

Video captioning in Vietnamese using deep learning

Dang Thi Phuc¹, Tran Quang Trieu¹, Nguyen Van Tinh¹, Dau Sy Hieu²

¹Department of Computer Science, Faculty of Information Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

²Department of Applied Physics, Faculty of Applied Science, University of Technology-Viet Nam National University HCMC, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received May 3, 2021

Revised Dec 20, 2021

Accepted Jan 12, 2022

Keywords:

Attention

Natural language processing

Sequence-to-sequence model

Transformer

Video caption

ABSTRACT

With the development of today's society, demand for applications using digital cameras jumps over year by year. However, analyzing large amounts of video data causes one of the most challenging issues. In addition to storing the data captured by the camera, intelligent systems are required to quickly analyze the data to correct important situations. In this paper, we use deep learning techniques to build automatic models that describe movements on video. To solve the problem, we use three deep learning models: sequence-to-sequence model based on recurrent neural network, sequence-to-sequence model with attention and transformer model. We evaluate the effectiveness of the approaches based on the results of three models. To train these models, we use Microsoft research video description corpus (MSVD) dataset including 1970 videos and 85,550 captions translated into Vietnamese. In order to ensure the description of the content in Vietnamese, we also combine it with the natural language processing (NLP) model for Vietnamese.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Dang Thi Phuc

Faculty of Information Technology, Industrial University of Ho Chi Minh City

12 Nguyen Van Bao, Ward 4, Go Vap District, Ho Chi Minh City, Vietnam

Email: phucdt@iuh.edu.vn

1. INTRODUCTION

Today, the increasing of demand for information exchange using digital cameras has spurred the development of many advanced techniques in retrieving video content. Among them, the problem of describing video content in natural language is receiving a lot of attention. The development of this problem opens up great opportunities in areas such as: self-driving vehicles, warning of dangerous actions for security purposes, assisting the blind, human-robot interactive.

The problem of describing video content requires recognition of objects and actions in the video and automatically generates short meaningful sentences in natural language to describe those objects and actions [1]. This problem involves both computer vision and natural language processing. In the past, there have been many studies about the problem of describing content through images [2]–[4] effectively. But the problem of describing actions via videos, it is more complicated in terms of image quality, data as well as difficulties in video feature extraction [5]. Some classical research directions such as extracting features by image processing [6]. Some camera action classification problems like [7], [8] are not very effective due to limited number of actions, the descriptions are not detailed enough when the data is diverse. Today, thanks to deep learning method, the problem has made great progress in a few recent studies [9], [10]. With the feature extraction technique by 2D/3D convolutional neural network (CNN) model and combined with recurrent neural network (RNN), it supports the storage of sequential information in sequence data, of which

prominent are long short-term memory (LSTM) [11]–[13] and gated recurrent unit (GRU) [14], [15] models, combined with natural language processing models to generate descriptions. Another challenge in the video activity description problem is that the sequences of activities in the video have remarkable features, the combination of these characteristics helps to assess the comprehensiveness of the activity. Moreover, the processing process with video data is difficult, the idea of using RNN model will increase the computational cost. To improve the processing speed, many researchers had tried using additional mechanism such as attention [16], [17], transformers [18] which obviously bring high efficiency to the result.

One more point that currently, there have been many studies on the problem of describing actions in languages such as English, German, Japanese, and Chinese. However, there has been no research with action's describing in Vietnamese. In the paper, we are building an automatic model describing actions through the camera in Vietnamese.

In this paper, we propose the three most advanced deep learning models today, which are sequence-to-sequence model based on RNN, sequence-to-sequence model with attention and transformer model combined with language processing in Vietnamese to generate meaningful Vietnamese video captions. Through these three models, we evaluate the accuracy, computational cost, and model efficiency, thereby providing future development directions when the data set is larger and the actions in the video are described in a variety of ways than. To train these models, we use Microsoft research video description corpus (MSVD) dataset including 1,970 videos [19] and 85,550 caption sentences translated into Vietnamese.

2. RESEARCH METHOD

2.1. Related algorithms

Convolutional neural network are deep learning algorithm with quite good results in most of machine vision problems such as classification, object recognition, CNN had been applied to many areas which require high accuracy such as medicine, industry, robotics [20]–[23]. CNN is basically a type of straight-forward artificial neural network, in which the model architecture consists of blocks containing convolution and pooling layers, the last layer is a fully connected layer as shown in Figure 1.

In CNN, convolutional layer and pooling layer play a role in image feature extraction: convolutional layers are designed based on convolution multiplication between small sections of the image with a filter to extract image features. The filter is then calculated based on the back-propagation technique. Pooling layers reduce the number of parameters without changing the network architecture, increasing computational speed. Fully connected layer will act as a classifier of previously extracted features, this layer will give the probability of an object in the image.

CNN is an efficient method to extract image features with high accuracy and minimal execution time compared to traditional neural networks. Some well-known models are researched and built based on the large ImageNet dataset and have significantly improved performance such as VGGNet [21], GoogleNet [22], ResNet [23], EfficientNet [24]. These state-of-the-art models provide better computational efficiency than using self-built models with high cost but low efficiency. In this paper, we propose to use the ResNet model, when the network model is deeper, the ResNet model can limit the vanishing gradient problem, thereby increasing the model's performance.

Recurrent neural network: RNNs are state of the art algorithm for problems with sequence data and achieves many good results in natural language processing problems such as machine translation, question answering, text-to-speed, time series prediction [9], [10]. RNNs allow connecting relationships by a memory at the input of each state to store information from previous computation steps. Hidden states are calculated using formula [9]:

$$s_t = f(Ux_t + Ws_{t-1}) \quad (1)$$

where s_t is new hidden state, s_{t-1} is previous hidden state; x_t is the input of step, U, W is weight matrices, f is activation functions.

One of the improvements in RNN model is LSTM, GRU [11]–[15]. The main component of the LSTM is the cell, which decides what information should be extracted from the input and how important the previous state is based on gate groups. These gates select appropriate information to remember based on previous context by removing unnecessary state information and adding some necessary ones with activation functions such as sigmoid and tanh. Thus, unlike traditional RNNs, LSTMs can adjust information at each step instead of overwriting states. It means the LSTM is possible to store long-range dependency information. Furthermore, the LSTM model also eliminates vanishing gradient encountered by the RNN. The GRU model has a similar architecture to the LSTM model but is simpler by removing ports and having no separate memory components to increase computational efficiency. In this paper, we propose LSTM and GRU network models.

Attention mechanism: attention was proposed by Bahdanau [17] to improve the efficiency of sequence-to-sequence problems such as machine translation, in addition, this technique is also used in a number of other fields such as computer vision. Whereas RNNs process information sequentially from previous information, attention will focus on the importance of each piece of data. The attention mechanism allows to retain interesting information in states of the input sequence by weights in hidden states. These weights will be aggregated for predicting the next information. This means a lot in the selection of specific elements that fit the context for the output string. Attention model A works with a set of key-value pairs K, V and query Q such that [25]:

$$A(Q, K, V) = \sum_i p(\alpha(K_i Q)) * V_i \quad (2)$$

where $\alpha(\cdot)$ is alignment functions that are designed for specific use-cases; $p(\cdot)$ is distribution function, where SoftMax function use to be used; K_i, V_i respectively the key-values, key of the string in hidden state i .

There are many other versions of attention were applied to improve each specific sequence-to-sequence problem with high efficiency such as: dot-product attention, adaptive attention, multi-level attention, multi-head attention and self-attention [17], [26], [27] In our problem, we propose scaled-dot-product attention for sequence-to-sequence model and multi-head attention for transformer model.

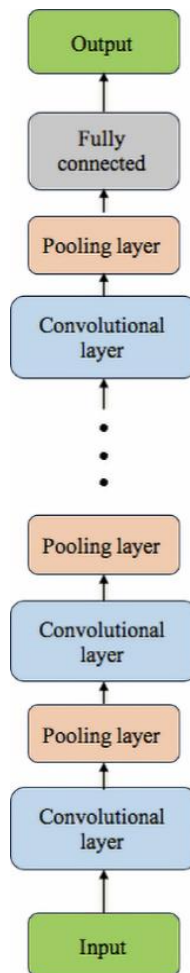


Figure 1. The CNN model

2.2. Model for describing action via camera

Based on deep learning techniques, we propose three models to solve the camera action description problem: sequence-to-sequence model based on RNN, sequence-to-sequence model with attention and transformer model.

2.2.1. Sequence-to-sequence model based on RNN

The sequence-to-sequence model based on RNN is built based on the encoder-decoder architecture. The model is widely applied in the problem of natural language processing (NLP) such as generating output from a given input: machine translation, question answering, text summarization, and text-to-speech. The model includes: the encoder is responsible for encoding input strings into a fixed vector. Usually, the encoder uses RNN network architecture, which is LSTM or GRU. The RNN will iterate over the elements of input string. At each moment, the RNN will convert the element in the string and the state h_t with the f transform to store the information at state h_t [28]:

$$h_t = f(x_t, h_{t-1}) \tag{3}$$

After the encoding process, the vector context is obtained through the custom function q [28]:

$$c = q(h_1, \dots, h_T) \tag{4}$$

The decoder usually uses a different RNN network for decoding. Output in state t' is decoded by function g with information obtained from the output $y_{t'-1}$ of state $t' - 1$, context vector c and status vector $s_{t'-1}$ [28]:

$$s_{t'} = g(y_{t'-1}, c, s_{t'-1}) \tag{5}$$

For object action's description in a video, we build a sequence-to-sequence model as shown in Figure 2. Image sequences, after being extracted features, will go through the encoder model built with an LSTM class for storing information from the previous image frames, which support to predict the actions of the next image frames. In the decoder model, after the sequence of image features go through the encoder model, there will be a context vector containing characteristic information. This feature vector will be combined with inputs (context vectors) and sent to the LSTM layers to decode the information.

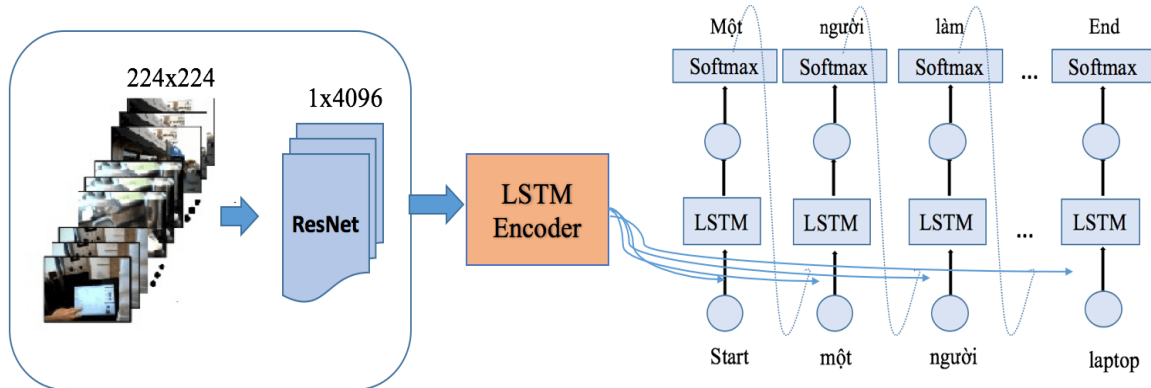


Figure 2. Sequence-to-sequence model is based on RNN

2.2.2. Sequence-to-sequence model with attention

The sequence-to-sequence model uses only one feature vector encoded over a sequence of information extracted from the image frames, which will lose a lot of notably important information in the states. To limit this, another improvement of the sequence-to-sequence model has been done when we combine it with attention mechanism. We propose scale dot-product attention for the model. Attention model is defined as in (6) [25], [28]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

where $Q \in \mathbb{R}^{T \times d_k}$, $K \in \mathbb{R}^{T \times d_k}$, $V \in \mathbb{R}^{T \times d_v}$ respectively the query values, key, and value of the string; d_k are length sizes of Q , K , V , respectively. We use GRU model instead of LSTM. The GRU model has the same architecture as the LSTM model but is simpler to increase computation efficiency. The model is depicted in Figure 3.

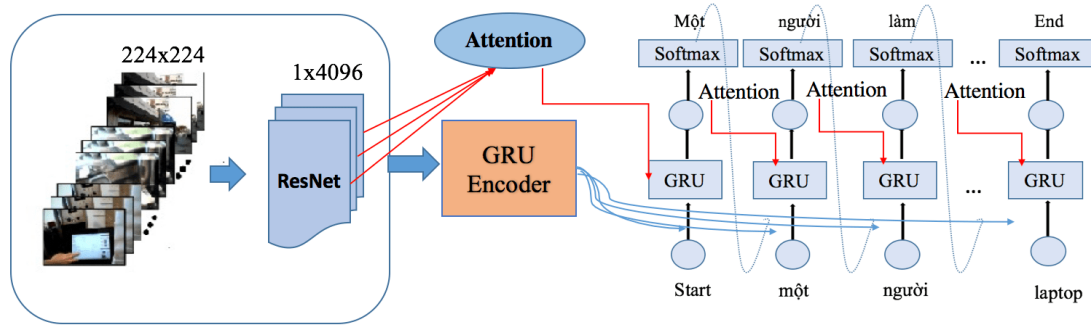


Figure 3. Sequence-to-sequence model with attention

2.2.3. Transformer model

In the transformer model, each image frame after feature extraction is passed through the self-attention mechanism to encode context information. In this encoder step, LSTM does not use regression neural network to store previous dependent information like in sequence-to-sequence model but uses attention technique to store weight-based information at each moment.

For sequence data, the elements in the sequence have a lot of information in many different aspects that need attention, taking the average weight will lose the meaning of the components. The use of multi-head attention allows the attention mechanism to be calculated independently, parallel to each other, and connected linearly with each other to store the attention information of different parts in the chain. Specially, for each query, key, value Q, K, V we convert into h sub-queries, sub-key, sub-queries, sub-key, sub-value, and compute head value by using dot product attention to combine the values. We then connect the heads to W^0 - the end weight matrix. The multi-head attention formula is defined as (7) [26], [27]:

$$MultiHead(Q, K, V) = Concat_{i=1,2,...,h}(head_i)W^0 \tag{7}$$

where,

$$head_i(Q, K, V) = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{8}$$

The transformer model is built by combining multi-head attention layers to increase the computational efficiency of the model.

At the feed-forward layer, we use the feed-forward neural network featured extraction. In this way, getting updated gradients at a time is independent, and learning potential dependencies are easy. The feed-forward neural network is a linear neural network using the rectified linear unit (ReLU) activation function. The network is defined as (9) [26], [27]:

$$FFN(X) = W_2 ReLU(X, W_1) \tag{9}$$

where W_2, W_1 are weighted matrices.

In the encoder and decoder models, we only use multi-head attention and feed-forward neural network without any additional convolutional or recurrent layer. Hence, to store information serially, we use an additional encoding method which is a positional encoder. The method is defined as in (10) and (11) [26], [27]:

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}}) \tag{10}$$

$$PE(pos, 2i + 1) = cos(pos/10000^{2i/d_{model}}) \tag{11}$$

where pos is the position of the image frame in the video image sequence or the position of the word in the sentence and i is the size of the embedding vector.

Thus, the transformer model architecture can support parallel computation between words or image frames instead of sequential processing like LSTM model in traditional sequence-to-sequence model. This greatly increases the efficiency of the model. The transformer model for the problem of describing actions through video is depicted in Figure 4.

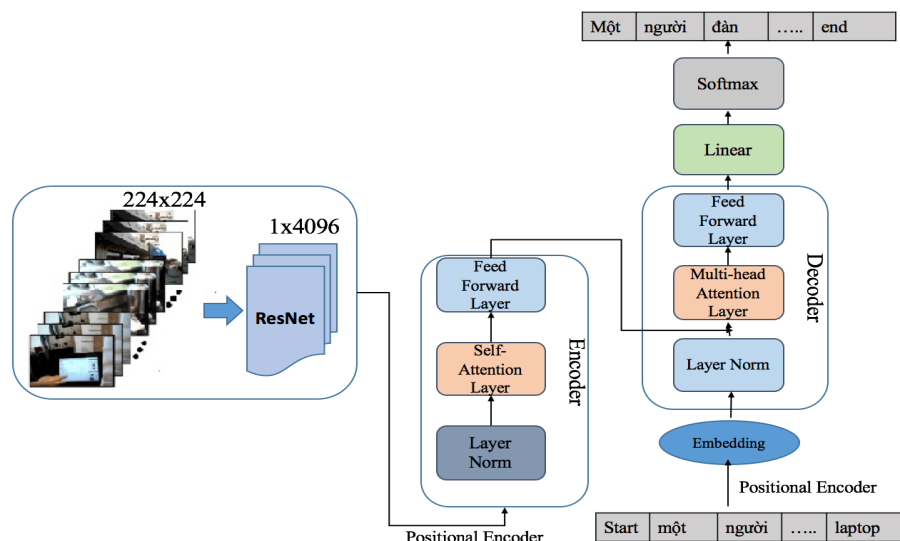


Figure 4. Transformer model

2.3. Model evaluation

In these three models, the sequence-to-sequence model based on RNN has a deep network architecture with RNN, which helps to extract a lot of information about objects but limits the ability to store important information of objects in the sequence. Combining with attention will overcome this limitation. On the other hand, transformer model does not use RNN architecture, it has a simpler network architecture, which can increase the speed of the training process.

To evaluate the model accuracy, we compared the generated annotation with the video label in the test dataset. We evaluate the model based on the following criteria:

- BLEU: The BLEU is widely used in machine translation problem to measure the quality of the output sequence. The BLEU operates based on the principle that any n-grams in the predictive sequence, BLEU will evaluate whether it appears in the label sequence or not [29].
- METEOR: The METEOR is designed to measure similar to BLEU. Besides, the METEOR adds some extra steps like looking at synonyms, comparing word roots. The METEOR is designed very clearly, suitable for comparing sentences with each other [30].
- ROUGE: Another popular criterion is ROUGE. ROUGE focuses on reminiscent, while BLEU focuses on accuracy. There are many types of Rouge such as: ROUGE-N: measures the number of n-grams in common between the predicted sequence and the label sequence, for example ROUGE1 measures the number of unigrams between two strings. ROUGE-L measures the longest common subsequence between the two prediction sequences and the sample sequence [31].

3. RESULTS AND DISCUSSION

3.1. Dataset

The dataset we use is Microsoft's video dataset - MSVD and the corresponding subtitle files to describe those videos. The dataset includes 1970 videos describing the activities in nature of humans and animals. The caption file includes 122,000 subtitles, each video has many different captions respectively, and these captions have been described in many different languages. We filter out 85550 English captions and translate all into Vietnamese. To prepare for model training process, we divided the dataset into 3 parts: which are 1500 videos were used for training the model, 100 videos were used for the validation dataset and 370 videos we used for testing our built model.

3.2. Implementation details

The dataset consists of 2 data types: video and text. To prepare the data to be put into the model, we treat these 2 types of data:

The image is cut from the video: The video input was recorded by camera, we cut a series of images from the video. These frames are cut successively into and sequentially at a certain rate. Each input video will be cut into 80 image frames, then processed to return the input size of 224x224. The extracted input images were characterized by the ResNet pretrain model with an output size of vector 4096.

Caption: For text data, we clean it up with operations, like converting uppercase to lowercase, removing special characters, removing words that contain digits... Next, we use the Underthesea library to separate words for Vietnamese [32], [33]. In the process of building dictionaries for text dataset, in order to optimize the dictionary, we remove words with little meaning, and words that appear less frequently in the dataset. From a data set of 85550 sentences, we created a dictionary containing 8885 Vietnamese words. After optimizing the dictionary, we obtained 3040 words. Each caption is added with the starting and ending characters (startseq, endseq) in preparation for processing the string in the model. These two special characters will also be added to the dictionary representing the beginning and ending of a sentence.

Each word in the dictionary, after being extracted from the dataset, will be represented as a dictionary-length one-hot vector. We also use Fasttext model for word embedding [34]. The Fasttext model is based on neural network model, helps to reduce the length of word vectors, captures the meaning of similar words, in addition, the model also works well when encountering rare words [35]. In the problem, we reduce the length of the vector from 3040 to 300.

Model architecture: to train the model, we use the Google Colab tool to take advantage of the GPU feature. For the sequence-to-sequence model based on RNN and the sequence-to-sequence model with attention, we design model with values of hyperparameters such as: the number of LSTM and GRU units in the encoder and decoder model is 1024 cells, the learning rate value is 0.0001, and the batch size is 100. For the transformer model, we design the model consisting of the encoder layer, including 6 classes of norm and self-attention, in the decoder layer, including 8 norm and multi-head attention classes. The learning rate value is adjusted during training to increase the accuracy of the model.

3.3. Results and analysis

The training results of 3 models on Google Colab are depicted in Figure 5. We can see that the plot for loss function is smooth, the training process of three models converged well. In Figure 5(a), the model sequence-to-sequence model based on RNN converged after 25 epochs. In Figure 5(b), the sequence-to-sequence model with attention converged after 15 epochs. In Figure 5(c), the transformer converged after 15 epochs.

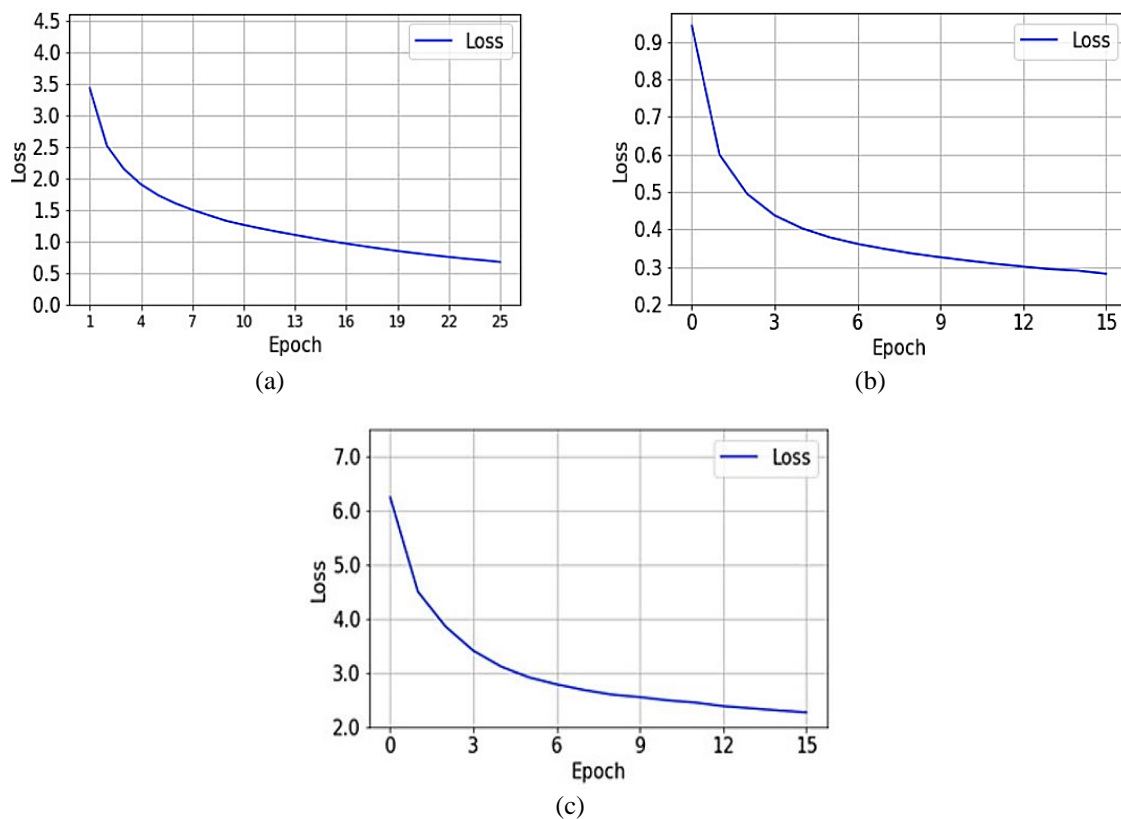


Figure 5. Loss function of three models over training epochs (a) sequence-to-sequence model based on RNN, (b) Sequence-to-sequence model with attention, and (c) transformer model

3.3.1. Evaluate the accuracy of the model

The accuracy of the model is evaluated based on the train and test dataset. Accuracy scores are calculated on 4 scores of BLEU, METEOR, ROUGE1, and ROUGEL. Figure 6(a) gives accuracy scores in the training process of the sequence-to-sequence model based on RNN. Figure 6(b) gives accuracy scores of the sequence-to-sequence model with attention. Figure 6(c) gives accuracy scores of the transformer model. Thus, through Figure 6, it shows that after the training process of 3 models, the sequence-to-sequence model based on RNN achieved the lowest accuracy, the transformer model achieved the highest accuracy.

To evaluate the accuracy of the model after training, we test all 3 models on a test dataset of 370 videos. Accuracy of three models for the test dataset are described in Table 1. The test results are also based on 4 evaluation scores, namely BLEU, METEOR, ROUGE1, and ROUGEL. The results in Table 1 show that: For the BLEU score, the sequence-to-sequence model based on RNN and sequence-to-sequence model with attention are not significantly different. For the METEOR score, the sequence-to-sequence model based on RNN gives better results than the sequence-to-sequence model with attention. For the evaluation scores ROUGE1, ROUGEL, sequence-to-sequence model with attention gives better results than sequence-to-sequence based on RNN. Particularly, the transformer model achieved the highest accuracy on all 4 evaluation points compared to the other 2 models.

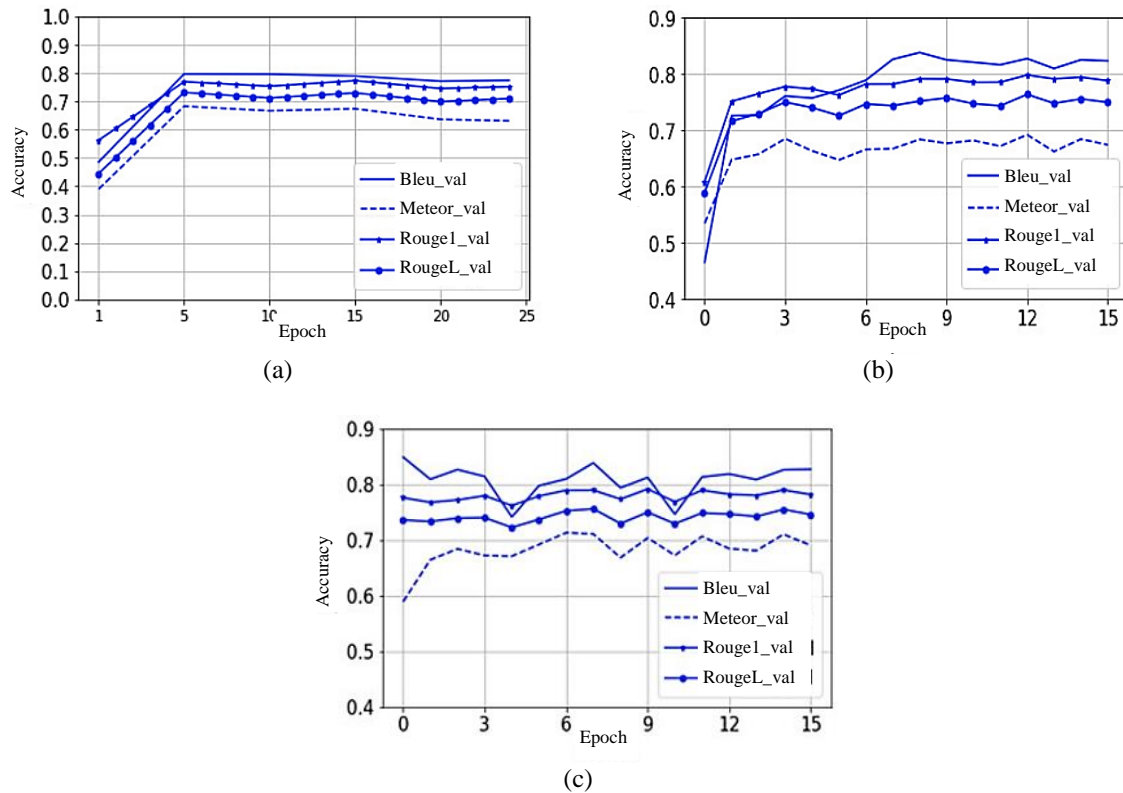


Figure 6. Accuracy of three models on train dataset over training epochs based on predicting BLEU (solid line), METEOR (dashed line), ROUGE1 (star line) and ROUGEL scores (dotted line) (a) sequence-to-sequence model based on RNN, (b) sequence-to-sequence model with attention, and (c) transformer model

Table 1. Accuracy of the models on test dataset

Model	Bleu	Meteor	Rouge1	RougeL
<i>Seq2seq with RNN</i>	0.789	0.653	0.758	0.716
<i>Seq2seq+Attention</i>	0.82	0.66	0.78	0.75
<i>Transformer</i>	0.84	0.70	0.78	0.76

3.3.2. Evaluate the performance of the model

In Table 2, based on training time of the models we can evaluate the effectiveness of the model. Easily to see that the training process of the sequence-to-sequence model based on RNN takes a long time to train, the transformer model has the lowest training time. Thus, based on the evaluation of the model's

accuracy and performance, the sequence-to-sequence model based on RNN and the sequence-to-sequence model with attention have no difference in accuracy. However, the performance efficiency of the sequence to sequence model based on RNN combined with attention is higher. Meanwhile, the transformer model has a simpler network architecture that achieves the highest performance and accuracy compared to the remaining models.

Table 2. Execution time of the models

Model	Number of converging epochs	Execution time per epoch(min)
<i>Seq2seq with RNN</i>	25	8.66
<i>Seq2seq + Attention</i>	15	8
<i>Transformer</i>	15	6

3.3.3. Some comparison results between models

In Figure 7(a), the sequence-to-sequence model based on the RNN predicts the comment: “*một con mèo đang chơi bowling (a cat is playing bowling)*”, the sequence-to-sequence model with attention predicts the comment: “*một con mèo đang chơi với một quả bóng (a cat is playing with a ball)*”, the transformer model predicts the caption: “*một con mèo đang chơi với một quả bóng (a cat is playing with a ball)*”. In Figure 7(b), the sequence-to-sequence model based RNN predicts the comment “*một em bé đang cười (a baby is smiling)*”, the sequence-to-sequence model with attention predicts the comment: “*một em bé đang cười (a baby is smiling)*”, the transformer model predict the caption as “*một em bé đang ngồi trên ghế sofa cười (a baby is sitting on the sofa end smiling)*”.



(a)



(b)

Figure 7. Video used for test models (a) a cat is playing bowling and (b) a baby is smiling

3.4. Deploy the model to the application

Based on the experimental results, we choose the transformer model to deploy on the system. We deploy the application using the Python package Tkinter, which allows support for GUI programming, and use the PIL image to crop the video from the camera with 8-9 s duration. Then the video is imported into the model for processing and generating the corresponding caption. In Figure 8(a), the application generate caption as "A woman is cutting onions", in Figure 8(b), the application generate caption as "A group of people dancing".



(a)



(b)

Figure 8. Deploy the model to the application, (a) the application generate caption as "a woman is cutting onions" and (b) the application generate caption as "a group of people dancing"

4. CONCLUSION

From the videos obtained from camera, we have built a model that automatically generates captions describing the content of the video in Vietnamese. We use 3 deep learning models for training: sequence-to-sequence model based on RNN, sequence-to-sequence model with attention and transformer model. Based on the comparison results, the sequence-to-sequence model based on RNN has a rather cumbersome network architecture, takes a lot of time to implement, and when combined with attention, it is more improved but not significant. Meanwhile, the transformer model with a simpler network architecture has improved the execution speed and accuracy, bringing optimal results for the problem. This is also the model chosen to deploy on systems with limited hardware devices. Some limitations of the approach to the problem is that when using deep learning techniques, we need a large amount of data for training to ensure accuracy. In addition, when training deep learning models for the problem, it requires powerful hardware to increase computation speed due to the long training time, and it has to be done many times to find the optimal model

architecture. In the future, we will build more diverse and larger datasets in Vietnamese, improve the model architecture to achieve even more optimal results.




REFERENCES

- [1] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence -- video to text," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 4534–4542, doi: 10.1109/ICCV.2015.515.
- [2] A. Puscasiu, A. Fanca, D.-I. Gota, and H. Valean, "Automated image captioning," in *2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, May 2020, pp. 1–6, doi: 10.1109/AQTR49680.2020.9129930.
- [3] H. Sharma, M. Agrahari, S. K. Singh, M. Firoj, and R. K. Mishra, "Image captioning: A comprehensive survey," in *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, Feb. 2020, pp. 325–328, doi: 10.1109/PARC49193.2020.236619.
- [4] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1151–1159, doi: 10.1109/CVPR.2017.128.
- [5] S. Islam, A. Dash, A. Seum, A. H. Raj, T. Hossain, and F. M. Shah, "Exploring video captioning techniques: a comprehensive survey on deep learning methods," *SN Computer Science*, vol. 2, no. 2, pp. 120–148, Apr. 2021, doi: 10.1007/s42979-021-00487-x.
- [6] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *International Journal of Computer Vision*, vol. 50, pp. 171–184, 2002, doi: 10.1023/A:1020346032608.
- [7] Z. Wu, T. Yao, Y. Fu, and Y.-G. Jiang, "Deep learning for video classification and captioning," in *Frontiers of Multimedia Research*, ACM, 2017, pp. 3–29.
- [8] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1494–1504, doi: 10.3115/v1/N15-1173.
- [9] S. Hershey et al., "CNN architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 131–135, doi: 10.1109/ICASSP.2017.7952132.
- [10] S. Chen and Y.-G. Jiang, "Motion guided spatial attention for video captioning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Jul. 2019, vol. 33, pp. 8191–8198, doi: 10.1609/aaai.v33i01.33018191.
- [11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [12] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1029–1038, doi: 10.1109/CVPR.2016.117.
- [13] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, Nov. 2016.
- [14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Workshop on Deep Learning*, Dec. 2014.
- [15] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.
- [16] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, Sep. 2017, doi: 10.1109/TMM.2017.2729019.
- [17] A. Vaswani et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009, Jun. 2017.
- [18] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8739–8748, doi: 10.1109/CVPR.2018.00911.
- [19] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 190–200.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2015.
- [22] C. Szegedy et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [24] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019.
- [25] C. Hori et al., "Attention-based multimodal fusion for video description," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 4203–4212, doi: 10.1109/ICCV.2017.450.
- [26] A. Wu, Y. Han, Y. Yang, Q. Hu, and F. Wu, "Convolutional reconstruction-to-sequence for video captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4299–4308, Nov. 2020, doi: 10.1109/TCSVT.2019.2956593.
- [27] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4467–4480, Dec. 2020, doi: 10.1109/TCSVT.2019.2947482.
- [28] Z. Liu, T. Chen, E. Ding, Y. Liu, and W. Yu, "Attention-based convolutional LSTM for describing video," *IEEE Access*, vol. 8, pp. 133713–133724, 2020, doi: 10.1109/ACCESS.2020.3010872.
- [29] 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 on Association for Computational Linguistics - ACL '02*, 2001, p. 311, doi: 10.3115/1073083.1073135.




- [30] M. Denkowski and A. Lavie, "Meteor universal: language specific translation evaluation for any target language," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014, pp. 376–380, doi: 10.3115/v1/W14-3348.
- [31] C.-Y. Lin, "ROUGE: a package for automatic evaluation of summaries," in *Association for Computational Linguistics*, 2004, pp. 74–81.
- [32] T.-H. H. Pham, X.-K. K. Pham, T.-A. A. Nguyen, and P. Le-Hong, "NNVLP: a neural network-based Vietnamese language processing toolkit," *8th International Joint Conference on Natural Language Processing - Proceedings of the IJCNLP 2017, System Demonstrations*. pp. 37–40, Aug. 2017.
- [33] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese Natural Language Processing Toolkit," *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, pp. 56–60, Jan. 2018, doi: 10.18653/v1/n18-5012.
- [34] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," *International Conference on Learning Representations 2017*. Dec. 2016, arxiv.org/abs/1612.03651.
- [35] S. Bodapati, S. Gella, K. Bhattacharjee, and Y. Al-Onaizan, "Neural word decomposition models for abusive language detection," in *Proceedings of the Third Workshop on Abusive Language Online*, 2019, pp. 135–145, doi: 10.18653/v1/W19-3515.

BIOGRAPHIES OF AUTHORS






Dang Thi Phuc    received the Specialist degree from the Lomonosov Moscow State University, Moscow, Russia in 2008. In 2018 she received Ph.D. degree in System analysis, Control and Information Processing from Peoples' Friendship University of Russia, Moscow, Russia. Since 2018, she is a lecturer of Faculty of Information Technology, Industry University of Ho Chi Minh City, Vietnam. Her research interests include Machine Learning, Computer Vision, NLP, Deep Learning, Operator network. Email: phucdt@iuh.edu.vn.






Tran Quang Trieu    is a student at Industrial University of Ho Chi Minh City, Vietnam. His study interests include Machine Learning, Computer Vision, NLP, Deep Learning. Email: trieutranq2000@gmail.com.



Nguyen Van Tinh    is a student at Industrial University of Ho Chi Minh City, Vietnam. His study interests include Machine Learning, Computer Vision, NLP, Deep Learning. Email: tinh2kqb@gmail.com.



Dau Sy Hieu    received the Specialist degree from the Lomonosov Moscow State University, Moscow, Russia in 2009. In 2015 he received Ph.D. degree in Physical Condensation State from Peoples' Friendship University of Russia, Moscow, Russia. Since 2009, he is a lecturer Faculty of Applied Science, University of Technology - Viet Nam National University HCM city, Vietnam. His research interests include Condensation State, Optical system design, Machine Learning, Computer Vision, NLP, Deep Learning. Email: dausyhieu@hcmut.edu.vn.