MDPI

Article

Scene Description with Context Information using Dense-LSTM

Varsha Singh [1,*] 0000-0003-4777-1278, PrakharAgrawal [1] and Uma Shanker Tiwary [1]

[1] Indian Institute of Information Technology, Allahabad; varshagaur@gmail.com; ust@iiita.ac.in
* Correspondence: varshagaur@gmail.com;

Abstract: Generating natural language description for Visual Content is a technique for describing the content available in the image(s). This technology has a vast range of applications combined with the Internet of Things (IoT). It requires knowledge of both the domains of computer vision and natural language processing. For this, various models with different approaches are suggested. One of them is encoder-decoder-based description generation. Existing papers used only objects for descriptions, but the relationship between them is equally important. Which in turn requires context information. For which technique like Long Short-Term Memory (LSTM) is required. This paper proposes an encoder-decoder-based methodology to generate human-like textual descriptions. Dense-LSTM is presented for better description as a decoder with modified VGG19 as an encoder to capture information to describe the scene. Standard datasets Flickr8K and Flickr30k are used for testing and training purposes. BLEU (Bilingual Evaluation Understudy) score is used to evaluate a generated text. For the proposed model, a GUI (Graphical User Interface) is developed, which produces the description for the output received and provides an interface for searching the related visual content and query-based search.

Keywords: Convolutional Neural Network (CNN); Dense-Long Short-Term Memory (Dense-LSTM); Internet of Things (IoT); Bilingual Evaluation Understudy Score (BLEU); Textual Description Generation

1. Introduction

1.1. Overview

In daily life, there is a lot of visual content through which we humans go, and as a human, it is a convenient task for us to interpret their meaning and usage. But for machines, detailed descriptions are required to understand the visual content.

Generating textual descriptions for explaining the context of visual content is a well-known area of artificial intelligence (AI). Identifying the scene type and objects in it that understand an image and its content requires both syntactic and semantic understanding for visual content as well as language[1].

Textual description for visual content has a wide range of applications. These applications drastically change or improve the way of living when combined with IoT devices. IoTs [2] are the objects or devices combined with either of these, like sensors, processing devices and other technologies to connect or exchange data. The proposed model can be used with any IoT-enabled technology like embedded systems, wireless sensor networks, cloud computing etc., depending on the required task.

In the proposed model, an encoder-decoder-based technique is used. Two neural networks are combined for a suitable description of the given visual content. The model works in two parts; one handles the visual content, and the other deals with the textual part. CNN is used as an encoder to extract features for the given visual content, and a vector is created for processing. VGG19 is used as an encoder with slight modification to get the desired dimensions. A novel Dense-LSTM is proposed as a decoder for the textual part. The existing feature extraction model took more time during training and had less promising results than the proposed model.

Citation: Singh, V.; Agrawal, P; Tiwary, U.S. Scene Description with Context Information using Dense-LSTM. *Journal Not Specified* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

In [3], descriptions are generated with tolerable efficiency and ResNet50 CNN is used for feature extraction. ResNet50 was one of the best performers, but they suffered from the vanishing gradient problem, which is sorted here using VGG19. VGG 19 shows comparatively better performance when time and space are considered during the training process.

The image is first preprocessed to convert into a $224 \times 224 \times 3$ dimension to pass through the encoder. Then, following the encoder-decoder translation method, the features are passed through the Dense-LSTM network. Paper [4] lacked on text generation part, for which LSTM network is used for enhancement. Using LSTM, better descriptions are generated as it can handle long-time information. In [43], Dense-LSTM is used to solve degradation problems and efficiently use the information in speech-emotion recognition. Therefore to upgrade the generated text quality, Dense-LSTM is used instead of simple LSTM or RNN in the proposed work. Here, Beam-search is used to opt for the better description generated by the decoder.

The basic concept is taken from the paper[9] in which they address that the textual descriptions could be improved using VGG19 as encoder and LSTM at the decoder part of the model. In [9] author showed promising outcomes on non-standard data. The proposed model further extends the same concept with changes using Dense-LSTM with a modified VGG19 model on a standard data set. The descriptions are preprocessed separately in the training set to develop a dictionary. For training purposes, the Flickr8k and Flickr30k datasets are used. A significant portion of such models is task-related to classifying images, which includes considerable complications in execution as identification of the content of visuals and objects is insufficient for such task. Identifying their relationship is equally important to generate a suitable human-like textual description. The main objective is efficiently producing textual descriptions in human-like language to get the semantics in the visual content for which Dense-LSTM is used.

1.2. Motivation

Many applications like image indexing, image editing and virtual assistance in computers and phones, etc., are the areas where text generation for visual content is used. While generating text for visual content, existing approaches use objects in an image, whereas the relation between them is equally important.Therefore, a novel Dense-LSTM is proposed to get the semantics in the visual content. When an image is posted on social media, the suggested tool helps predict the text for the content and offers emoticons according to the sentiment in the description. This tool can also generate descriptions in audio form so that it can help visually impaired/incapable people in their daily activities. It can help them understand the surroundings by taking video frames as input and generating descriptions of that frame when used with an IoT-enabled device, which can be directly transferable in audio form to that person.

Children give more attention to the visual content in comparison to the text. It helps in child education by providing the facility with an audio and textual description of the visual content to grasp more attention. Search engines like google are also used for such purposes. Still, google API is combined with the proposed model to give a more relevant description of the visual content, which is not available in a simple search. In the same way, when the proposed model is combined with an IoT-enabled device, the applications will get broader aspects and areas.

1.3. Related work

A lot of work has been done, and active research is going on in this area of textual description for visual content, as still there is a lot of scope for enhancement and addition. Much research in Image processing and Deep Learning focuses on IoT. One of them is using the same concept for IoT-enable visual content. In 1999, Ashton K[10] first proposed the term "Internet of Things". Elias et al. [11] use deep learning and IoT technology in their

work for wildlife. Kapoor A et al. [12] combined IoT and image processing to evaluate the plant's growth. Similarly, various application areas are still left untouched.

On the other side, various models are used to create a description for visual content. Here, the encoder-decoder-based approach is considered. In this approach, CNN, a Convolutional Neural Network, is taken as encoder and RNN, that is, Recurrent Neural Network as decoder are combined to address the textual description generation. As RNN lacks in storing information for longer, alternatives like LSTM, GRU etc., can also be used. LSTM (Long Short-term Memory) is a particular type of RNN with feedback connections. GRU (Gated Recurrent Unit), like LSTM with forgetting gate, and TNN (Temporal Neural Network), which works on low-level and high-level features, are existing alternatives to RNN. A good number of models like VGG, ResNet, Xception and AlexNet with their variations are available for encoding. Similarly, a good number of standard datasets like Flickr8K, which has 8k images, Flickr30K, with 30k pictures, MSCOCO with 80 object categories, and SUN dataset, etc., are available for description generation tasks.

Many researchers support the encoder-decoder model using CNN with LSTM. The proposed model follows the same approach. Two widely used models from Visual geometry group(VGG) OxfordNet with 16-layer model - **VGG16** and 19-layer model - **VGG19** which are used for feature extraction, are compared by Aung, San Pa, Win nwe, tin[9]. As per their results, in terms of accuracy, VGG-19 performs better. But as it has more layers, it took a little more space in memory compared to VGG-16. In [3], Chu, Yan Yue et al. show that ResNet50 and LSTM with a soft attention layer give considerably good results. Although, the problem faced in ResNet was Vanishing Gradient.

In the image-to-text generation field, LSTM is getting more attention among Computer Vision enthusiasts. In the LSTM model, some contextual cell states are there. Based on the requirement, these states behave like long-term or short-term memory cells. The possibility of better description generation using LSTM than RNN in understandable natural language is addressed by A Karpathy[14] in their research. They used an image dataset with their descriptions in natural language and checked for various correspondence of words with their description and information related to visual content. In this approach, the CNN model is used for feature extraction, and these features are used as raw data of an image. Words connectivity is done using contextual cells in LSTM to generate the description. Beam-search is used to select the most suitable description. An integrated model (CNN-LSTM) is developed in paper[5] to automatically view an image with appropriate description generation in the English language. As per the previous work shown in Table 1, most of the approaches used VGGNet, giving comparatively better results. Considering the same, VGGNet is used on the encoder part of the model with slight modification as per the required dimension for feature extraction.

Table 1. Related work in the same area

References	Image Encoder	Language Model
Rennie et al. [34]	ResNet	LSTM
Vsub et al. [35]	VGGNet	LSTM
Zhang et al. [36]	Inception-V3	LSTM
Wu et al. [37]	VGGNet	LSTM
Aneja et al. [38]	VGGNet	Language CNN
Wang et al. [39]	VGGNet	Language CNN

Table 2. Detailed comparison of similar approaches

Title	Authors	Dataset	Methodology	Metrics Used	Future Work
Automatic Image Captioning Based on ResNet50 and LSTM with Soft Attention (AICRL)	Yan Chu et al.[3] 2020	MSCOCO 2014	ResNet50 as encoder and LSTM as decoder	BLEU, METEROR, and CIDEr	—
Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning	Ning Xu et al. [4] 2020	MSCOCO and Flickr30k	CNN-RNN, Attention, Adaptive, and Stacked models	BLEU, METEOR, ROUGE, and CIDEr	Investigate the multi-agent algorithm to train the policy network for image captioning
Show, Attend and Tell: Neural Image Caption Generation with Visual Attention	Kelvin Xu et al. [5] 2015	Flickr8k, Flickr30k and MSCOCO	Oxford VGGnet as encoder and LSTM as decoder	BLEU and METEOR	Encoder-decoder approach with attention to different applications in other domains
"Factual" or "Emotional": Stylized Image Captioning with Adaptive Learning and Attention	Tianlang Chen et al. [6] 2018	FlickrStyle 10K, style captioning dataset: image sentiment captioning dataset based on MSCOCO	Encoder-decoder based stylized image captioning model Encoder as VGG-16 and ResNet152 and Decoder as LSTM	BLEU-1,2,3,4, ROUGE, CIDEr, METEOR	—
Image Captioning with Deep Bidirectional LSTMs	Cheg Wang et al. [13] 2016	Flickr8K, Flickr30K and MSCOCO	Deep CNN (AlexNet and 16-layer VggNet) and two separate LSTM (Bidirectional LSTM)	BLEU, METEOR and CIDEr	Incorporating multitask learning, attention mechanism and apply model to other sequence learning tasks: text recognition and video captioning
Image Captioning With Semantic Attention	Quanzeng You et al. [17] 2016	MSCOCO and Flickr30K	Combine top-down and bottom-up strategy with RNN	MSCOCO caption evaluation tool, BLEU, Meteor, Rouge-L and CIDEr	Phrase-based visual attribute with its distributed representations and new models for proposed semantic attention mechanism

[1] Table continued to next page.

BabyTalk: Understanding and Generating Simple Image Descriptions	Girish Kulkarni et al. [18] 2011	UIUC PASCAL sentence dataset (20 Pascal object categories)	Graph-based approach using conditional random field (CRF) and sequential tree re-weighted message passing (TRW-S) algorithm	Human subject-based evaluations, BLEU and ROUGE	To handle more image content (beyond the 20 Pascal object categories), produce more natural image descriptions, incorporating content in description, include actions, scenes, and describe the videos.
Deep image captioning using an ensemble of CNN and LSTM based deep neural networks	Alzubi, Jafar A et al. [19] 2021	Flickr8k and GloVe Embeddings dataset for vector representation of words	Custom ensemble model consisted of an Inception-V3 model as encoder and a 2-layer LSTM model as decoder	BLEU	Proposed to use Flickr30k and MsCOCO dataset.

This work mainly focuses on the decoder part responsible for description generation. A novel Dense-LSTM is proposed as a decoder with CNN instead of simple LSTM. Dense-LSTM is more suitable for utilizing information efficiently and as a solution to degradation[43]. By using this, description generation for the input visual content in natural language is done. Our prior work[42] was on the encoder part to get the features and possibly better description generation in terms of semantics. This work focused on the decoder to generate a more suitable description in terms of semantics and the implicit relationship between objects with context information. A GUI is also developed to use the model. Audio is also generated for the resulting textual description for visual content, and one can also search similar images for a given image and generated text. Table 2 provides a detailed comparison of the dataset and methodology with evaluation metrics.

2. Methodology

An encoder-decoder-based architecture is proposed for generating semantically correct textual descriptions. A novel "Dense-LSTM" architecture is proposed to describe the scene with context information. As Dense-LSTM is widely used for semantic extraction, in the proposed work, it is used to provide a more accurate description based on semantics. Some previous models [46,48] also used the Dense-LSTM with different architecture for different tasks. In [46], the authors compared the performance of their model on the original and augmented data. In [48], using frames, captions are generated for recognized action and Bi-directional LSTM is used. Similar work with a different approach is also presented in [47]. Encoding is done using a modified VGG19 CNN model, and Dense-LSTM is used for decoding. The probability distribution for each word in the vocabulary is considered for each word in the generated description. Then it is given to the decoder to transform them into a final description considered as the final output. One neuron for each word in output vocabulary and a softmax activation function is used on the encoder. VGG 19 is one of the variants of VGGNet, having 19 layers, out of which convolution layers are 16, and Fully connected layers are three with five MaxPool and one SoftMax layer.

$$Softmax(\overrightarrow{V_i}) = \frac{e^{v_i}}{\sum_{j=1}^{C} e^{v_j}} \quad (1)$$

where, V_i is input vector, V_j is output vector, e^{v_i} standard exponential function for V_i, e^{v_j} standard exponential function for V_j

The encoder does the task of image encoding to create feature vectors which are further given as input to the model for descriptions generation using Dense-LSTM. In the proposed architecture of Dense-LSTM, four layers of LSTM are used with two dense layers to enhance the resultant text. As the name "Dense" in Dense-LSTM suggests, each layer is densely connected with the other three layers. At each layer, five descriptions are generated, out of which best is selected using beam search. That best output is further given as input to the subsequent and successive layers, which improves the description semantically. The final generated description is passed to the dense layers to get the final output. In the encoder part of the proposed model, the last fully-connected layer is omitted to get the required dimensions. Beam search is a popular heuristic search that returns the list of most related sequences. Here, a novel Dense-LSTM is used, a variant of LSTM, and LSTM is one of the RNN forms [13][40]. In LSTM, the sigmoid gates group controls the reading and writing process. For different inputs, the updation of gates in LSTM takes place as follows:

$$g_{i_t} = sig(W_{xg_i} x_t + W_{hi} h_{t-1} + b_i) \quad (2)$$

$$g_{f_t} = sig\left(W_{xg_f} x_t + W_{hf} h_{t-1} + b_f\right) \quad (3)$$

$$g_{o_t} = sig(W_{xg_o} x_t + W_{ho} h_{t-1} + b_o) \quad (4)$$

$$G_t = phi(W_{xc_m} x_t + W_{hc} h_{t-1} + b_c) \quad (5)$$

$$c_{m_t} = g_{f_t} \odot c_{m_{t-1}} + g_{i_t} \odot G_t \quad (6)$$

$$h_t = g_{o_t} \odot \phi(c_{m_t}) \quad (7)$$

where, g_{i_t}, g_{f_t}, g_{o_t} are input, forget and output gates at time t and W, b, c_m, sig, $\odot$, and phi(ϕ) are weight matrices and bias vectors, memory gate, sigmoid activation function, products of gate values and hyperbolic tangent respectively.

In the proposed architecture of Dense-LSTM, all gates are updated according to the equations given from 2 to 7, except h_t, the hidden state. h_t of each LSTM unit is passed to the next unit and updated using the output gate and hyperbolic tangent of the memory gate to feed as input to the dense layer and further passed to another dense layer to get the desired output. Parameters used for the proposed work is shown in figure 1.

2.1. Model Architecture

Initially, visual content is used from the standard dataset for training purposes. Visual content is passed through the encoder to get the features in vector form. A $224 \times 224 \times 3$ image is passed to get output in 4096×1 dimensions. The last layer of the CNN model is removed to get the output in the desired shape. Feature vectors are generated in 4096×1 size through this process. All these vectors are saved in a separate file for feature extraction of each image during the training, testing, and validation process. Then, description pre-processing is done by eliminating the punctuation, single-letter, and alphanumeric words. With the vocabulary and word embeddings generated in the dataset, the maximum length of the description is obtained. In the case of Flickr8k and Flickr30k, the maximum description length are 34 and 75, respectively. Then word embedding with feature vectors is passed through the model. Five descriptions are generated for each image using LSTM with beam-search width=5. The final description is given by the final time-stamp, which is

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 75)	0	
embedding (Embedding)	(None, 75, 256)	4310528	input_2[0][0]
lstm (LSTM)	(None, 75, 256)	525312	embedding[0][0]
lstm_1 (LSTM)	(None, 75, 256)	525312	lstm[0][0]
lambda (Lambda)	(None, 75, 256)	0	lstm_1[0][0]
lambda_1 (Lambda)	(None, 75, 256)	0	lstm[0][0]
add (Add)	(None, 75, 256)	0	lambda[0][0] lambda_1[0][0]
lstm_2 (LSTM)	(None, 75, 256)	525312	add[0][0]
input_1 (InputLayer)	(None, 4096)	0	
lambda_2 (Lambda)	(None, 75, 256)	0	lstm_2[0][0]
lambda_3 (Lambda)	(None, 75, 256)	0	lstm_1[0][0]
lambda_4 (Lambda)	(None, 75, 256)	0	lstm[0][0]
dropout (Dropout)	(None, 4096)	0	input_1[0][0]
add_1 (Add)	(None, 75, 256)	0	lambda_2[0][0] lambda_3[0][0] lambda_4[0][0]
dense (Dense)	(None, 256)	1048832	dropout[0][0]
lstm_3 (LSTM)	(None, 256)	525312	add_1[0][0]
add_2 (Add)	(None, 256)	0	dense[0][0] lstm_3[0][0]
dense_1 (Dense)	(None, 256)	65792	add_2[0][0]
dense_2 (Dense)	(None, 16838)	4327366	dense_1[0][0]

Total params: 11,853,766
Trainable params: 11,853,766
Non-trainable params: 0

None

Figure 1. Layers of proposed decoder with dimension details

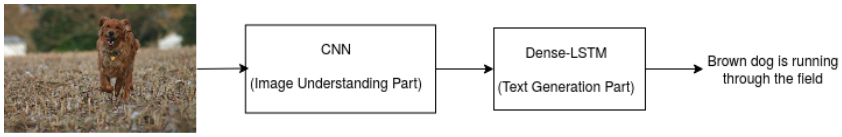

Figure 2. Block Diagram for Proposed Model

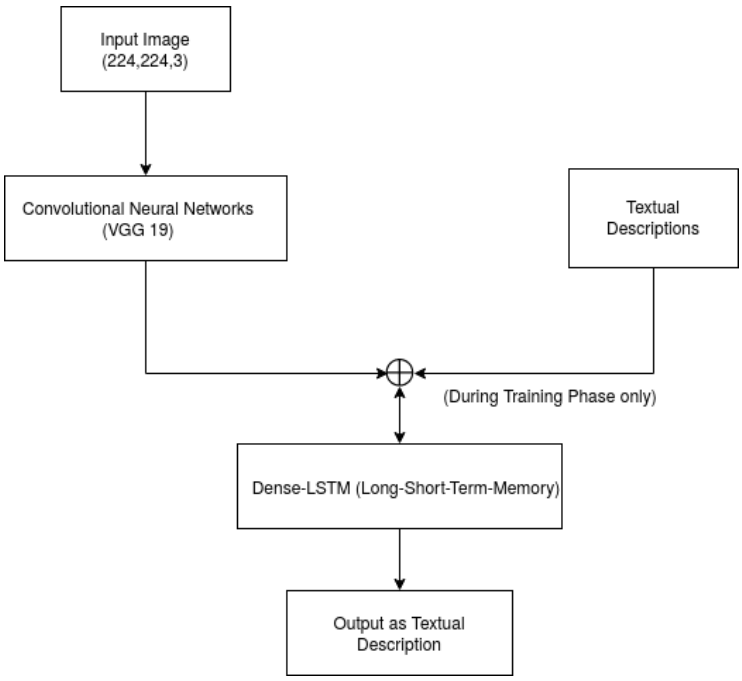

Figure 3. Architecture for Proposed Model

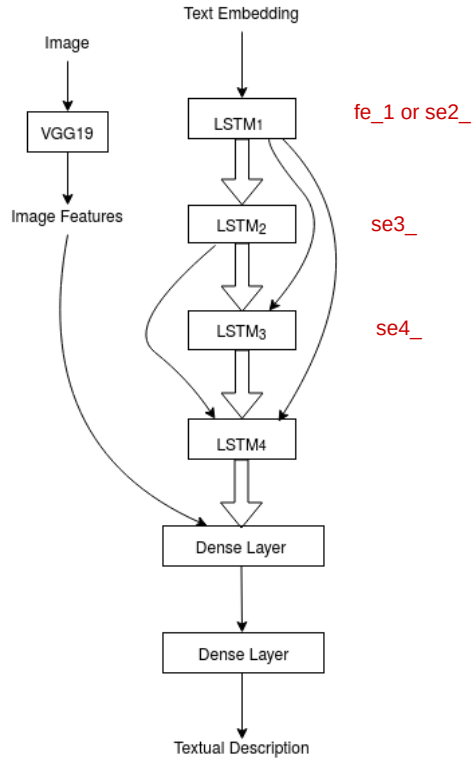

Figure 4. Proposed architecture of Dense-LSTM as decoder

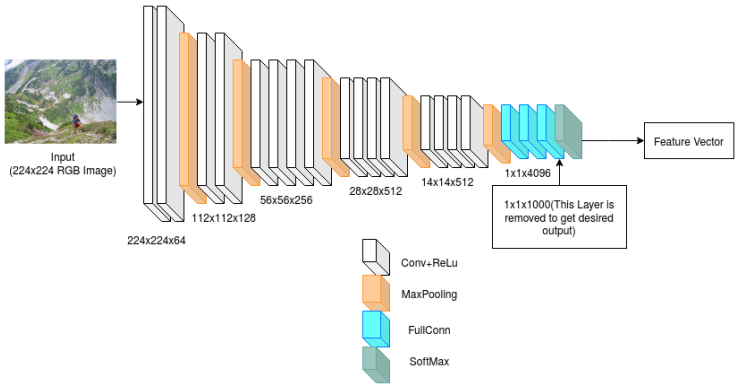

Figure 5. A Descriptive Representation of VGG19 encoder for desired output.

the most suitable of these five. The output is passed to the three other densely connected LSTMs and then to the two dense layers to get the enhanced description. The architecture of the proposed Dense-LSTM is shown in figure 4.

LSTM is a particular type of RNN with the capability of remembering, forgetting and information updating in long-term dependencies. Hence, LSTM is preferred for language modelling. In the architecture of Dense-LSTM, the used dimension for embedding is 256, and LSTM layers are four. Each layer is densely connected to other layers. The output of LSTMs and the image feature vector is given as input at the first dense layer. A block diagram and detailed architecture for the model are shown in figure 2 and figure 3 respectively. A brief description of the encoder and decoder is shown in figure 5 and figure 4 respectively. The detailed architecture with dimensions details is given in figure 6.

In figure 4, text embedding is given as input to the LSTM layer, which is further given to the densely connected three LSTMs. Output from these layers is given as input to the two dense layers connected sequentially. Output of $LSTM_1$ is given to the $LSTM_2$, $LSTM_3$, and $LSTM_4$ with weights 1, 0.25 and 0.1 respectively. Similarly, output of $LSTM_2$ is given

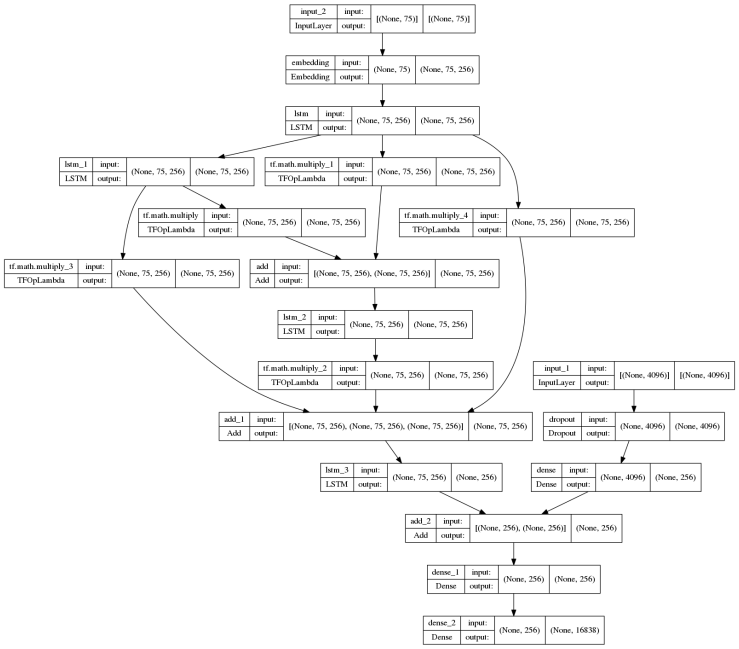

Figure 6. A Descriptive Representation of Complete model with dimension details.

as input to $LSTM_3$ and $LSTM_4$ with weights 0.75 and 0.1 respectively. In same way, output of $LSTM_3$ with weight 0.8 is passed to $LSTM_4$. Then the output of $LSTM_4$ and image features are given as input to the first dense layer, which is further passed through the second dense layer to get the desired output. Using LSTM sequentially introduces the short-term dependencies between the source (image features) and target sentence (required description) [44]. Due to this, performance is also improved.

The proposed model is trained for 20 epochs on Flickr8k and Flickr30k datasets for an automatic text generation task on a training set of 6000 and 25426 images, respectively. However, it is noticed that less number of epochs are sufficient for efficient model training when stochastic gradient descent is used in model regularization. As in this case, multi-class classification is required; therefore, Categorical Cross-Entropy Loss is used for classification purposes.

$$Loss = -\sum_{i=1}^{outputsize} y_i \cdot log\, \hat{y}_i \tag{8}$$

where, i, y_i, $\hat{y}_i$ represents the scalar value in the model output, target and output respectively.

An optimizer is used through this loss function for all the parameters for tuning the learning rate. The learning rate of the parameter aid the optimizer in weight updating in the direction opposite of the gradient. For which 0.2 learning rate is used here. Equation 8 represents the way to calculate the required loss. The minimum validation loss model is saved to use further for testing purposes. The configuration used during training is Intel(R) Xeon(R) CPU @ 2.30GHz and 12GB NVIDIA Tesla K80 GPU. Once the system is learned, then could be used for different purposes like security, content analysis, IoT-based applications etc.

For the performance evaluation of the model, some metric is required. Here, several evaluation matrices are available for the quality evaluation of textual data. The metric for assessment depends on the task for which it is needed. In the same field, several types of models are used. The proposed model is based on the CNN-RNN model, and based on the facts shown in the figure **??**, the BLEU metric gives better results for such types of models in evaluation. BLEU stood for Bilingual Evaluation Understudy and was used to determine the quality of text which has been translated. In BLEU score, quality is measured

Figure 7. Example images and descriptions from Flickr8K dataset

by calculating the difference between the machine translated text and human translated text. Formula to calculate BLEU score is given below:

$$BLEU = BP * e^{\sum_{n=1}^{N} W_n log p_n} \quad (9)$$

where, BP, N, W_n, and p_n are brevity penalty, number of n-grams, weight for each modified precision, and modified precision respectively.

3. Results and Discussion

3.1. Datasets

Flickr8K [21] and Flickr30k [45] are the datasets used for this work. In Flickr8K, a total of 8k images are there. Each image has five sentences as a description. Pictures are selected from six groups of the Flickr8k dataset and are not intended to contain any individuals or areas. Flickr30k dataset consists of 31783 images with 158915 descriptions, i.e. five descriptions for each image. However, Flickr30k contains Flickr8k with extended images. In the Flickr8k and Flickr30k datasets, images and descriptions are kept separately in two folders. A unique id is used for each image, and five different descriptions for that image with that same unique id are listed in a file. Data-set contains all images in RGB format. Preprocessing is done before passing them to the model.

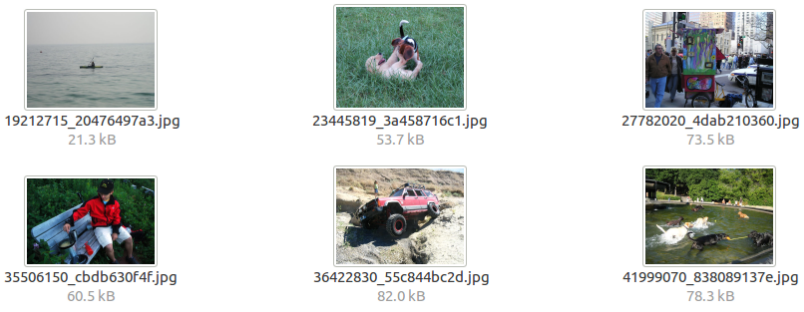

Figure 8. Sample images in the Flickr8K and Flickr30k datasets

Fig. 7 represents the sample images from the dataset with their respective descriptions. Fig. 8 represents the images with their unique id and size in the Flickr8k and Flickr30k datasets. Dataset splits used for Flickr8k and Flickr30k are as follows: In Flickr8k, 6k, 1k, and 1k images are used for training, testing and validation purposes respectively. Flickr30k contains 25k, 3k and 2k images for training, testing and validation purposes, respectively.

3.2. Results

In the proposed model, translation is from visual content to natural language. Therefore it is used to identify the model's accuracy in terms of the quality of generated text for a given image. It makes the comparison in n-gram where 'n' could be 1 to 4. Accordingly, scores are named BLEU-1 for n=1, BLEU-2 for n=2, and so on. These scores give the accuracy of the description generated.

In [20], different metrics comparison for the CNN-RNN model is discussed, which indicates that BLEU gives more accurate results than the other evaluation metrics for similar model types.

Table 3. Performance evaluation of CNN models with LSTM on BLEU-score

Score	VGG-19	VGG-16	ResNet-50	Xception	InceptionV3
BLEU-1	0.59	0.58	0.55	0.53	0.59
BLEU-2	0.36	0.34	0.31	0.29	0.35
BLEU-3	0.26	0.25	0.23	0.21	0.25
BLEU-4	0.16	0.14	0.12	0.11	0.14

In these datasets, a thousand images for Flickr8k and 3k for Flickr30k are used for testing purposes. For each image, a description is generated to give 1000 descriptions for Flickr8k and 3k for Flickr30k. BLEU score is calculated for the generated text based on the illustrations available in the dataset. Performance evaluation of the model using the BLEU score is shown in Table 3 in the case of different CNN models with LSTM on the Flickr8k dataset.

The detailed review shows that VGGNet is the most preferred network over other networks for such types of tasks. It is a deep convolutional neural network with 16 layers in VGG16 and 19 in VGG19. As VGG19 is deeper than VGG16, three additional layers should give a better result supported as per the results in table 3. Because of this, it creates better feature vectors than VGG16. VGG19-the pre-trained model used is trained on a vast data set, 'ImageNet' having around million images with thousand object categories and therefore rich features vector representation is learned. Compared with different models in table 3, all four scores, BLEU-1 to BLEU-4, are better for VGG19. This model is further trained on the Flickr8k and Flickr30k training dataset. Features given by this model are passed through the Dense-LSTM for sentence formation.

Table 4. Performance evaluation on Flickr8K and Flickr30k Dataset using BLEU score

References	Model	B1 on Flickr8k	B1 on Flickr30k
Yin Cui[23]	Show N Tell	0.56	0.58
Junhua Mao[22]	AlexNet + m-RNN	0.565	0.59
Karpathy[14]	CNN + m-RNN	0.579	-
Komal[42]	VGG19+LSTM	0.59	-
Proposed model	**VGG19+Dense-LSTM**	**0.60**	**0.62**

Results are shown in Table 4, which also supports that in the CNN-RNN model, using VGG19 as CNN and Dense-LSTM as RNN gives considerably good results compared to similar approaches while using on Flickr8k and Flickr30k datasets. Model performance can be more promising on larger datasets like MSCOCO. As evaluation for the proposed model is done on Flickr30k comes out better in just ten epochs than on flickr8k in 20 epochs. In Flickr30k, some of the images' original descriptions are not as per the content of the picture.

The diversion between visual content and their descriptions is quite significant. This kind of training data also affects the model performance. Therefore, if training data provided (Flickr30k) is processed further to overcome this issue, results may improve further.

The proposed model performs significantly on some of the images of Flickr30k testing data. Observations are listed in table 5. The number of images for which the BLEU score is less than 0.40 are 331, 1956, 2582 and 3039 for BLEU-1, BLEU-2, BLEU-3, and BLEU-4, respectively.

Table 5. BLEU scores for more than 60 percentile for the proposed model

Percentile	BLEU-1	BLEU-2	BLEU-3	BLEU-4
90th	0.76923	0.56613	0.46638	0.33926
85th	0.74708	0.52422	0.42623	0.29982
80th	0.69230	0.48038	0.38622	0.23566
75th	0.69230	0.46389	0.33543	5.04602e-78
70th	0.69230	0.43852	0.30282	4.51330e-78
65th	0.64104	0.41602	0.27778	4.20009e-78
60th	0.61538	0.39223	0.25481	3.88912e-78

Therefore, in some cases, the generated descriptions are not as accurate as those given by humans in areas like colour or context. This can be resolved by training the model on a large data set. Or preprocessing the dataset at the description level could be done. Descriptions generated by the proposed model having mixed results are shown in figure 9.

Figure 9. Results of textual descriptions generated using proposed method.

4. Application

Text generation for the visual content has a wide range of applications like image editing tools, video summarizing, aid for visually impaired people etc. A GUI (Graphical User Interface) is developed in the proposed work, which could be used for child education. Image-based question answering system is also one of the applications. The developed interface is shown in figure 10. The interactive study, which includes visuals and sounds, was found more attractive, interesting, involving and easy for children. This tool provides more relevant information in text form about the content given as input, as this model uses both foreground and background details while generating the output. Due to this, it provides almost all the details that a human can tell, and the most suitable textual description is generated about the content given as input. It also provides the generated text in audio form. Using this proves to be useful for the visually impaired/ incapable people by fixing the camera in an individual's walking cane to guide them about the path, scene or object obstacles. In addition, a user can search for similar images and for google queries using GUI buttons.

5. Conclusion

An encoder-decoder-based framework with novel Dense-LSTM architecture is proposed that provides natural language descriptions for scene descriptions with context information. Two neural networks are used, one as an encoder and one as a decoder.

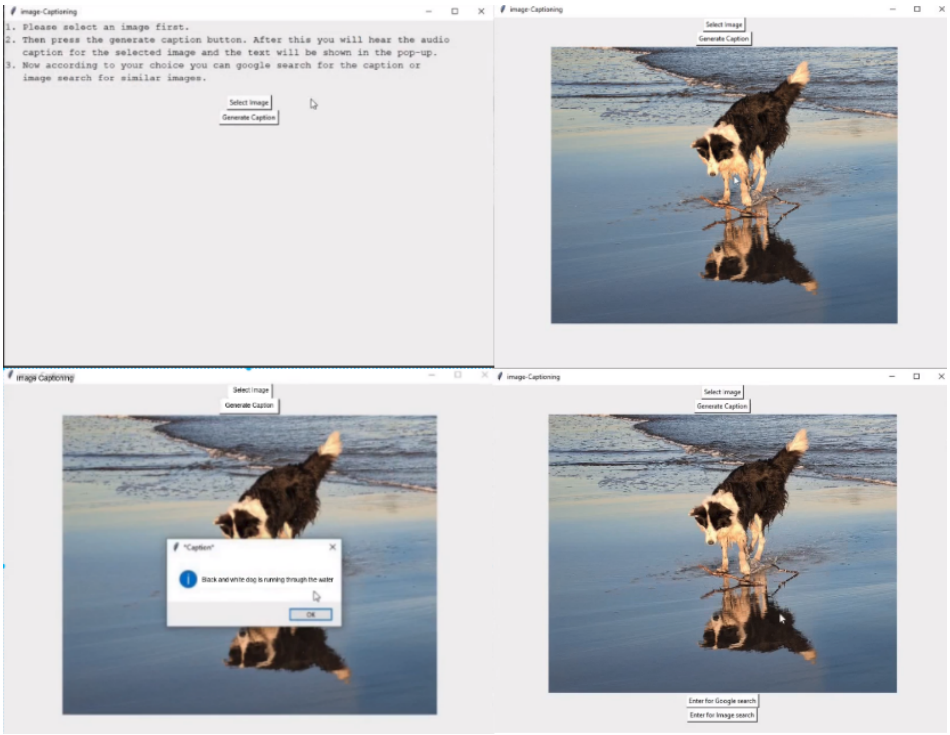

Figure 10. Step-wise screenshots of GUI from top left to bottom right: (a) User is required to select an image using the select image button, (b) selected image will be displayed, (c) by pressing generate caption button, a caption is generated in a pop-up window and audio form, (d) Two more option will be displayed: enter for a google search for generated cation and enter for an image search for the selected image.

A CNN is used as an encoder for object identification in given input and to find the in-between relationship between objects by creating a feature vector for the given content. Dense-LSTM is used as a decoder for description generation as it provides better results than LSTM, one of the RNN variants, when used with CNN. Evaluation is done using BLEU score on the Flickr8k and Flickr30k datasets. The model is suitable and could be used for the IoT-enabled visual content to generate the more relevant description in practical applications.

The proposed model generates comparatively good descriptions and audio for the visual content given as input. In general, the descriptions generated are good enough to consider. Still, the wrong object in terms of colour or context identification is done which will be considered in future work. Comprehensively model can generate considerably good results and can be used in similar applications. A GUI is developed to provide the usability ease.

6. Future Work

Textual descriptions for visual content can proven to be helpful in day-to-day activities. Suitable descriptions for the visual content received from the IoT device can be used for surveillance applications. The proposed model addresses the context information for scene using Dense-LSTM. Still, scope of improvement is there to address in the future work. The model deals with the time optimization with increased accuracy for text generation tasks for visual content by fine-tuning the model. Model is implemented to deal with real-world problems to solve related issues like summarizing videos, guiding path and providing information, etc. An application using IoT devices could be developed to take advantage of descriptions for visual content. The model's accuracy can be enhanced; if the model is trained on larger datasets, it could generate more general descriptions for new images. Similarly, more applications are there for which this model can be used.

Author Contributions: Authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: "Not applicable" as no involvement of humans or animals.

Informed Consent Statement: "Not applicable" as this study did not involve humans.

Data Availability Statement: Flickr8k and Flickr30k datasets are used.

Acknowledgments: Center for Cognitive Computing, IIIT Allahabad for providing facility to this research.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Vinyals, O., Toshev, A., Bengio, S., Erhan, D. (2016). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. IEEE transactions on pattern analysis and machine intelligence, 39(4), 652-663.
2. Singh, Dhananjay, Gaurav Tripathi, and Antonio J. Jara. "A survey of Internet-of-Things: Future vision, architecture, challenges and services." 2014 IEEE world forum on Internet of Things (WF-IoT). IEEE, 2014.
3. Chu, Yan Yue, Xiao Yu, Lei Sergei, Mikhailov Wang, Zhengkui. (2020). Automatic Image Captioning Based on ResNet50 and LSTM with Soft Attention. Wireless Communications and Mobile Computing. 2020.
4. N. Xu et al., "Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning," in IEEE Transactions on Multimedia, vol. 22, no. 5, pp. 1372-1383, May 2020.
5. Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. PMLR, 2015.
6. Chen, Tianlang, et al. ""Factual"or"Emotional": Stylized Image Captioning with Adaptive Learning and Attention." Proceedings of the European Conference on Computer Vision (ECCV). 2018.
7. Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, Jiebo Luo; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4651-4659
8. Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. 2016. Image Captioning with Deep Bidirectional LSTMs. In Proceedings of the 24th ACM international conference on Multimedia (MM '16). Association for Computing Machinery, New York, NY, USA, 988–997.
9. Aung, San Pa, Win nwe, tin. (2020). Automatic Myanmar Image Captioning using CNN and LSTM-Based Language Model. 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020)At: Marseille, France, 2020.
10. Ashton, Kevin. "That 'internet of things' thing." RFID journal 22.7 (2009): 97-114.
11. Elias, Andy Rosales, et al. "Where's the bear?-automating wildlife image processing using iot and edge cloud systems." 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE, 2017
12. Kapoor, Ayush, et al. "Implementation of IoT (Internet of Things) and image processing in smart agriculture." 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). IEEE, 2016
13. Wang, Cheng, et al. "Image captioning with deep bidirectional LSTMs." Proceedings of the 24th ACM international conference on Multimedia. 2016
14. A. Karpathy and L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, Stanford University, 2017.
15. K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "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, Philadelphia, 2002.
16. Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, Wei Xu. "CNN RNN: A Unified Frame-work for Multi-Label Image Classification". The IEEE Conference on Computer Vision and Pattern Recogniion (CVPR), 2016, pp. 2285-2294.
17. You, Q.; Jin, H.; Wang, Z.; Fang, C.; and Luo, J. 2016. Image captioning with semantic attention. In CVPR, 4651–4659.
18. Kulkarni, G.; Premraj, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A. C.; and Berg, T. L. 2011. Baby talk: Understanding and generating simple image descriptions. In CVPR, 1601–1608.
19. Alzubi, R. Jain, P. Nagrath, S. Satapathy, S. Taneja and P. Gupta, "Deep image captioning using an ensemble of CNN and LSTM based deep neural networks", Journal of Intelligent Fuzzy Systems, vol. 40, no. 4, pp. 5761-5769, 2021. Available: 10.3233/jifs-189415.
20. M. D. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," ACM Computing Surveys, vol. 51, no. 6, pp. 1–36, 2018.
21. Rashtchian, Cyrus, Young, Peter, Hodosh, Micah, and Hockenmaier, Julia. Collecting image annotations using amazon's mechanical turk. In NAACL-HLT workshop 2010, pp. 139–147, 2010.
22. Junhua Mao and Wei Xu and Yi Yang and Jiang Wang and Zhiheng Huang and Alan Yuille, "Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN)", ICLR 2015.arXiv:1412.6632

23. Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, Serge Belongie; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5804-5812
24. Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, Tao Mei; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4894-4902
25. Jyoti Aneja, Aditya Deshpande, Alexander G. Schwing; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5561-5570
26. Yang Feng, Lin Ma, Wei Liu, Jiebo Luo; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4125-4134
27. Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, Vaibhava Goel; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7008-7024
28. Y. Zhou, Y. Sun and V. Honavar, "Improving Image Captioning by Leveraging Knowledge Graphs," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 283-293, doi: 10.1109/WACV.2019.00036.
29. Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, Chris Sienkiewicz; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 49-56
30. Sun, B., Yang, L., Lin, M., Young, C., Dong, P., Zhang, W. and Dong, J., 2019. Supercaptioning: Image captioning using two-dimensional word embedding. arXiv preprint arXiv:1905.10515.
31. S. Amirian, K. Rasheed, T. R. Taha and H. R. Arabnia, "Automatic Image and Video Caption Generation With Deep Learning: A Concise Review and Algorithmic Overlap," in IEEE Access, vol. 8, pp. 218386-218400, 2020, doi: 10.1109/ACCESS.2020.3042484.
32. Sharma, P., Ding, N., Goodman, S. and Soricut, R., 2018, July. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2556-2565).
33. Bai, S. and An, S., 2018. A survey on automatic image caption generation. Neurocomputing, 311, pp.291-304.
34. Rennie, Steven J., et al. "Self-critical sequence training for image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
35. Venugopalan, Subhashini, et al. "Captioning images with diverse objects." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
36. Zhang, Li, et al. "Actor-critic sequence training for image captioning." arXiv preprint arXiv:1706.09601 (2017)
37. Wu, Qi, et al. "Image captioning and visual question answering based on attributes and external knowledge." IEEE transactions on pattern analysis and machine intelligence 40.6 (2017): 1367-1381
38. Aneja, Jyoti, Aditya Deshpande, and Alexander G. Schwing. "Convolutional image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
39. Wang, Qingzhong, and Antoni B. Chan. "Cnn+ cnn: Convolutional decoders for image captioning." arXiv preprint arXiv:1805.09019 (2018).
40. Zaremba, Wojciech, and Ilya Sutskever. "Learning to execute." arXiv preprint arXiv:1410.4615 (2014)
41. Onita, Daniela, Adriana Birlutiu, and Liviu P. Dinu. "Towards Mapping Images to Text Using Deep-Learning Architectures." Mathematics 8.9 (2020): 1606
42. Garg, Komal, Varsha Singh, and Uma Shanker Tiwary. "Textual Description Generation for Visual Content Using Neural Networks." International Conference on Intelligent Human Computer Interaction. Springer, Cham, 2021.
43. Xie, Yue, et al. "Attention-based dense LSTM for speech emotion recognition." IEICE TRANSACTIONS on Information and Systems 102.7 (2019): 1426-1429.
44. Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." Neural computation 31.7 (2019): 1235-1270.
45. Young, Peter, et al. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions." Transactions of the Association for Computational Linguistics 2 (2014): 67-78.
46. Zhang, Jin, et al. "Data augmentation and dense-LSTM for human activity recognition using WiFi signal." IEEE Internet of Things Journal 8.6 (2020): 4628-4641.
47. Niu, Zhenxing, et al. "Hierarchical multimodal lstm for dense visual-semantic embedding." Proceedings of the IEEE international conference on computer vision. 2017.
48. He, Jun-Yan, et al. "DB-LSTM: Densely-connected Bi-directional LSTM for human action recognition." Neurocomputing 444 (2021): 319-331.