Received: January 15, 2021. Revised: March 1, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Two-Tier LSTM Model for Image Caption Generation

Phyu Phyu Khaing[1]* **May The` Yu[2]**

[1]*Image Processing Lab, University of Computer Studies, Mandalay (UCSM), Myanmar*
[2]*Faculty of Information Science, University of Computer Studies, Mandalay (UCSM), Myanmar*
* Corresponding author's Email: phyuphyukhaing@ucsm.edu.mm

Abstract: Image captioning is systematically generating the caption of the image with a sentence description. In the past few years, the automatic process of creating image caption has fascinated the great interest in Artificial Intelligence (AI) field. Image captioning defines as the basic process of building the conjunction of image processing and natural language processing at input and output position. All image processing tasks, such as the segmentation of image, object tracking, object detection, image recognition, and many others, are mostly performed using Convolutional Neural Networks (CNNs). To perform the natural language processing tasks, just as semantic role labelling, neural machine translation, speech recognition, question and answering, and many others, Recurrent Neural Networks (RNNs) and long-term memory networks (LSTMs) are essential for some of the biggest breakthroughs. This paper proposes the efficient encoder-decoder framework-based image captioning model, namely Two-Tier LSTM (TT-LSTM) Model. The TT-LSTM model is designedly implemented upon the encoder-decoder framework with two LSTM layers. This research is implemented on the MSCOCO, Flickr30k, and Flickr8k datasets; and evaluated with standard evaluation matrixes such as ROUGE-L, CIDEr, and four BLEU scores. The outcomes of the experiments on the typical datasets reveal that the proposed model generates meaningful natural language sentences. The proposed model also improves the sentence generation efficiency and can achieve better performance for image caption generation.

Keywords: Image caption generation, Deep learning, Encoder-decoder, Two-tier LSTM model.

1. Introduction

Image captioning is a system by which descriptions are produced from the image with a sentence to provide the knowledge of the different components of the image. Components contain the objects/individuals in the image, the context where it focuses and associates the objects to all image entities and its environment. The objects should be combined with all the image entities. The massive amounts of knowledge existing around us in the world can be described with formal language or any other kind of interaction. In the same way, language may be used for the valuable and meaningful awareness of the scenes in images. This helps to understand the scene more clearly, by creating image captions and uses captions to completely grasp the contents of the images.

The purpose of image description is to describe an image automatically in one or more natural language sentences. The main challenges occur when interpreting two separates, but often combined approaches of computer vision and natural language processing [1]. First, the objects on the scene needed to be recognized and their relationships identified, followed by properly forming sentences for expressing the contents of the image [2]. Image caption generation also varies significantly from image presentation, since people focus on common sense and observation, emphasize relevant details, and neglect the objects and the connections of objects [3].

In the last few years, image captioning is an automated production of a natural language sentence that is related to input images to permit significant changes via the attention mechanism to the encoder-decoder system [4, 5]. Convolution Neural Network

(CNN) is commonly utilized at the encoder-decoder system at the encoding process of the image to extract the features from the image and a Recurrent Neural Network (RNN) is implemented to construct an input image's description [6, 7]. Then, the image caption generation, which focuses on some part of an input image, can be produced by the attention mechanism [8-10].

This paper proposes the Two-Tier LSTM (TT-LSTM). Captioning model for image-focused upon the inject and merge model. TT-LSTM Model build as an efficient encoder-decoder framework. Unlike the previous works, the proposed model is utilized two LSTM model. In the proposed model, Convolutional Neural Network (CNN) is to encode the input image, and Long Short-Term Memory (LSTM) is to encode the sentence caption and to generate the sentence as the decoder. The pretrained CNN (XceptionNet) is availed to extract the high-level visual semantic knowledge as the feature vector of the image from the second last activation layer of XceptionNet by dropping the last classification layer. The typical LSTM is performed for encoding the embedded sequence as the language decoder. Bidirectional LSTM is utilized as the decoder by entering the blended input, that combined the encoded image vector with an encoded sentence vector, to catch up with the previous and next contexts.

Generally, this paper comprises three main technical contributions as follow:

- To achieve the preciseness of the generated captions for image caption generation, this paper develops a Two-Tier LSTM (TT-LSTM) Model for generating image caption.
- Three benchmark datasets: MSCOCO [11], Flickr30k [12], and Flickr8k [13], are validated for detailed experiments to judge the capability of the proposed model.
- Our methodology is assessed by appraisal measurements to illustrate comparative outcomes of the state-of-the-art approaches.

The structure of this paper is constructed with the following sections. Section 2 describes the related works that concerned with the proposed method. The proposed architecture of the system demonstrates in Section 3. The experimental implementation of the proposed model represents in Section 4. The summary of the paper discusses in Section 5 under the conclusion.

2. Related work

This part discusses the literature of previous research works for image caption generation. Traditional image captioning methods and deep learning-based methods are the existing methods for image captioning.

2.1 Traditional image captioning models

For traditional image captioning methods, there are two particular types. The first method is image captioning focused on the template called template-based methods and the second one is image captioning focused on the retrieval processes called retrieval-based methods.

The template-based methods for image captioning must predefine the template variety of relationships between the objects from the image and the labels of these objects. Such relationships and names of objects fill the empty slots, and the caption of the image can be obtained from the empty slots. The phrases, however, that describe as the caption of the image generated by these methods, are only one sentence for one image. That sentence is not a separate expression. To generate relevant image descriptions, Hidden Markov Model (HMM) is employed in [14] by filtering the highest log-likelihood from four corpora that consist of objects, verbs, scenes, and prepositions. Similarly, in [15], the Conditional Random Field (CRF) model is implemented with the correspondence attributes, objects, and prepositions prediction to construct the sentence according to predefined templates. A new subspace embedded approach is suggested for image caption generation, called Common Subspace for Model and Similarity (CoSMoS) in [16].

The problem of the image captioning process is known to be an information collection problem with the retrieval methods. The first is to construct the image description from the training sample by deriving the meaningful sentence for a similar image. After that, updating the image description is processed to achieve the information expression for the input image. The retrieval-based method generates the caption of the image depended on a large dataset without any doubt, and the generated image caption is restricted with the description of the training dataset. An automated visual concept discovery (VCD) algorithm is initiated by utilizing concurrent text and image corpora for bidirectional tasks for the image and sentence retrieval and tagging tasks for the image in [17]. To choose the consensus caption for the image, in [18], a simple K-Nearest Neighbor (KNN) retrieval model is utilized by relying on a neural network to excerpt image features. In [19], the authors investigated the natural language description generation methods for the image by developing data-driven strategies using retrieval-

based methods that applied two strategies: applying the global features and utilizing the detection of objects, regions, and scenes.

2.2 Deep learning-based methods

An encoder-decoder framework is the most efficient deep learning model to generate the caption of images. The recent encoder-decoder frameworks have emerged and have been successful, which allow an image caption with fluent phrases and various phrases to be produced.

The encoder-decoder framework for generating the image caption was suggested in [1]. At the image captioning, the widespread implementation of the encoder-decoder method proceeds from one powerful effect throughout the challenges of machine translation. Google NIC model, which applied Inception v3 for the encoder and long-short term memory (LSTM) for the decoder, has been introduced by the authors of [5]. At the start of the LSTM network, the image information was only given as the input for the image captioning. In [20], the authors suggested an extension of the LSTM model that would implement the semantic knowledge to direct the LSTM network in producing a text description for the image.

In [21], the new image captioning framework, namely Neural Baby Talk, is proposed by applying object detection to generate the natural language sentence. The authors applied two perspectives of the RNN task for an image caption generator in [22]. These two perspectives are called inject model and merge model. The inject model encourages that the image and words are entered into RNN, and the merge model is that RNN is only performed for the encoder of the word sequence. In our proposed method, RNN has been used for two purposes: the word sequence encoder and the sentence generator. Therefore, our proposed model is more accurate than the previous research work.

In order to obtain clearer image definitions, the authors implemented a coarse-to-fine image captioning model, which uses a stacked visual focus model in combination with various LSTM networks in [23]. The guiding network is developed that extends the encoder-decoder framework at the decoder step and the guiding vector can be adjusted to integrate image and language information in [24]. The RNN-based reinforcement learning framework is proposed by integrating with a novel multi-level policy function (word-level policy and sentence-level polity) and multi-level reward function (vision-language reward and language-language reward) in [25]. The language model for image captioning, in [26], namely character-level RNN (c-RNN), is developed by composing the descriptive sentence with characterization. c-RNN model is based on the inject model that one of encoder-decoder architecture by substituting with character level instead of word-level image captioning model. Although a character-level RNN model (c-RNN) is efficient than other word-level language models, our proposed TT-LSTM model achieves more performance than c-RNN.

In [27], the authors proposed a structural comparison of the various forms of 'conditioning' language choices based on the visual input, exploring their repercussions for the framework of caption generators. In the comparison, based on the encoder-decoder framework, init-inject, per-inject, par-inject, and merge models are utilized. The init-inject model uses an image vector at the initial state for the RNN with the first word of sequence; the per-inject model applies an image vector as an initial word of sequence; par-inject utilizes image vector and word in every time steps; merge model takes RNN for the word embedding. All four models use RNN in one place for different purposes, but we apply RNN at two places in the proposed model.

Image captioning framework with semantic element discovery and embedding is developed in [28]. The element embedding framework has comprised the integration of the CNN-LSTM model and object region detection, called LSTM (FD+RD). To connect the interval between visual image and semantic caption, element embedding long-short term memory (EE-LSTM) used both visual and semantic; local and global features. The LSTM (FD+RD) and EE-LSTM are more efficient than baseline models for image understanding but the sentence generation process is not fully completed. So, our proposed model constructs with two-tier LSTM to get more efficient.

In [29], an image captioning framework is developed with a two-stage process; scene graphs generation from the image and caption generation with pre-trained language generators; that based on encoder-decoder architecture. The authors developed a CNN+Transformer design network for image captioning, namely CaPtion TransformeR (CPTR), that build global context at every encoder layer in [30].

After the encoder-decoder model, it was added the attention-based approach as the extension. In [8], the authors initially suggested the development of image caption applied an attention-based approach

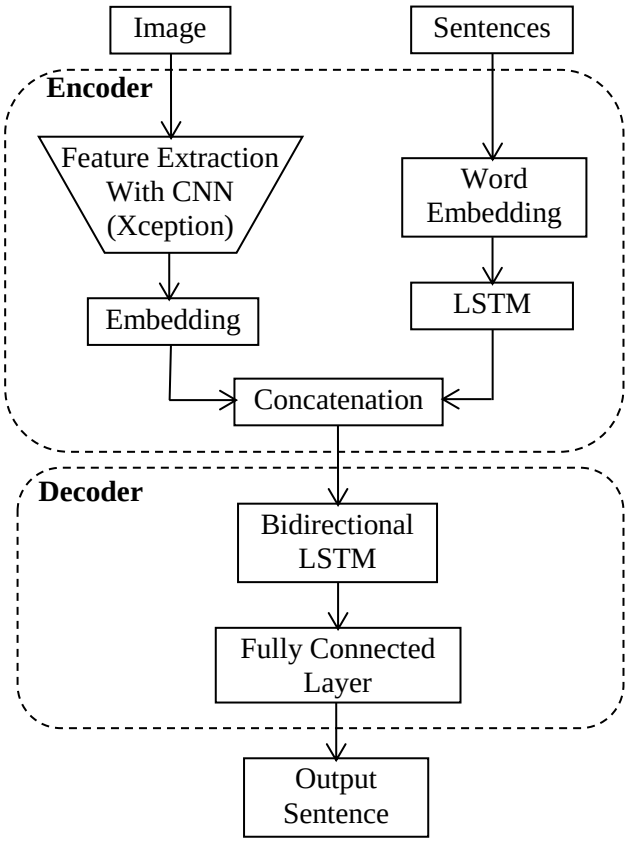


Figure. 1 Two-Tier LSTM (TT-LSTM) model

by implementing a convolutional layer for deriving the location-based spatial characteristics. The authors in [31] proposed a semantic attention mechanism to extract the text-related image features and integrated them with a bidirectional gLSTM (Bi-gLSTM) for the image captioning model. The work proposed in [32] is the top-down and bottom-up approaches. The combination process of these approaches and the semantic concepts are suggested to combine the attention mechanism at the image caption generation model in [33]. In the interest of language description generation for the image, the authors in [34] increased the text-guided attention approach. In [35], a text-based approach is suggested to strengthen the current guiding LSTM (gLSTM) and to utilize text-based knowledge to increase local attention. The authors in [36] also increased the cumulative attention functions to focus on the target item and other major regional rates that are bottom-up and top-down to be measured. Although the attention mechanism is mostly popular, our proposed model is only utilizing the encoder-decoder framework and we will extend our model with attention mechanism in the future study.

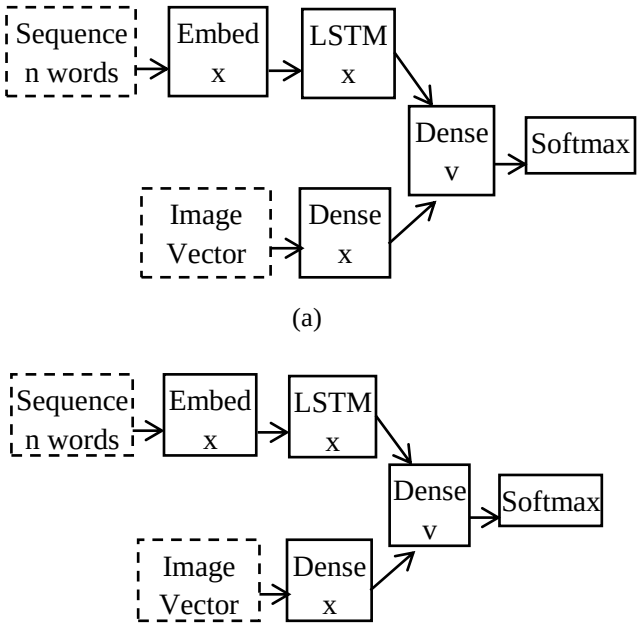


Figure. 2 Encoder-decoder framework: (a) inject model and (b) merge model

3 Proposed architecture of the system

Fig. 1 exhibits the efficient encoder-decoder models for image captioning, namely Two-Tier LSTM (TT-LSTM) model. This model is focused on the structure of the encoder-decoder framework to strengthen the model for the image captioning.

3.1 Encoder-decoder framework

The encoder-decoder framework is a basic deep learning framework for image captioning. This involves two elements:

- **Encoder:** A network model that reads the input image and encodes that image with an internal representation into a fixed-length vector.
- **Decoder:** A network model that can read the encoded vector and generate the output of the text description.

In the encoder-decoder framework, there are two basic models: the inject model and the merge model [22], that show in Fig. 2 (a) and (b).

In the inject architecture, the encoder first encodes the image into the vector view with a fixed length; and the decoder function uses an image-and-word sequence as an input text generation model to extract the continuous word in the sentence sequence. The injection model integrates the encoded image type and the word sequence produced in the text description.

The merge model blends the encoded forms of the image entry and the created text definition. A basic decoder model is then used to construct the next word in the sequence to merge these two encoded inputs. It distinguishes the modelling of the image entry, the text entrance, and the combination of the encoded outputs.

3.2 Two-tier LSTM (TT-LSTM) model

Two-Tier LSTM (TT-LSTM) Model shown in Fig. 1 blends the two-basic encoder-decoder framework: Inject Model and Merge Model. The proposed TT-LSTM model constructs with two encoder models for both image and text. Both encoding processes of the query image and the encoded form of the textual description generated are combined. The combination uses a decoder model to construct the sequence of the word. For image encoder, Convolutional Neural Networks (CNNs) apply. There are many pre-trained CNN models. Among them, Xception, a CNN model trained in the imagenet dataset, uses in this study. Long-Short Term Memory (LSTM) applies to the language encoder, and Bi-directional LSTM utilizes for the decoder.

3.2.1. Convolutional neural network (CNN)

Advanced deep neural network: Convolutional Neural Networks (CNNs), [37] can handle data that is an input shape like a 2D matrix. Image conveniently shows as a 2D matrix and CNN is highly useful for the image work. For the image feature extraction function, CNN is mostly used. The typical constituents of a CNN are the input layer, convolution layer, pooling layer, activation layer, and fully connected layer. The implementations of those typical constituents are:

- The ***input layer*** includes the image's raw pixel values and has three colour channels R, G, B.
- In order to construct output feature maps for the images, the ***convolution layer*** takes images from the previous layers and complements the specified number of filters. The output map numbers are equal to the given filter number.
- The ***pooling layer*** will conduct the down-streaming procedure through width and height (spatial dimensions). For the pooling layer, maximum pooling is primarily used.
- For the CNN ***activation function***, the ReLu function is generally applied. An activation function for element-wise activation like the maximum (0, x) threshold at zero is used in the ReLU layer.
- The class values are determined by the ***Fully Connected (FC) layer***. Any units in this layer are related to all numbers in the previous volume as with ordinary Neural Networks and as the name suggests.

3.2.2. Pre-trained convolutional neural network

Pre-trained convolutional neural networks are the models that someone else created to solve a similar problem. The pretrained models use several forward and backward iterations to define the right weights for the network. Much deep learning software is existing for the pre-trained model. Among them, this research will use Keras written by Python. Many pre-trained models include in Keras. There are denseNet, inceptionNet, mobileNet, nasNet, resNet, vggNet and xceptionNet. Each model will work the associated preprocessing. This research uses the XceptionNet model at the place of feature extraction from the images.

In this study, XceptionNet [38] model performs to extract the features of the image for the CNN model. The Xception model already trains on imagenet datasets and an extension of the Inception architecture, substituted with depth-separable convolutions for regular Inception units. Model Xception works pre-trained on ImageNet with weight. This model has a default input size of 299x299. The 36 convolution layers organize into 14 modules, and all of which, except the first and last modules, have linear residual links. Fig. 3 demonstrates the architecture of the Xception model.

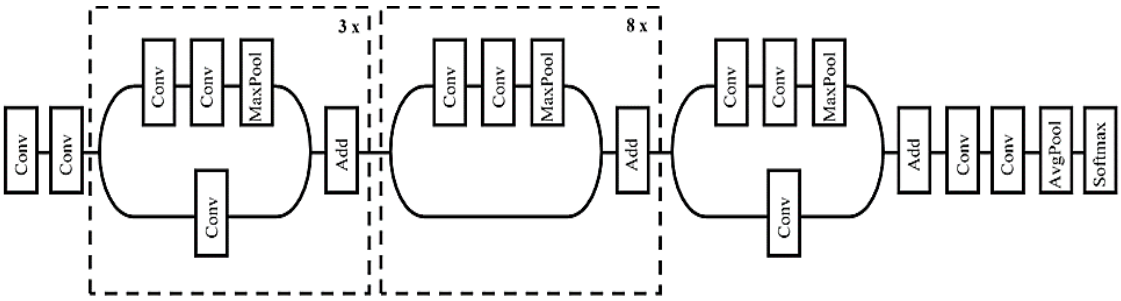


Figure 3. Xception Architecture

3.2.3. Long-short term memory

Long-Short Term Memory (LSTM) [39] networks are a special kind of RNN and can use to predict sequence problems and learn long-term dependencies. LSTM cell composes with the gate instead of memory such as input gate, forget gate, output gate, candidate layer. The sigmoid activation function utilizes for the forget gate, input gate, and output gate as single-layered neural networks; the Tanh function is for the Candidate layer as the activation function. Forget gate is to determine the knowledge thrown away from the cell state and a sigmoid layer makes the decision. The cell state determines what relevant materials will process, and the output gate is to determine what the output will be.

Fig. 4 shows the structure of the LSTM cell, and the following equations are applied to perform the LSTM cell.

$$i_t = \sigma(x_t \times W_{xi} + h_{t-1} \times W_{hi}) \quad (1)$$

$$f_t = \sigma\left(x_t \times W_f + h_{t-1} \times W_{hf}\right) \quad (2)$$

$$\bar{c_t} = \tanh(x_t \times W_{xc} + h_{t-1} \times W_{hc}) \quad (3)$$

$$o_t = \sigma(x_t \times W_{xo} + h_{t-1} \times W_{ho}) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \bar{c_t} \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

In the figure and equations, x_t is the cell input: called the input gate, $\bar{c_t}$ is the input activation, f_t is the forget gate, h_t is the next hidden state, h_{t-1} is the previous hidden state, o_t is the output gate, $c_{t-1}$is the previous cell state, c_t is the current cell, W_x is the weight for the input, W_h is the weight for the hidden state. The sigmoid (σ) function and the hyperbolic tangent (*tanh)* function are the element-wise nonlinear activation functions. $\odot$ is an element-wise multiplication.

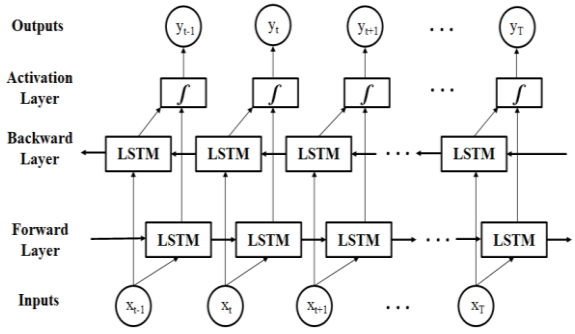


Figure 4. Structure of LSTM cell

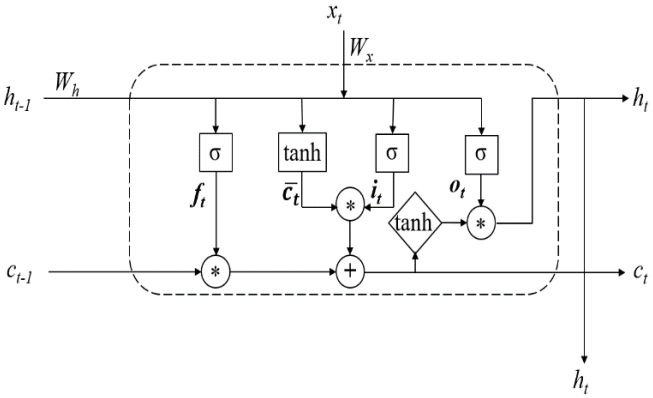


Figure 5. Bidirectional LSTM structure

3.2.4. Bidirectional LSTM

Bidirectional LSTM [40] is an extension of standard LSTM to increase the model output for the problems with sequence classification. Bi-LSTMs have the forward and backward pass. Instead of one LSTM on the entry sequence, Bi-LSTM trains with two LSTMs; a backward (future) layer and a forward (past) layer. The usage of Bi-LSTM does not make sense for all sequence prediction issues but can provide some advantage to those areas where it gets the acceptable results with the significant improvements. The network utilizes this additional context, and the results are faster output. Fig. 5 displays the bidirectional LSTM structure.

4 Experiments

This section presents datasets, evaluation metrics, and experimental results for the TT-LSTM model for image captioning.

4.1 Datasets

During classification, recognition, and detection processes for the image, and image caption generation process, there are various types of data sets. We will use the most well-known standard datasets for image captioning, such as MSCOCO [11], Flickr30k [12], and Flickr8k [13]. Fig. 6 shows the sample images and captions pair format of the MSCOCO dataset.

MSCOCO dataset published in 2014 by the authors [11], with the paper publication, in Computer Vision and Pattern Recognition (CVPR). It also continuously published updated versions of the dataset in 2015 and 2017. This research used the 2017 dataset and contains 123,287 images with five-sentence descriptions. It comprised 113,287 images, 5,000 images, and 5,000 images for the trained process, the validated process, and the tested process respectively.

Flickr30k dataset published in 2014 by the authors [12], with the paper publication, in Transactions of the Association for Computational Linguistics. It contains five-sentence descriptions for each image of 31,783 images. It comprised 29,783, 1,000 images, 1,000 images for the trained process, the validated process, and the tested process respectively.

Flickr8k dataset published in 2013 by Hodosh, Young, and Hockenmaier [13], with the paper publication, in Journal of Artificial Intelligence Research. It contains 8,091 images with five-sentence descriptions. It comprised 6,000 images, 1,000 images, and 1000 images for the trained process, the validated process, and the tested process respectively.

	1. Woman in swim suit holding parasol on sunny day. 2. A woman posing for the camera, holding a pink, open umbrella and wearing a bright, floral, ruched bathing suit, by a life guard stand with lake, green trees, and a blue sky with a few clouds behind. 3. A woman in a floral swimsuit holds a pink umbrella. 4. A woman with an umbrella near the sea. 5. A girl in a bathing suit with a pink umbrella.
	1. A young boy in winter clothes skiing in a very snowy landscape. 2. A little boy in a bright jacket on skis in the snow. 3. There is a young boy that is riding his skies down hill. 4. A little boy that is standing on ski. 5. A young boy in an orange snow jacket is on skis.
	1. Two giraffes standing together and looking towards an area of trees and bushes. 2. A couple of giraffes walk next to some trees 3. Two giraffes standing in the grass near trees. 4. Two giraffes' side by side in the tall grass look into the shaded tree line. 5. A couple of giraffes that are walking in the grass

Figure 6. Sample images and captions

4.2 Evaluation metrics

In our experiment, automated evaluation metrics utilize at the computation for the proposed model, such as ROUGE-L (RG-L) [41], CIDER (CDR) [42], and BLEU (B-4, B-3, B-2, B-1) [43].

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measures the automatic assessment of the content description by contrasting the other human-generated summaries and focuses on the recall.

Consensus-based Image Description Evaluation (CIDEr) calculates how most people identify an image similar to a candidate sentence by executing TF-IDF (Term-Frequency-Inverse Document Frequency).

Bilingual Evaluation Understudy (BLEU) evaluation metric judges the similitude at the interval of the sentence of ground truth and the sentence generated from the machine by a fraction of n-gram (n=1,2,3,4).

4.3 Experimental setup

The code for the TT-LSTM model writes with Python programming language. Keras CNN pre-trained models are used for this study and run on the Pycharm editor. Those models run on the RTX 2070 super 8GB GPU, 32GB Memory, and 64-bit Ubuntu operating system.

For the image encoder, we utilize the Xception model, the ReLU activation function, and the Repeat Vector layer. 2048 feature vector extracts from the fully connected layer of the Xception model by removing the last classification layer. ReLu activation function processes with embedding size: 256. Repeat vector uses to regenerate a 2D array for the input for the continuous layer.

For language encoder model, the system firstly constructs with a word embedding, LSTM model, and TimeDistributed layer. The embedding size and units are 256. The input vector size of the language encoding model is the maximum length of the sentence. The maximum length of a sentence can be different according to the dataset. TimeDistributed layer adds to the language encoder since this layer is incredibly helpful in dealing with the time series process, such as the sequence of video frames and sentences. Before decoding, the concatenation process performs to combine the image encoder and language decoder model. At the decoder model, the concatenation result enters into the bidirectional LSTM with 256 units. Finally, a softmax activation layer processes with vocabulary size. The training

model structure for the Flickr8k dataset is shown in Fig. 7.

The training process conducts with 0.0001 hyper-parameter of the Adam optimization algorithm [45] and a batch size of 32. Categorical cross-entropy is used in the cost function. Early stopping parameter is used to stop the training process by measuring the validation performance after the training epoch as soon as validation data output began to decline. We also generated captions for the images in the testing dataset with beam search that used the beamwidth 2 and a cumulative clipping period of the maximum number of words at the sentence to assess the trained models. Fig. 8 shows the training and validation loss for all datasets of our proposed model.

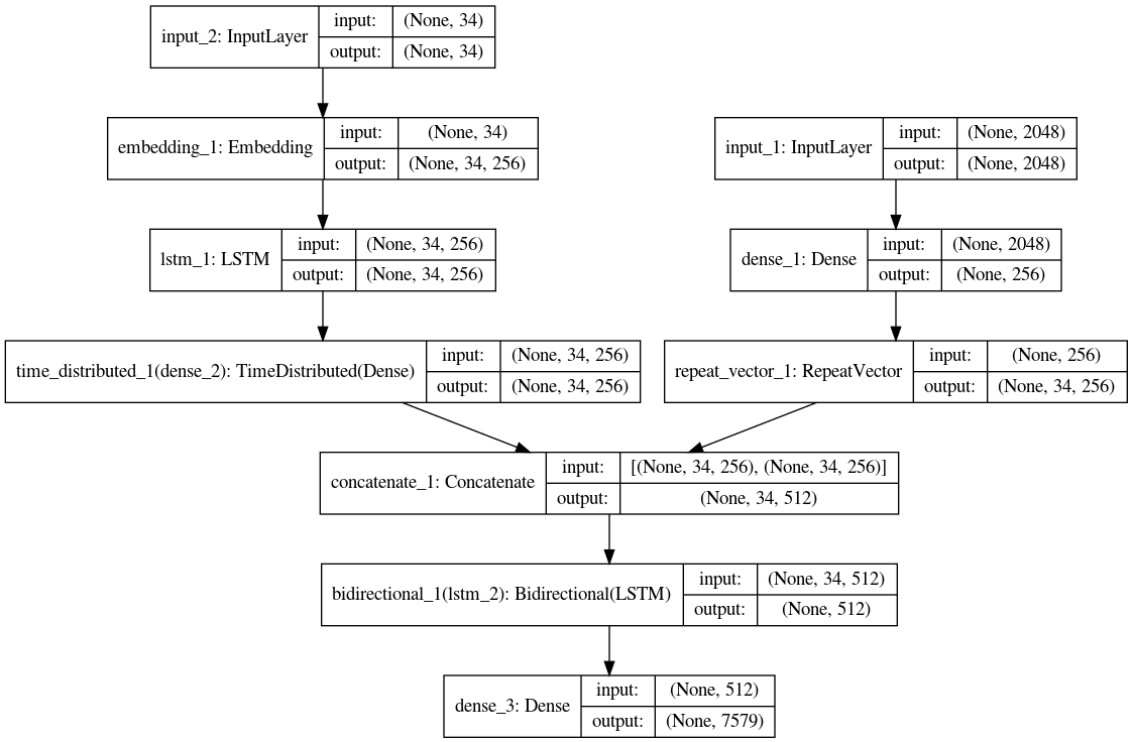


Figure 7. Training model structure for Flickr8k

Table 1. Comparative results with baseline model for the MSCOCO dataset

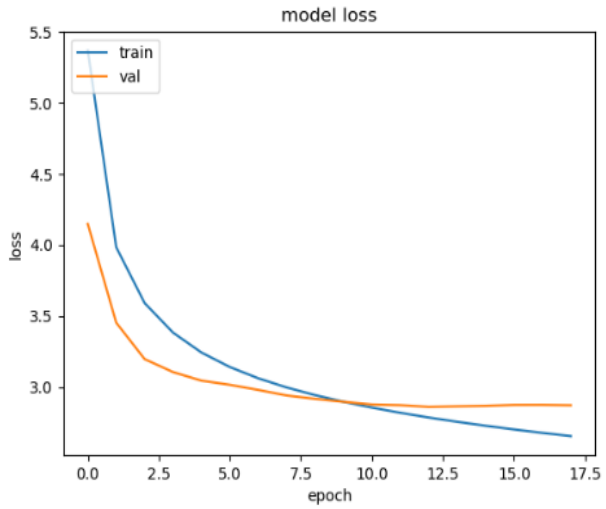
Models	RG-L	CDR	B4	B3	B2	B1
Inject	54.5	79.0	22.0	32.0	46.1	68.0
Merge	54.3	79.0	21.7	31.5	45.7	67.9
TT-LSTM	**55.0**	**79.1**	**22.5**	**32.9**	**47.6**	**69.4**

Table 2. Comparative results with baseline model for the Flickr30k dataset

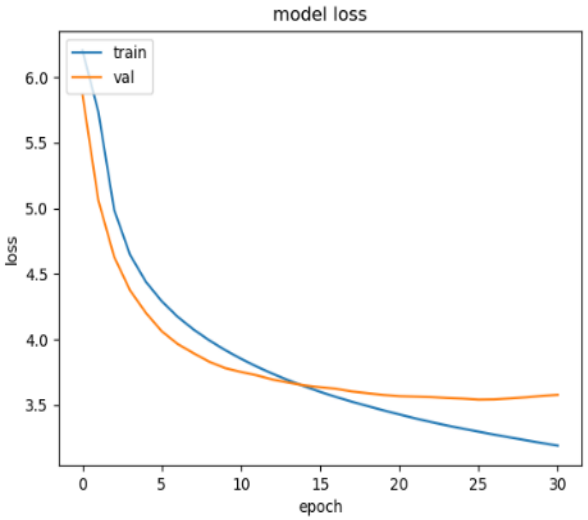
Models	RG-L	CDR	B4	B3	B2	B1
Inject	48.9	39.8	16.8	26.2	38.5	62.5
Merge	49.1	40.1	17.0	26.6	40.0	63.3
TT-LSTM	**50.2**	**40.3**	**20.0**	**29.7**	**44.2**	**66.7**

Table 3. Comparative results with baseline model for the Flickr8k dataset

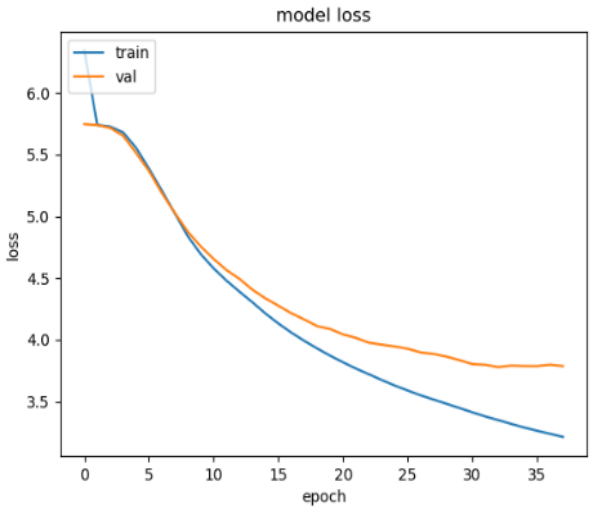
Models	RG-L	CDR	B4	B3	B2	B1
Inject	49.3	43.3	15.3	24.2	37.0	58.2
Merge	49.2	41.1	15.2	24.5	37.9	60.1
TT-LSTM	**51.9**	**49.1**	**18.4**	**28.9**	**43.5**	**65.8**



(a)



(b)



(c)

Figure 8. Training and validation loss: (a) MSCOCO dataset, (b) Flickr30k dataset, and Flickr8k dataset

4.4 Results

For the experimental results, we choose the inject model and the merge model as baseline methods to compare with the proposed efficient encoder-decoder model. And then, we analyse the Two-Tier LSTM (TT-LSTM) model with 1D LSTM and Bi-directional LSTM. Tables 1, 2, and 3 show the evaluation results of the TT-LSTM model on MSCOCO, Flickr30k, and Flicker8k datasets, respectively.

When compared with two baseline models, the proposed model achieves better results on all evaluation metrics. At the large dataset MSCOCO, all comparative results are slightly higher than baseline models. To compared on Flickr30k dataset, four BLEU scores (B4, B3.B2, and B1) of the proposed are more improved than baseline models. On the Flickr8k dataset, all evaluation results are higher than inject model and merge model. Therefore, the performance of the proposed model confirms the effectiveness of the caption generation process than two baseline models: inject and merge models, on all three datasets.

The proposed method compares with other classical models: Google NIC [5]; Init-Inject, Pre-Inject, and Par-Inject [27]; and LSTM (FD+RD) and EE-LSTM [28]. The comparative results with classical models describe in Tables 4, 5, and 6 on MSCOCO, Flickr30k, Flickr8k, respectively.

In Table 4, on the MSCOCO dataset, the proposed encoder-decoder framework is more efficient than other comparative methods. Only BLEU-1 (B1), BLEU-3 (B2), and ROUGE-L (RG-L) have good results by comparing with the start-of-the-art. Since the MSCOCO dataset contains many images with multiple and complex scenes, and all contents are difficult to completely detect. However, the proposed method can generate the sentence with correct words but cannot completely correct for the compound words.

On the Flickr30k and Flickr8k datasets, the proposed model achieves the moderately improved results through categorical cross-entropy loss. For all evaluation metrics except BLEU-3 (B3) and BLEU-4 (B4) scores, TT-LSTM produces considerable improved outcomes than all competing approaches at the Fickr30k and Flickr8k datasets. Since the long sequence of words are not completely correct in this study, BLEU-3 (B3) and BLEU-4 (B4) evaluation calculated with 3-gram and 4-gram of the sentence is a little worse for the result. The evaluation scores are decreased, extraordinarily, if the number of terms is long. Since the model equally considers all words, the usage of words cannot be fully utilized. Generally, the proposed model harmoniously performs better

than other state-of-the-art methods on almost all evaluations.

Fig. 9 shows the sample captioning results of image caption generation using the proposed approaches. Due to the fact that the proposed model developed with two encoder model for the image and the language, the system can prominently apply the features of image and the language. Additionally, the decoder model is also used the bi-directional LSTM, the training model can be more accurate. So, the proposed system model is able to generate good captions for images automatically.

5 Conclusion

In this paper, we investigate the efficient encoder-decoder framework for the image caption generation, namely Two-Tier LSTM (TT-LSTM). TT-LSTM model is applied two encoder models for the image encoder and language encoder, and one decoder model. For the image encoder, Xception Net is utilized, and LSTM is used for the language encoder. And then, two outputs from the image encoder and language encoder are concatenated to enter into the decoder model. Bi-directional LSTM is processed for the decoder model and to generate the relevant caption for the query image. We implement learning schemes to train the proposed model on three benchmark datasets: MSCOCO, Flickr30k, and Flickr8k, to perform the image caption generation. Our methods enhanced the quality of generated caption for the image. So, compared to the related image captioning processes, the proposed approach achieves reasonable competitive efficiency. Furthermore, we will add attention mechanism into the TT-LSTM model to achieve the more accurate caption and the best performance in the future study. The efficiency and generalization of our model is further demonstrated by this observation.

Table 4. Comparative results with classical models for the MSCOCO dataset

Models	**RG-L**	**CDR**	**B4**	**B3**	**B2**	**B1**
GoogleNIC [5]	-	-	24.6	32.9	46.1	66.6
init-inject [27]	49.9	81.8	27.1	36.7	50.2	67.9
pre-inject [27]	49.8	80.7	26.7	36.6	50.1	67.7
par-inject [27]	49.3	77.4	26.5	35.9	49.2	66.7
LSTM (FD+RD) [28]	-	-	25.3	36.6	51.1	**69.1**
EE-LSTM [28]	-	-	26.9	36.4	49.8	**67.5**
TT-LSTM	**55.0**	79.1	22.5	**36.7**	47.6	**69.4**

Table 5. Comparative results with classical model for the Flickr30k dataset

Models	**RG-L**	**CDR**	**B4**	**B3**	**B2**	**B1**
GoogleNIC [5]	-	-	18.3	27.7	42.3	66.3
init-inject [27]	42.5	38.3	19.1	28.3	41.9	61.3
pre-inject [27]	42.0	38.0	19.2	28.4	41.9	61.3
par-inject [27]	41.8	36.1	18.3	27.5	41.0	60.5
LSTM (FD+RD) [28]	-	-	**20.5**	**30.9**	**43.2**	**64.0**
EE-LSTM [28]	-	-	**17.0**	**25.7**	**39.1**	**59.2**
TT-LSTM	**50.2**	**40.3**	20.0	29.7	**44.2**	**66.7**

Table 6. Comparative results with classical model for the Flickr8k dataset

Models	**RG-L**	**CDR**	**B4**	**B3**	**B2**	**B1**
GoogleNIC [5]	-	-	-	27.0	41.0	63.0
init-inject [27]	44.5	48.1	19.1	28.5	42.4	61.1
pre-inject [27]	44.4	46.9	19.0	28.5	42.1	60.9
par-inject [27]	44.8	47.5	19.1	28.7	42.4	61.1
LSTM (FD+RD) [28]	-	-	19.5	**33.9**	**42.8**	**63.8**
EE-LSTM [28]	-	-	18.4	**27.5**	**40.8**	**59.8**
TT-LSTM	**51.9**	**49.1**	18.4	28.9	**43.5**	**65.8**

Generated Caption: two giraffes standing in the middle of field

Generated Caption: an airplane is flying through the sky

Generated Caption: man riding bike on the side of street

Generated Caption: group of people sitting on bench in front of building

Figure 9. Example results of proposed approach

Conflicts of Interest

The authors declare no conflict of interest on this research, publishing, and/or authorship of this article.

Author Contributions

Phyu Phyu Khaing contributed to develop the system and carry out the analysis, interpret the findings and compose the manuscript. This study was supervised by Dr. May The' Yu.

Acknowledgments

I am thankful to my supervisor, Dr. May The` Yu, for her supportive and constructive guidance on the planning and development of this research work.

References

[1] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal Neural Language Models", In: *Proc. of International Conf. on Machine Learning*, pp. 595-603, 2014.

[2] M. Ivasic-Kos, I. Ipsic, and S. Ribaric, "A Knowledge-Based Multi-Layered Image Annotation System", *Expert Systems with Applications*, Vol. 42, No. 24, pp. 9539-9553, 2015.

[3] M. Ivašić-Kos, M. Pavlić, and M. Pobar, "Analyzing the Semantic Level of Outdoor Image Annotation", In: *Proc. of MIPRO 2009-32nd International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp. 293, 2009.

[4] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, G. Zweig, "From Captions to Visual Concepts and Back", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1473-1482, 2015.

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA*, pp. 3156-3164, 2015.

[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", In: *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar*, pp. 1724-1734, 2014.

[7] A. Graves, "Generating Sequences with Recurrent Neural Networks", *arXiv preprint arXiv:1308.0850*, pp. 1-43, 2013.

[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", In: *Proc. of International Conf. on Machine Learning, Lille, France*, pp. 2048-2057, 2015.

[9] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen, "Encode, Review, and Decode: Reviewer Module for Caption Generation", *Computing Research Repository (CoRR), arXiv:1605.07912v3*, pp. 1-9, 2016.

[10] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pp. 375-383, 2017.

[11] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context", In: *Proc. of European Conf. on Computer Vision, Springer, Cham*, pp. 740-755, 2014.

[12] B. A. Plummer, L. Wang, M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k Entities: Collecting Region-To-Phrase Correspondences for Richer Image-To-Sentence Models", In: *Proc. of the IEEE International Conf. on Computer Vision*, pp. 2641-2649, 2015.

[13] M. Hodosh, P. Young, and J. Hockenmaier, "Framing Image Description as A Ranking Task: Data, Models and Evaluation Metrics", *Journal of Artificial Intelligence Research*, Vol. 47, pp. 853-899, 2013.

[14] Y. Yang, C. Teo, H. Daumé III, and Y. Aloimonos, "Corpus-Guided Sentence Generation of Natural Images", In: *Proc. of the 2011 Conf. on Empirical Methods in Natural Language Processing*, pp. 444-454, 2011.

[15] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Babytalk: Understanding and Generating Simple Image Descriptions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 12, pp. 2891-2903, 2013.

[16] Y. Ushiku, M. Yamaguchi, Y. Mukuta, and T. Harada, "Common Subspace for Model and Similarity: Phrase Learning for Caption Generation from Images", In: *Proc. of the IEEE International Conf. on Computer Vision*, pp. 2668-2676, 2015.

[17] C. Sun, C. Gan, and R. Nevatia, "Automatic Concept Discovery from Parallel Text and Visual Corpora", In: *Proc. of the IEEE International Conf. on Computer Vision*, pp. 2596-2604, 2015.

[18] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell, "Language Models for Image Captioning: The Quirks and What Works", *arXiv preprint arXiv:1505.01809*, 2015.

[19] V. Ordonez, X. Han, P. Kuznetsova, G. Kulkarni, M. Mitchell, K. Yamaguchi, K. Stratos, A. Goyal, J. Dodge, A. Mensch, H. Daumé, A. C. Berg, Y. Choi, and T. L. Berg, "Large Scale Retrieval and Generation of Image Descriptions", *International Journal of Computer Vision*, Vol. 119, No. 1, pp.46-59, 2016.

[20] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the Long-Short Term Memory Model for Image Caption Generation", In: *Proc. of the IEEE International Conf. on Computer Vision*, pp. 2407-2415. 2015.

[21] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural Baby Talk", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 7219-7228, 2018.

[22] M. Tanti, A. Gatt, and K. P. Camilleri, "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?", In: *Proc. of the 10th International Conf. on Natural Language Generation*, pp. 51–60, 2017.

[23] J. Gu, J. Cai, G. Wang, and T. Chen, "Stack-Captioning: Coarse-To-Fine Learning for Image Captioning", In: *Proc. of the 32nd AAAI Conf. on Artificial Intelligence*, pp. 6837–6844, 2018.

[24] W. Jiang, L. Ma, X. Chen, H. Zhang, and W. Liu, "Learning to Guide Decoding for Image Captioning", In: *Proc. of the 33rd AAAI Conf. on Artificial Intelligence*, pp. 6959–6966, 2018.

[25] N. Xu, H. Zhang, A.A. Liu, W. Nie, Y. Su, J. Nie, and Y. Zhang, "Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning", *IEEE Transactions on Multimedia*, Vol. 22, No. 5, pp. 1372-1383, 2019.

[26] G. Huang, and H. Hu, "c-RNN: A Fine-Grained Language Model for Image Captioning", *Neural Processing Letters*, Vol. 49, No. 2, pp.683-691, 2019.

[27] M. Tanti, A. Gatt, and K. Camilleri, "Where to put the Image in an Image Caption Generator", *Natural Language Engineering*, Vol. 24, No. 3, pp. 467-489, 2018.

[28] X. Zhang, S. He, X. Song, R.W.H. Lau, J. Jiao, and Q. Ye, "Image Captioning via Semantic

Element Embedding", *Neurocomputing*, Vol. 395, pp. 212-221, 2020.

[29] S. Vishnubhatla, and N. Sinha, "Image Captioning with Pretrained Language Generators", In: *Proc. of 8th ACM IKDD CODS and 26th COMAD*, pp. 427-427, 2021.

[30] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "CPTR: Full Transformer Network for Image Captioning", *arXiv preprint arXiv:2101.10804*, 2021.

[31] P. Cao, Z. Yang, L. Sun, Y. Liang, M. Qu Yang, and R. Guan, "Image Captioning with Bidirectional Semantic Attention-Based Guiding of Long Short-Term Memory", *Neural Processing Letters*, Vol. 50, No. 1, pp. 103-119, 2019.

[32] J. Chen, X. Song, L. Nie, X. Wang, H. Zhang, and T.S. Chua, "Micro Tells Macro: Predicting the Popularity of Micro-Videos via A Transductive Model", In: *Proc. of the 24th ACM International Conf. on Multimedia*, pp. 898-907, 2016.

[33] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image Captioning with Semantic Attention", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4651-4659, 2016.

[34] J. Mun, M. Cho, and B. Han, "Text-Guided Attention Model for Image Captioning", In: *Proc. of the Thirty-First AAAI Conf. on Artificial Intelligence*, pp. 4233–4239, 2017.

[35] L. Zhou, C. Xu, P. Koch, and J. J. Corso, "Watch What You Just Said: Image Captioning with Text-Conditional Attention", In: *Proc. of the on Thematic Workshops of ACM Multimedia*, pp. 305-313, 2017.

[36] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 6077-6086, 2018.

[37] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks", *arXiv preprint arXiv:1511.08458*, 2015.

[38] F. Chollet, "Xception: Deep learning with Depthwise Separable Convolutions", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1251-1258, 2017.

[39] S. Hochreiter, and J. Schmidhuber, "Long short-term memory", *Neural Computation*, Vol. 9, No. 8, pp.1735-1780, 1997.

[40] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, "Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks", In: *Proc. of the Seventh International Workshop on Health Text Mining and Information Analysis*, pp. 17-27, 2016.

[41] C.Y. Lin, "Rouge: A Package for Automatic Evaluation of Summaries", In: *Text Summarization Branches Out*, pp. 74-81, 2004.

[42] R. Vedantam, C.L. Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4566-4575, 2015.

[43] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation", In: *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311-318, 2002.

[44] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization", In: *Proc. of 3rd International Conf. on Learning Representations (ICLR)*, pp. 1-15, 2015.