available in the sequence of frames of a video and generating a suitable description or caption for the visual content. Bin et al. [4] used two layered LSTM for the visual encoding and natural-language generation. Their stacked global temporal structure in video clips is achieved by encoding video sequences with a forward and backward directional LSTM network and attributing attention to the original neural network structures. However, they paid attention only to the CNN features, not the embedding semantics. The research articles [21–26] used a variety of machine learning algorithms to retrieve significant characteristics and reduce the dimension of high-dimensional datasets for the purpose of classification.

The two-stage training method in Olivastri et al. [5], wherein the first-stage, the architecture is pre-trained with encoders and decoders. The second-stage trained the entire network to learn the most appropriate video visual captioning features end-to-end manner. The visual attention module in [6] selectively picks most related frames and the appropriate regions in each frame. The method also suggested an attention module to focus on the most similar phrases to exploit more accurate text descriptions. However, it [6] used two attention modules. A method which follows hierarchical LSTMs with two-level abstraction proposed in [27] for captioning. Hossain et al. [28] presented two layered self-attention to obtain the words' diverse context in captions. The advantage of this method over the other is that it captures long-range dependencies of text sequence and less computation time. To guarantee the sentence description's semantic consistency and the visual video content, an attention mechanism with a local two-dimensional encoder and LSTM decoder to map the visual and textual features into a joint space is suggested in [29].

The approach in [30] uses a soft attention mechanism with a dynamic spatial attention mechanism to consider the spatial context of the image regions. The [31] proposes to encode an input video sequence to output shot sequences. In this method, the LSTM part has supplemented with two additive and multiplicative objective functions. The idea in [32] is to mine and construct multitask attributes from the human captioned videos by learning models with CNN and RNN. Xu et al. [13] combined the Vector of Locally Aggregated Descriptor (VLAD) and the Recurrent Convolution Networks (RCNs) framework to develop a sequential layer called Sequential Vector of Locally Aggregated Descriptor (SeqVLAD), which generates a better representation of video. Gao et al. [7] proposed a neural architecture with an LSTM decoder, attention, and new loss function.

A bidirectional LSTM (BiLSTM) tried to exploit global features of videos in [8, 33] as most of the existing methods only capture local temporal information. Song et al. [34]

proposed a novel captioning framework for videos, which combines two-directional LSTM and an attention layer to generate better representations across entire video frames. Multiple encoder attention and fusion module explored in [35]. This approach adaptively learn the salient features. Yang et al. [8] has proposed standard Generative Adversarial Network (GAN) architecture. This standard approach with generator and discriminator maintain a balance between texts generated and the accuracy. The discriminator part works as an "adversary" to the generator. To efficiently use all the advancements in advanced machine learning, primarily to address the research gaps in image understanding and caption, we proposed a new framework that could better handle the problems. The proposed framework bettered some of the state-of-art methods.

Proposed methodology

Proposed framework

The proposed framework consists of encoder-decoder for generating an appropriate caption for a video as shown in Fig. 1. The pre-processing stage focuses on preparing the image frames derived from the input video to match the dimension requirements of pre-trained CNN. The visual encoder combines CNN-based visual features and the stacked LSTM. The decoder part is defined as a combination of attention and a single LSTM layer. To select the significant features, the Soft Attention has been used.

The advantage of NASNet and stacked LSTM is that a varying number of convolutional cells and the number of filters in the convolutional cells yields better accuracy than the traditional methods. Another view is that nonlinearity, and careful selection of connections among neurons together add to better results. Although two stages search the feature space created by two types of cells, stacked LSTM predicts the best captions.

The deeper layers in stacked LSTM are understood to combine the learned representation from previous layers to create new representations at high levels of abstraction. This adding depth is a type of representation optimizations.

In this proposed framework, the visual language model called encoder-decoder was used with NASNet [36], InceptionV3 [37], VGG16 [38]. The decoder consists of the attention mechanism to address long sequences in machine translation-this action of selectively concentrating on a relevant word to be predicted while ignoring others in

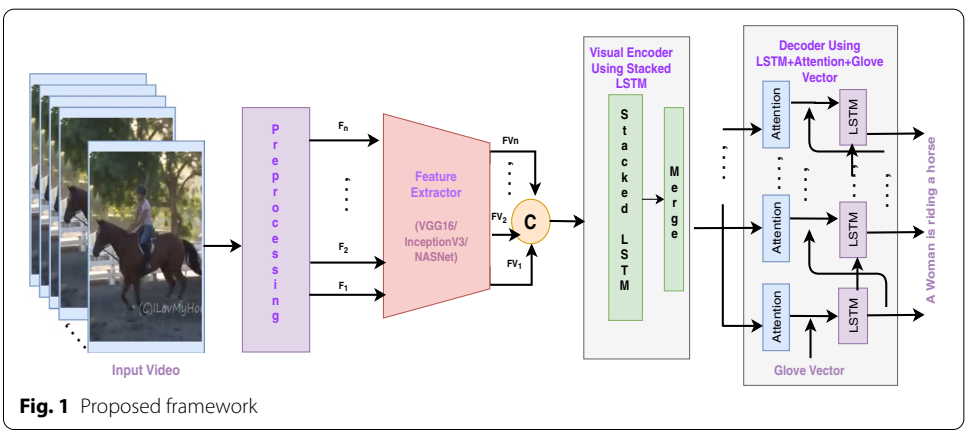

Fig. 1 Proposed framework

succession. Each sub-components of the proposed framework is described in the following section of the paper.

Preprocessing

In the preprocessing stage, extraction of frames are done. The extracted frames are resized to meet the input dimensions of deep learning models, namely VGG16, NASNet, and InceptionV3.

Feature extraction

Primarily, feature vectors are extracted, which are a high-level representation of videos, using three distinct models with varying dimensions.

NASNet large

The NASNet is a convolutional network originally used for image captioning. It takes 331×331 image size as input and resulting feature vector dimension of 4032 per frame. The NASNet architecture is defined as the blocks or cells, and these blocks are defined as the feature map with normal and reduction dimension. The blocks are called as Normal Blocks and Reduction Blocks. The Normal Block usually measures the feature map from the respective layer, and the Reduction Cell/Block reduces the feature map by a factor of 2. The controller decoder finds these Normal and Reductions Blocks information.

InceptionV3

Google developed this deep learning architecture for image captioning. The input image size should be 299×299. It results in a vector of dimension 2048 per frame. This model consisting of an "inception cell" working in parallel and then ultimately give the concatenated results. The kernel size in this model uses 1×1 convolutions to reduce the input channel depth. Each cell consists of different kernels with 1×1, 3×3, 5×5 dimensions, which learn to extract features from the input. Max pooling and padding is used to retain the dimensions for concatenation.

VGG16

Oxford developed VGG16 deep neural network. It takes an input image of size 224×224 pixels. The output feature vector is of size 4096. This deep neural network's advantage is using a small receptive field with a kernel size 3×3 dimension. The smallest possible size kernel captures the abstract information within frames through traversal all along the image grid's directions. The potential smaller values as the kernel with 11 dimensions act as a linear transformation of the input. The process is followed by a ReLU unit.

Given a video as a sequence of frames $V = \{F_1, F_2, ..., F_n\}$, where the video V has n frames and F_i represents ith frame of the video. The Feature Extractor generates set of feature vectors $FV = \{FV_1, FV_2, ..., FV_n\}$.

Visual encoder

The visual encoder is a stacked/layered approach. Visual features are further processed using stacked LSTM to capture temporal information. LSTM units' output is merged and then send to the decoder.

Single LSTM unit

The LSTM Network introduced in [39]. Architecture of single LSTM unit based on [40] is given in Fig. 2, and relation is defined in Eq. (1).

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ \widetilde{C_t} &= \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\ C_t &= f_t * C_{t-1} + i_t * \widetilde{C_t} \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{1}$$

Where, i_t, f_t, and o_t denotes input, forget, and output gates respectively. x_t, C_t, and h_t represent the current input, cell state, and hidden states, respectively. C_{t-1} and h_{t-1} are the input from preceding timestep. The symbol * represents the element wise multiplication. W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xo}, W_{ho}, W_{xg}, W_{hg}, b_i, b_f, b_o, and b_g are the parameters.

Stacked LSTM with dropout

We introduce a stacked LSTM visual encoder to encode the spatial CNN feature vectors and to exploit temporal information. To improve Deep learning model performance and avoid overfitting, the dropout is used on the feature vectors to randomly switch off few cells during the training. We used multiple layers of LSTM and finally, the output of layers are merged, and the result is given to the next layers. The output of layer i is defined as in Eq. (2).

$$o_t^{(i)}, h_t^{(i)} = LSTM^{(i)}(x_t, h_{t-1}^{(i)}) \tag{2}$$

The output of each previous LSTM layers are concatenated to obtain the output vector $o_t^{(f)}$ of the encoder as shown in Eq. (3).

$$o_t^{(f)} = \sum_{t=0}^{n} o_t^{(1)} + o_{n-t}^{(2)} \tag{3}$$

The proposed framework of the stacked LSTM unit is given in Fig. 3. The network length is the measure of the time span of a training set. The h_t, c_t and x_t denotes the output of last moment, current cell state and, current input respectively. The experimental result

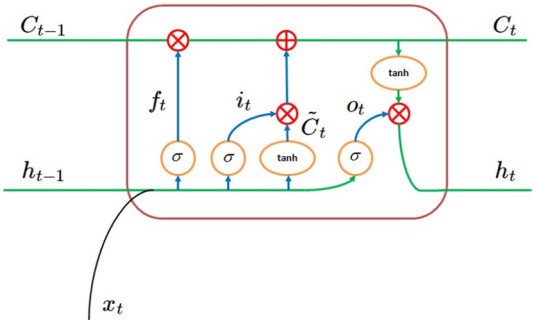

Fig. 2 LSTM unit [40]

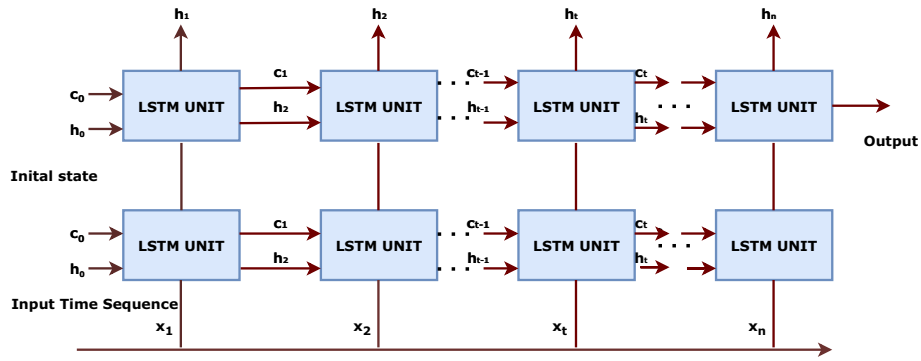

Fig. 3 Proposed framework of stacked/layered LSTM unit

showed that 2-layered LSTM with combinations of NASNet, attention, and embedding is better than 3-layered LSTM. Though the 3-layered LSTM seems better in abstract representation, overall better performance has resulted in 2-layered stacked LSTM because of different combinations.

Decoder

The decoder takes the feature vector from the encoder and utilized give best match to the original input using attention and GloVe vectors.

Attention mechanism

The output context vector from the encoder is fed to the decoder and generates a sequence of words describing the video. Training the model and giving the input sequence with a very long text sequence is not good. This sort of single, less contextual information from the encoder does not give the decoder excellent semantic and specific information. The attention approach gives more contextual meaning to the used decoder. The decoder learned how much semantic "attention" it should give to each input word at every decoding step.

The encoder output ($o_t^{(f)}$) fed to the decoder which is more contextually meaningful. The decoder's last hidden state and encoder hidden states are combined to calculate attention weights. A feed-forward neural network learns these weights.

The value for context vector c_i for the output word y_i is determined using Eq. (4).

$$c_i = \sum_{j=1}^{n} \alpha_{ij} o_j^{(f)} \tag{4}$$

The value for weights α_{ij} is computed by using a standard softmax function given by the Eq. (5).

$$\alpha_{ij} = exp(e_{ij}) / \sum_{k=1}^{n} exp(e_{ik}) \tag{5}$$

e_{ij} is the calculated output score for the input at j and output at i using Eq. (6).

$$e_{ij} = a(s_{i-1}, o_j^{(f)}) \tag{6}$$

Attention based LSTM

The decoder is an attention-based LSTM network. Attention mechanism combined with LSTM to focus on input sequence when predicting specific output sequence with more contextual understanding. Hence, the proposed decoder attention helps in selecting salient features for producing output sequence using a layer of LSTM.

In the proposed framework, every word in the caption encoded using GloVe. The vector representation model GloVe is used as an unsupervised learning technique to enable word representations of the given input word sequence. This model's training stage gives a cumulative global word-word co-occurrence representation from an input word corpus. The resulting vector depicts more informative and exciting linear structures of the word vector space. These embeddings are passed to the last layer to generate the sequence.

The previous hidden states h_{t-1}, the previous predicted word w_{t-1}, and the present context vector are combined to form the LSTM's [9] hidden state. During each time step context vector is adjusted so that decoder selectively attends the input sequence. Hence, the output of the decoder is given by Eq. (7).

$$o_t, h_t = LSTM([w_{t-1} + Attention[h_{t-1}; o_{t-1}^{(f)}]], h_{t-1}) \tag{7}$$

Loss functions

The approach used here is an attention-based LSTM. The main idea of using two loss functions is to ensure the contextual relationship between words generated and the semantic relations between the video features and the descriptions to be developed for the video's respective scene. The process maintains a simultaneous check between video translation and semantic efficiency.

Loss 1: translation from videos to words

Cross entropy loss is used for calculating the cost of translation and is given in Eq. (8).

$$Loss1 = -\frac{1}{N}\sum_{n}^{N} y \ln a + (1-y)\ln(1-a) \tag{8}$$

where, $N =$ represents number of training examples, y indicates actual values, and a denotes predicted values.

Loss 2: to bridge the semantic gap

Mean squared error loss helps bridge the semantic gap by estimating how far off the average predicted value is from the ground truth value. Thus, minimal value sees the close relationship between an estimated and actual value and ensures higher semantic similarity. The relation is defined in Eq. (9).

$$Loss2 = -\frac{1}{N}\sum_{n}^{N}\sum_{k}^{c}(y_k^n - a_k^n)^2 \tag{9}$$

where, N represents number of training examples, c indicates dimension of output vector, y denotes actual values, and a represents predicted values.

Combined loss

The combined loss measure is given in Eq. (10).

$$NewLoss = \lambda Loss1 + (1 - \lambda)Loss2 \tag{10}$$

Where, λ is a hyperparameter between 0 and 1.

Experimental results and analysis

Dataset

The experiments performed on Microsoft Video Description (MSVD) [41] dataset, which is a benchmark dataset for video captioning. The dataset consists of a total of 1970 short video clips from YouTube and 41 descriptions in English for each video. It also contains 80,000 clip-description pairs in different languages. For the proposed framework, English captions and dataset split in [9] used.

Experiments

Training parameters

Using the dataset split up mentioned earlier, proposed framework was trained for 600 epochs. The proposed framework performed well with the following training parameters: batch size = 128, learning rate = 0.001, and optimizer = Adam. Figure 4 is a plot of training loss against the number of epochs trained for 2-layered and 3-layered LSTM respectively. One can observe that the loss decreases drastically, up to 100 epochs, and then the decrease gradual. The proposed framework got stabilized between 400 and 600 epochs.

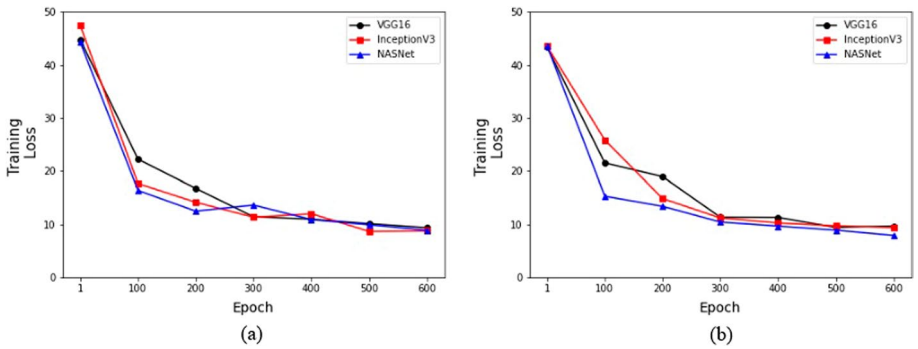

Fig. 4 Plot of training loss versus number of epochs.: **a** 2-layered LSTM **b** 3-layered LSTM

Sample results with built models

In this work, three models proposed based on the different pre-trained models for feature extraction in the visual encoder part. Proposed Model_1 utilizes VGG16 based visual Feature Extractor, Model_2 uses InceptionV3 to extract features, and Model_3 uses NASNet as Feature Extractor. All these models used GloVe embedding and attention in the decoding part.

Figure 5 shows some frames of five different test samples. The corresponding ground-truth and generated captions by three models are given in Table 1. The proposed model with VGG16 feature extractor performs poorly in catching the features of blurry images like Sample-5. Deeper networks, such as VGG16, InceptionV3, and NASNet, exhibit a slower decline in efficacy. This could be because those networks have a more complex structure, which gives them more room to learn attributes of

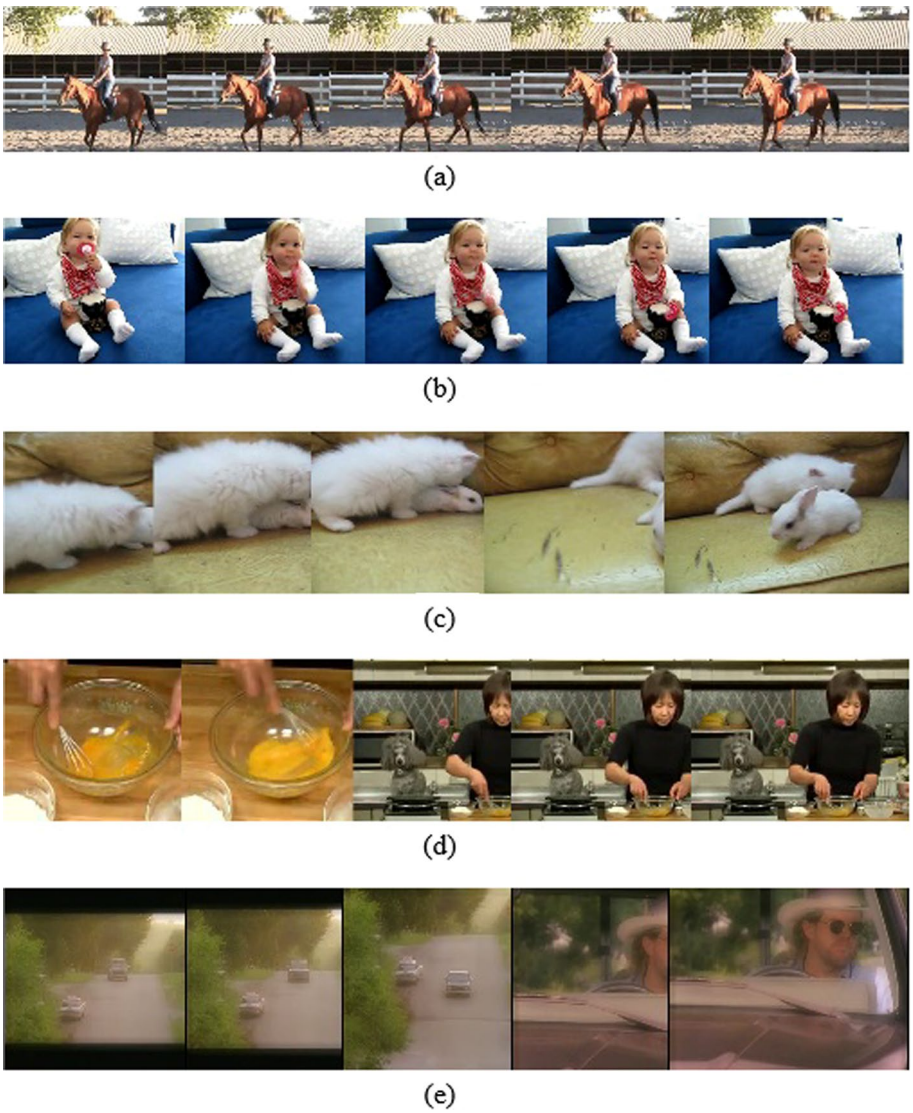

Fig. 5 Test samples: **a** Sample-1 **b** Sample-2 **c** Sample-3 **d** Sample-4 **e** Sample-5

Table 1 Sample input and output of proposed framework

Figure No.	Ground-truth	Model_1	Model_2	Model_3
Figure 5a	{A girl is riding a horse and jumping barriers}, { A girl is riding a horse}, {A woman is riding a horse in an outdoor arena and she makes a jump}, { A woman is riding a horse}, {A person riding a horse is jumping hurdles}.	{a man is riding a horse}.	{a man is riding a horse }.	{a girl is riding a horse}.
Figure 5b	{A baby is playing}, {A baby is playing with a pacifier}, {A baby plays on a bed}, {a baby is sucking on a soother and watching the camera}, {The toddler put the pacifier in and out of his mouth}.	{a man is slicing a potato}.	{a baby is playing with toys}.	{3a baby is playing with a camera}.
Figure 5c	{A cat is playing with a bunny}. ; {A cat is playing with a rabbit}, {A kitten is playing with a rabbit},{a white cat playing with a white bunny}, {A kitten and a rabbit are playing}.	{ a kitten is trying to climb a tree}.	{a cat is playing}.	{ a white kitten is playing}.
Figure 5d	{A woman is cutting something},{A woman cuts up some worms},{someone show how to prepare the japanese food},{a person coking},{ a women is making dish}.	{A woman is preparing a dish}	{The woman is mixing ingredients in a bowl}	{A woman is mixing some eggs}.
Figure 5e	{A person is driving a car},{ the man is drive the car on the road and seeing the place}, {Someone is driving a car}, {A car is driving down a road}, {A car is moving}.	{a man is running in the water}	{a man is driving a car}.	{a man is driving a car }.

the images that are unaffected by noise. The blurring, noise, or fogy produces a tiny shift in the filter responses in the primary convolutional layer. However, the penultimate convolutional layer exhibits significant variations in the filter responses. This modifies the first layer reaction, resulting in more or less significant alterations at the higher layer.

Obtained experimental results shows that Model_3 performed well in identifying the objects in the images and the semantic consistency than the other two models. For test samples in Fig. 5a and c, Model_3 gave captions close to the ground-truth captions than other two models. In test sample Fig. 5b, Model_1 identified a non-existing object. For test samples in Fig. 5d and e all the models gave almost similar captions to the ground-truth captions.

Comparison of models performance

Table 2 shows the BLEU performance metrics evaluated for the proposed framework with 2-layered and 3-layered LSTM on the MSVD dataset. The model with NASNet extracted features, GloVe, and 2-layered LSTM almost performed equally compared to the 3-layered NASNet model. But this NASNet model with 2-layered and 3-layered almost outperformed other proposed frameworks with VGG16 and InceptionV3 as NASNet identifies videos' objects more accurately with the help of more abstract representations from layered LSTMs.

In stacked LSTM, a level of abstractions of temporal input observations is also added. The GloVe represents words in n-dimensional space with unique meaning in each dimension. It captures a correlation between other words, which helps the stacked LSTM map from videos to descriptions correctly. The overall observations with this level of experiment and values conclude that Model_3 with a 2-layered approach is optimal and less costly, considering the BLEU metrics of all three models.

The suggested framework outperforms other current methodologies in terms of overall model performance. NASNet's LSTM and GloVe embedding are unusual in their two-layered structure. There are fewer floating-point operations and parameters in NASNets than in competing designs. To create a cell with the optimum performance, NASNet uses a controller RNN to identify the best combination of operations from a set of operations, rather than creating the block by hand. The input values to the network are fed through many levels of LSTM and propagate over time within a single LSTM cell with two layers of LSTM. Consequently, the parameters are well spread throughout several layers of the system. As a result, each time step has a complete set of inputs. While Word2Vec relies solely on local statistics (such as the context in which words are used), GloVe takes into account global data (such as the co-occurrence of terms) in order to produce word vectors.

To further validate the model's performance, we added an additional LSTM layer than the suggested framework, demonstrating the critical role of LSTM and its properties in producing superior outcomes to the two-layered strategy. As a result, proposed Model_3 outperformed the other two methods in the combinations indicated.

Table 3 shows the inappropriate predicted output, which is not very close to the ground truth for all three models, which has considered the 2-layer LSTM stack. Though the failure cases ascertain, Model_3 is slightly better in giving results than the other two models. In this, Figure 6 (a) depicts a sample input image featuring ground truth captions. The output is slightly near the ground truth with the combination of attention and the stacked LSTM in the proposed Model_3, due to the more

Table 2 Evaluation of 2-layered and 3-layered LSTM in proposed framework using BLEU metrics

Models	MSVD							
	2 Layer Stacked LSTM				3 Layer Stacked LSTM			
	B@1	B@2	B@3	B@4	B@1	B@2	B@3	B@4
VGG16 + Stacked LSTM + GloVe (Model_1)	69.1	50.1	38.2	27.0	68.1	48.8	37.0	25.58
InceptionV3 + Stacked LSTM + GloVe (Model_2)	74.3	60.1	49.7	40.2	73.6	59.8	49.5	38.5
NASNet + Stacked LSTM + GloVe (Model_3)	**78.4**	64.8	54.2	43.7	78.2	**65.3**	**55.1**	**44**

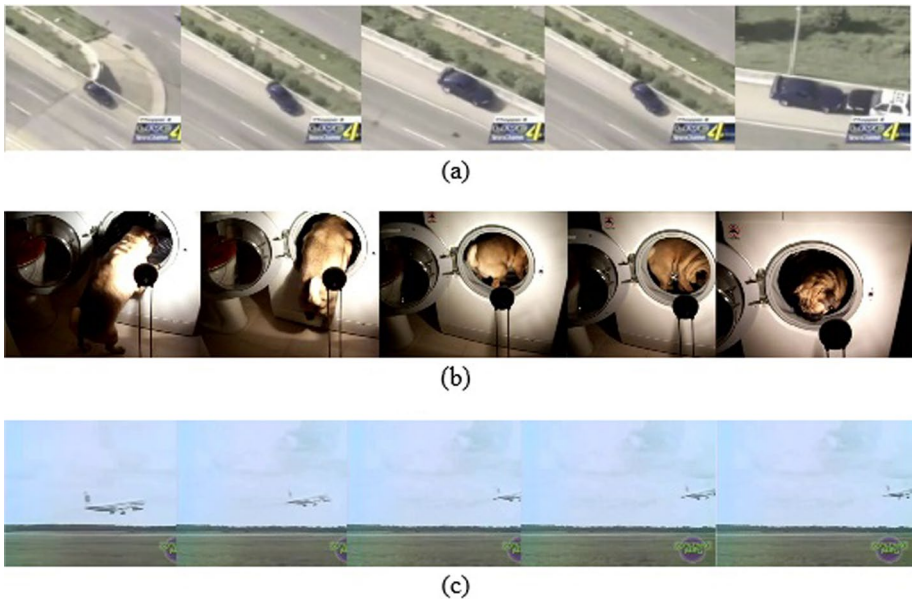

Fig. 6 Failure cases: **a** Sample-1 **b** Sample-2 **c** Sample-3

Table 3 Failure cases: sample input and output given by the framework

Figure No.	Ground-truth	Model_1	Model_2	Model_3
Figure 6a	{A car running from the police},{A guy is riding too fast in his bike.},{A man is driving backward and spins the car around.}	{A man is playing a guitar}	{A car is going up }.	{A car is chasing a car}.
Figure 6b	{A dog climbed into a clothes washing machine.}, {A bull dog is jumping into a washing machine.}, {The puppy went into the dryer.}, {The dog crawled into the dryer.}	{A man is putting some vegetables in a pan}	{A man is beating a concrete into a water}	{A man is making a fancy dish}.
Figure 6c	{ Airoplane in the Air}, {The plane took off from the runway.}, {An airplane is taking off.},{the person going on the airplane}	{A man is riding a bike}.	{A woman is pushing a rock}.	{A woman is running in the air }

in-depth learning of parameters without convergence and the increased focus on the required captions appropriate for the image locations. The model with NASNet has fewer parameters than the other conventional networks, but it makes the best use of the features to accurately predict over half of the ground truth words, outperforming some of the existing approaches. Model_1 fared poorly, as there were no matches because VGG16 entirely misread the words due to insufficient learning. Model_2 predicted the terms as accurately as Model_3 but with a better score than Model_1. This is due to the inception modules, composed of smaller filters, technically known as

pointwise convolutions, accompanied by convolutional layers with various filter sizes applied concurrently. This enables Inception networks to learn more complicated features and predict words with a high degree of accuracy compared to the ground truth. In Fig. 6b and c, all models failed to forecast accurately due to the complicated nature of the frames, which prevented them from learning all the features precisely due to an abundance of complex textures.

Table 4 provides performance achieved by the proposed three models with 2-layered and 3-layered stacked LSTM with four standard metrics mentioned and are compared. We observe 2-layered Model_3 result compared to the 3-layered counterpart outperforms the latter with efficient utilization of NASNet cells and connections. Though the results seem to be significantly closer considering all three models with different LSTM levels and different performance metrics, 2-layered proposed framework slightly have an edge on their 3-layered counterparts. In general, Model_3 consider being more efficient in utilizing inherent features of that models to give the best result.

While the additional strength garnered by the more deeper architecture in LSTMs is not fully understood theoretically, it has been observed empirically that deep RNNs may perform better than shallower ones on certain tasks and datasets. Generally, two layers of LSTMs have been demonstrated to be sufficient for detecting more complicated features [4, 42] . Additional layers make training more difficult due to increased layering results in information saturation, and increased complexity and also may lead to poor performance. As a result of our trials, it is clear that two-layered LSTM performed admirably across all measurement parameters.

Inception has inception layers and fewer parameters than VGG16, which is merely a simple array of convolutional max-pooling layers with dropouts added at the outset for speed optimization. Also, these dropouts effectively handle the model's overfitting issues by dynamically flipping connections with the activation layer. For regularisation, there is additionally an auxiliary classifier. A complicated collection of filters within a 'cell' can considerably improve outcomes in InceptionV3. The NASNet model outlines creating such a cell as an optimization process and then stacks numerous copies of the best cell to create a large network. NASNet has designed a new optimized architecture that employs a controller RNN module to choose the top-performing cells. As we see the unique combinations, all of these structural modules performed on the MSVD dataset more effectively.

Table 4 Evaluation of 2-layered and 3-layered LSTM in proposed framework using METEOR, ROUGE, CIDEr and SPICE

Models	**MSVD**							
	2 Layer Stacked LSTM				**3 Layer Stacked LSTM**			
	METEOR	**ROUGE**	**CIDEr**	**SPICE**	**METEOR**	**ROUGE**	**CIDEr**	**SPICE**
VGG16 + Stacked LSTM + GloVe (Model_1)	24.7	60.7	32.4	3	24.1	60.9	29.6	3
InceptionV3 + Stacked LSTM + GloVe (Model_2)	**33.3**	66.6	58.4	4.8	31.1	67.0	64.4	4.9
NASNet + Stacked LSTM + GloVe (Model_3)	32.3	**68.8**	70.7	**5.1**	31.8	**67.5**	**71.4**	4.9

When other indicators such as METEOR, ROUGE, CIDEr, and SPICE are evaluated, the proposed Model_3 with two-layered LSTM outperforms the ROUGE and SPICE scores. This implies that the SPICE score always takes the textual dataset's semantics into account, as well as the association of an additional attention layer in our model. Whereas ROUGE is similar to BLUE and has a higher score, it makes logical that Model_3 constantly outperforms all other measures, as we demonstrated in other measurements. The Model_3 with three LSTM layers outperformed the CIDEr score because the more abstract level of information learned by the third layer automatically captures better grammaticality, saliency, and accuracy.

Evaluation metrics and results

The performance of the proposed framework was evaluated on different metrics. The BLUE [43] algorithm evaluates the quality of the text, considered to be the matching between machines output and that of reference. The score is always between 0 and 1. This score indicates how similar the machines predicted output to that of reference with values closer to one representing more similar texts. The METEOR [44] is evaluating the machine translation output based on the harmonic mean of unigram precision and recall. The SPICE [45], is to alleviate the limitations of existing n-gram based metrics. This method uses the semantic propositional context component of caption evaluation. The CIDEr [46] metric measures the similarity of generated text against human-generated sentence. This measure uses grammaticality, saliency, and accuracy inherently captured. The ROUGE [47] is similar to BLUE compares predicted text with reference sentence.

Single loss vs hybrid loss

Figure 7 depicts the variation in BLUE4 and CIDEr metrics' performance while using single and hybrid loss during training. It is observed that the proposed NASNet Feature Extractor model performed well in the hybrid loss since single loss focus on only translation loss whereas hybrid loss, considers the semantic gap between video and captions. Hence, hybrid loss proved to work better for the proposed framework.

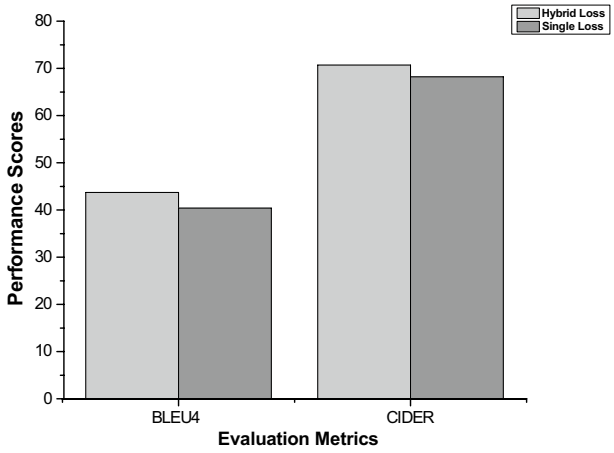

Fig. 7 Single loss versus hybrid loss

Table 5 Evaluation of Model_3 using different λ hyper parameter

Lambda Values	Model_3							
	BLUE score				Other metrics			
	B@1	B@2	B@3	B@4	METEOR	ROUGE	CIDEr	SPICE
$\lambda = 0.1$	78.4	64.8	54.2	43.7	32.3	68.8	70.7	5.1
$\lambda = 0.3$	76.1	62.1	50.9	39.7	31.3	66.9	67.5	4.9
$\lambda = 0.7$	74.9	60.7	49.7	38.9	30.9	66.6	63.3	4.8
$\lambda = 0.8$	75.2	61.4	50.7	40.3	31.2	67.1	66.9	5.0
$\lambda = 0.9$	75.2	61.3	50.4	39.9	30.8	66.6	64.1	4.9

Table 6 Comparison with existing methods

Method	B@1	B@2	B@3	B@4
S-VC [48]	–	–	–	35.1
SA [49]	–	–	–	40.3
MM-VDN [50]	–	–	–	37.6
LSTM-E [51]	74.9	60.9	50.6	40.2
HBNEVC [52]	–	–	–	42.5
LVMVP [53]	–	–	–	40.1
LSTM-GAN [8]	–	–	–	42.9
SE-GRU [54]	–	–	–	42.9
BPLSTM [55]	78.4	64.8	53.8	42.9
UTS [56]	–	–	–	43.00
STAT_LOC_V [10]	–	–	–	43.2
STAT_LOC_L [10]	–	–	–	42.9
p-RNN(VGGNet) [11]	77.3	64.5	**54.6**	**44.3**
Model_3 (Proposed)	**78.4**	**64.8**	**54.2**	**43.7**

Table 5 shows the different performance scores experimented for hyperparameter λ with different values. We observe different performance values are corresponding to the different λ values. The best performance score has resulted with $\lambda = 0.1$ for the proposed 2-layered Model_3.

The values of λ, one of the tweaking factors relating with hybrid loss. Overfitting occurs in any model primarily as a result of the model learning even the slightest details contained in the data. Thus, after learning all conceivable patterns, the model performs admirably on the training set however fails to deliver satisfactory results during the testing phase. It crumbles when confronted with previously unknown data. To avoid overfitting, the model's complexity should be reduced. This applies a regularisation parameter λ. As a result, comparatively simple models are less prone to overfitting than complicated models. In this context, a simple model is one in which the dispersion of hyperparameters has a low entropy, and hence many possibilities are attempted. We discovered that the optimal value is 0.1.

Comparison with existing works

Table 6 compares the obtained experimental results of the proposed NASNet Feature Extractor, Model_3 with some of the existing state-of-the-art video captioning works on

the MSVD dataset. The proposed NASNet model gave better results for B@1, B@2, B@3, and, B@4 metrics. The reason behind this is because BLEU score calculation searches for the same words in the text. The combination of NASNet with layered approach betters the representation and prediction.

Though the p-RNN [11] outperforms our Model_3 in B@3 and B@4 attributed to the fact that their RNN model is not compelled and video features are generally fed into the multilayer, our model outperforms all of the other approaches listed in Table 6 due to the inclusion of soft attention in the decoder and a contextual vector generated for the captions. The results in [55] are identical in B@1 and B@2 because the measure is a just lexical matching of words between reference[input sentence] and candidate sentence[predicted sentence].when it comes to B@3 and B@4 the proposed model is bettered due to the contextual understanding of preceding and succeeding words of the any target word. Moreover the proposed model is attached to an attention mechanism, which finds the lexical and semantics of the words surrounded whereas the paper [55] is not with attention mechanism.

Table 7 shows the obtained experimental results of the proposed framework using NASNet with some of the existing static frame-level approaches on video captioning works on the MSVD dataset. The proposed NASNet framework gave better results for METEOR metrics because it first compares tokens, synonyms, and paraphrases. Some of the existing baseline papers having multiple different features on the same video dataset.

Table 7 Comparison of proposed framework with the state-of-the-art methods w.r.t METEOR and CIDEr score

Method	METEOR	CIDEr
S-VC [48]	29.3	–
SA [49]	29.6	51.7
S2VT [57]	29.2	–
S2VT[VGGNet+Optical flow] [57]	29.8	–
MM-VDN [50]	29.0	–
MP-LSTM [9]	29.1	–
LSTM-E[VGGNet] [51]	29.5	–
LSTM-E[C3D] [51]	29.9	–
LSTM-E[VGGNet+C3D] [51]	31.0	–
LSTM-GAN [8]	30.4	–
p-RNN[C3D] [11]	30.3	–
p-RNN[VGGNet] [11]	31.1	–
LVMVP [53]	29.9	51.1
BPLSTM [55]	32.0	62.20
HRNE [12]	32.1	–
HBNEVC [52]	–	63.5
SE-GRU [54]	–	62.3
STAT [58]	–	67.5
MA-LSTM [29]	–	70.4
UTS [56]	33.20	71.10
STAT_LOC_V [10]	30.5	62.8
STAT_LOC_L [10]	31.0	62.5
Model_3 (Proposed)	32.3	70.7

We observe that 2-layered Model_3 shows better performance. Table 7 also compares the CIDEr metrics and the proposed framework gave better results over the existing works because CIDEr uses lengthier n-grams to capture the grammatical properties and higher semantics of the text.

Experimentation environment details

All studies are done on a machine configured with an Intel Core i7-10750H CPU running at 2.60GHz, 2592Mhz, six cores, twelve logical processors, sixteen gigabytes of RAM, and an NVIDIA GeForce GTX 1650 GPU. Keras with TensorFlow is used as the backend.

Advantages

Real-world applications like automatic video subtitling, surveillance footage, text-based video retrieval affordability for blind users, video comprehension, multimedia recommendation is made possible by video and image captioning advances. These include helping people with various degrees of vision disability, self-driving vehicles, sign interpretation, human-robot interaction, and intelligent video subtitling. Various 2D-CNN models are experimented with layered LSTM to obtain the suitable model to extract spatio-temporal features from the video in the proposed work. Also, the attention mechanism captures the contextual information to predict the best phrases for the videos. The experimental results have also proven the same and made the proposed approach practically applicable in various real-world scenarios mentioned above.

Limitations

While our approach is capable of producing a sentence for video and has demonstrated promising outcomes, it has significant drawbacks. The majority of our failures result in an inaccurate object name being used in phrases, for example, when small objects with similar shapes or appearances are confused. As a result, reliably finding correct objects in images, including those that are hazy or obscured, and anticipating associated captions would remain an open topic. Video and Sentential data goes unidirectionally down to the next level via the visual encoder. As a result, utilising the Bidirectional LSTM, erroneous information can still be eliminated. While we built a sentence vector with GloVe, the model can still incorporate the most recent embeddings such as BERT [59].

Conclusion

The proposed framework fully explores the spatial and temporal information among the video frames' whole sequence. In this paper, an efficient and new framework is proposed by integrating multiple LSTM, different Feature Extractors, Soft Attention, hybrid loss functions and GloVe embedding mechanism at the decoding stage. The visual encoder is a combination of CNN-based visual features and the layered LSTM. The decoder part is defined as a combination of attention and a single LSTM layer. To select the significant features, the Soft Attention has been used. This paper induced the hybrid loss to focus on semantic consistency.

Based on the experiments, the framework achieved approximately 24.5 % more than S-VC, 9% more than LSTM-E, and 2% more than BPLSTM in BLUE score criteria.

Further, the proposed technique outperformed SA by 9% and 36% in terms of METEOR and CIDEr, respectively. Thus, the suggested model outperformed the majority of current studies in terms of a variety of evaluation metrics.

In the future, we intend to update our model to work with domain-specific datasets, such as movies and documentaries, and to extend the architecture to incorporate Generative Adversarial Networks (GAN). Additionally, we would like to experiment with techniques such as beam search, which is used to determine the optimal word combination for a caption.

Acknowledgements
Not applicable.

Authors' contributions
Both authors contributed to design and implementation, analysis of results, and preparation of the manuscript. Both authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 19 September 2021 Accepted: 26 January 2022
Published online: 10 February 2022

References

1. Suryawati E, Pardede HF, Zilvan V, Ramdan A, Krisnandi D, Heryana A, Yuwana RS, Kusumo R, Arisal A, Supianto AA. Unsupervised feature learning-based encoder and adversarial networks. J Big Data. 2021;8(1):1–17. https://doi.org/10.1186/s40537-021-00508-9.
2. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021;8(1):1–74. https://doi.org/10.1186/s40537-021-00444-8.
3. Sampath V, Maurtua I, Martín JJA, Gutierrez A. A survey on generative adversarial networks for imbalance problems in computer vision tasks. J Big Data. 2021;8(1):1–59. https://doi.org/10.1186/s40537-021-00414-0.
4. Bin Y, Yang Y, Shen F, Xie N, Shen HT, Li X. Describing video with attention-based bidirectional LSTM. IEEE Trans Cybern. 2019;49(7):2631–41. https://doi.org/10.1109/TCYB.2018.2831447.
5. Olivastri S, Singh G, Cuzzolin F. End-to-end video captioning. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 1474–1482, 2019. https://doi.org/10.1109/ICCVW.2019.00185.
6. Zhao B, Li X, Lu X. CAM-RNN: co-attention model based RNN for video captioning. IEEE Trans Image Process. 2019;28(11):5552–65. https://doi.org/10.1109/TIP.2019.2916757.
7. Gao L, Guo Z, Zhang H, Xu X, Shen HT. Video captioning with attention-based LSTM and semantic consistency. IEEE Trans Multimedia. 2017;19(9):2045–55. https://doi.org/10.1109/TMM.2017.2729019.
8. Yang Y, Zhou J, Ai J, Bin Y, Hanjalic A, Shen HT, Ji Y. Video captioning by adversarial LSTM. IEEE Trans Image Process. 2018;27(11):5600–11. https://doi.org/10.1109/TIP.2018.2855422.
9. Venugopalan S, Xu H, Donahue J, Rohrbach M, Mooney RJ, Saenko K. Translating videos to natural language using deep recurrent neural networks. CoRR 2014. https://doi.org/10.3115/v1/N15-1173. arXiv:1412.4729.
10. Yan C, Tu Y, Wang X, Zhang Y, Hao X, Zhang Y, Dai Q. STAT: spatial-temporal attention mechanism for video captioning. IEEE Trans Multimedia. 2020;22(1):229–41. https://doi.org/10.1109/TMM.2019.2924576.
11. Yu H, Wang J, Huang Z, Yang Y, Xu W. Video paragraph captioning using hierarchical recurrent neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4584–4593, 2016. https://doi.org/10.1109/CVPR.2016.496.
12. Pan P, Xu Z, Yang Y, Wu F, Zhuang Y. Hierarchical recurrent neural encoder for video representation with application to captioning. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1029–1038, 2016. https://doi.org/10.1109/CVPR.2016.117.

13. Xu Y, Han Y, Hong R, Tian Q. Sequential video VLAD: training the aggregation locally and temporally. IEEE Trans Image Process. 2018;27(10):4933–44. https://doi.org/10.1109/TIP.2018.2846664.
14. Amaresh M, Chitrakala S. Video captioning using deep learning: an overview of methods, datasets and metrics. In: 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0656–0661, 2019. https://doi.org/10.1109/ICCSP.2019.8698097.
15. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2016. arXiv:1409.0473.
16. Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar, 2014. https://doi.org/10.3115/v1/D14-1162.
17. Jing Y, Zhiwei X, Guanglai G. Context-driven image caption with global semantic relations of the named entities. IEEE Access. 2020;8:143584–94. https://doi.org/10.1109/ACCESS.2020.3013321.
18. Vinyals O, Toshev A, Bengio S, Erhan D. Show and tell: a neural image caption generator. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3156–3164, 2015. https://doi.org/10.1109/CVPR.2015.7298935.
19. You Q, Jin H, Wang Z, Fang C, Luo J. Image captioning with semantic attention. CoRR 2016. arXiv:1603.03925.
20. Fang H, Gupta S, Iandola F, Srivastava RK, Deng L, Dollár P, Gao J, He X, Mitchell M, Platt JC, Zitnick CL, Zweig G. From captions to visual concepts and back. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1473–1482, 2015. https://doi.org/10.1109/CVPR.2015.7298754.
21. Arowolo MO, Ogundokun RO, Misra S, Kadri AF, Aduragba TO. In: Garg, L., Chakraborty, C., Mahmoudi, S., Sohmen, V.S. (eds.) Machine learning approach using KPCA-SVMs for predicting COVID-19, pp. 193–209. Springer, Cham, 2022. https://doi.org/10.1007/978-3-030-72752-9_10.
22. Arowolo MO, Adebiyi MO, Nnodim CT, Abdulsalam SO, Adebiyi AA. An adaptive genetic algorithm with recursive feature elimination approach for predicting malaria vector gene expression data classification using support vector machine kernels. Walailak J Sci Technol (WJST). 2021;18(17):9849.
23. Arowolo MO, Adebiyi M, Adebiyi AA, OKesola J. Predicting RNA-Seq data using genetic algorithm and ensemble classification algorithms. Indonesian J Electr Eng Comput Sci. 2021;21(2):1073–81.
24. Arowolo MO, Adebiyi MO, Adebiyi AA, Olugbara O. Optimized hybrid heuristic based dimensionality reduction methods for malaria vector using KNN classifier. 2020.
25. Arowolo MO, Adebiyi MO, Adebiyi AA. Enhanced dimensionality reduction methods for classifying malaria vector dataset using decision tree. Sains Malaysiana. 2021;50(9):2579–89.
26. Adebiyi MO, Arowolo MO, Olugbara O. A genetic algorithm for prediction of RNA-seq malaria vector gene expression data classification using SVM kernels. Bull Electr Eng Inf. 2021;10(2):1071–9.
27. Gao L, Li X, Song J, Shen HT. Hierarchical LSTMs with adaptive attention for visual captioning. IEEE Trans Pattern Anal Mach Intell. 2020;42(5):1112–31. https://doi.org/10.1109/TPAMI.2019.2894139.
28. Hossain MZ, Sohel F, Shiratuddin MF, Laga H, Bennamoun M. Bi-SAN-CAP: Bi-directional self-attention for image captioning. In: 2019 Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7, 2019. https://doi.org/10.1109/DICTA47822.2019.8946003.
29. Xu J, Yao T, Zhang Y, Mei T. Learning multimodal attention LSTM networks for video captioning. In: Proceedings of the 25th ACM International Conference on Multimedia. MM '17, pp. 537–545. Association for Computing Machinery, New York. 2017. https://doi.org/10.1145/3123266.3123448.
30. Khademi M, Schulte O. Image caption generation with hierarchical contextual visual spatial attention. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2024–20248, 2018. https://doi.org/10.1109/CVPRW.2018.00260.
31. Ji Z, Xiong K, Pang Y, Li X. Video summarization with attention-based encoder-decoder networks. IEEE Trans Circuits Syst Video Technol. 2020;30(6):1709–17. https://doi.org/10.1109/TCSVT.2019.2904996.
32. Li L, Gong B. End-to-end video captioning with multitask reinforcement learning. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 339–348, 2019. https://doi.org/10.1109/WACV.2019.00042.
33. Li S, Tao Z, Li K, Fu Y. Visual to text: survey of image and video captioning. IEEE Trans Emerg Topics Comput Intel. 2019;3(4):297–312. https://doi.org/10.1109/TETCI.2019.2892755.
34. Song J, Li X, Gao L, Shen HT. Hierarchical LSTMs with adaptive attention for visual captioning. CoRR 2018. arXiv:1812.11004.
35. Xu N, Liu A, Nie W, Su Y. Attention-in-attention networks for surveillance video understanding in internet of things. IEEE Internet Things J. 2018;5(5):3419–29. https://doi.org/10.1109/JIOT.2017.2779865.
36. Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. CoRR 2017. arXiv:1707.07012.
37. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826, 2016. https://doi.org/10.1109/CVPR.2016.308.
38. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: The 3rd International Conference on Learning Representations (ICLR2015) 2015. arXiv:1409.1556.
39. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
40. Sha L, Chang B, Sui Z, Li S. Reading and thinking: Re-read LSTM unit for textual entailment recognition. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2870–2879. The COLING 2016 Organizing Committee, Osaka, 2016.
41. Chen D, Dolan WB. Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 190–200, 2011.
42. Salman AG, Heryadi Y, Abdurahman E, Suparta W. Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting. Proc Comput Sci. 2018;135:89–98.

43. Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia. 2002. https://doi.org/10.3115/1073083.1073135.
44. Banerjee S, Lavie A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation And/or Summarization, pp. 65–72. Association for Computational Linguistics, Ann Arbor, Michigan. 2005. https://aclanthology.org/W05-0909.
45. Anderson P, Fernando B, Johnson M, Gould S. Spice: Semantic propositional image caption evaluation. In: European Conference on Computer Vision, pp. 382–398, 2016. Springer.
46. Vedantam R, Zitnick CL, Parikh D. CIDEr: Consensus-based image description evaluation. CoRR 2014. arXiv:1411.5726.
47. Lin CY. ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81. Association for Computational Linguistics, Barcelona. 2004. https://www.aclweb.org/anthology/W04-1013.
48. 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 '15, pp. 1191–1194. Association for Computing Machinery, New York. 2015. https://doi.org/10.1145/2733373.2806314.
49. Yao L, Torabi A, Cho K, Ballas N, Pal C, Larochelle H, Courville A. Describing videos by exploiting temporal structure. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4507–4515, 2015. https://doi.org/10.1109/ICCV.2015.512.
50. Xu H, Venugopalan S, Ramanishka V, Rohrbach M, Saenko K. A multi-scale multiple instance video description network. CoRR 2015. arXiv:1505.05914.
51. Pan Y, Mei T, Yao T, Li H, Rui Y. Jointly modeling embedding and translation to bridge video and language. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4594–4602, 2016. https://doi.org/10.1109/CVPR.2016.497.
52. 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. https://doi.org/10.1109/CVPR.2017.339.
53. Nian F, Li T, Wang Y, Wu X, Ni B, Xu C. Learning explicit video attributes from mid-level representation for video captioning. Comput Vis Image Underst. 2017;163:126–38. https://doi.org/10.1016/j.cviu.2017.06.012. (**Language in Vision**).
54. Hao X, Zhou F, Li X. Scene-Edge GRU for Video Caption. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1, pp. 1290–1295, 2020. https://doi.org/10.1109/ITNEC48623.2020.9084781.
55. Nabati M, Behrad A. Video captioning using boosted and parallel Long Short-Term Memory networks. Comput Vis Image Underst. 2020;190:102840. https://doi.org/10.1016/j.cviu.2019.102840.
56. Sah S, Nguyen T, Ptucha R. Understanding temporal structure for video captioning. Pattern Anal Appl. 2020;23(1):147–59.
57. 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. https://doi.org/10.1109/ICCV.2015.515.
58. Tu Y, Zhang X, Liu B, Yan C. Video description with spatial-temporal attention. In: Proceedings of the 25th ACM International Conference on Multimedia. MM '17, pp. 1014–1022. Association for Computing Machinery, New York, 2017. https://doi.org/10.1145/3123266.3123354.
59. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, 2019. https://doi.org/10.18653/v1/N19-1423.

Publisher's Note