

**CAN THO UNIVERSITY  
COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY**



**SPECIALIZED THESIS  
(HIGH-QUALITY PROGRAM)**

**BUILD DESCRIPTIONS FOR IMAGES  
WITH INCEPTION-V3 AND TRANSFORMER**

**Student: Nguyen Thi My Khanh  
Student ID: B1910657  
Class: 2019-2023 (K45)  
Advisor: Dr. Tran Cong An**

**Can Tho, 12/2022**

**CAN THO UNIVERSITY  
COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY  
DEPARTMENT OF INFORMATION TECHNOLOGY**



**SPECIALIZED THESIS  
(HIGH-QUALITY PROGRAM)**

**BUILD DESCRIPTIONS FOR IMAGES  
WITH INCEPTION-V3 AND TRANSFORMER**

**Student: Nguyen Thi My Khanh  
Student ID: B1910657  
Class: 2019-2023 (K45)  
Advisor: Dr. Tran Cong An**

**Can Tho, 11/2022**

## **ACKNOWLEDGMENTS**

First of all, I would like to express my deep gratitude to Dr. Tran Cong An, who directly guided and oriented me to complete this graduation thesis. He spent time and created favorable conditions in all aspects so that I could complete the topic.

Can Tho, December ..., 2022  
Student

Nguyen Thi My Khanh

## TABLE OF CONTENT

TABLE OF CONTENT .....	2
LIST OF FIGURES .....	3
LIST OF TABLE .....	3
LIST OF ABBREVIATIONS .....	4
ABSTRACT .....	5
CHAPTER 1. INTRODUCTION .....	6
1.1. Statement of the problem .....	6
1.2. Related work .....	6
1.3. The purpose of the topic .....	7
1.4. Object and scope research .....	7
1.5. Research approach and methods .....	7
1.6. Topic outline .....	8
CHAPTER 2: LITERATURE REVIEW .....	9
2.1. Inception-V3 model .....	9
2.2.1. Factorized convolutions .....	9
2.1.2. Factorization into smaller convolutions .....	9
2.1.3 Dataization of asymmetric filter .....	10
2.1.4. Auxiliary Classifier .....	11
2.1.5. Grid Size Reduction .....	12
2.2. Transformer model .....	12
2.2.1. Self-attention mechanism .....	12
2.2.2. Masked Self-Attention Mechanism .....	13
2.2.3. Multi-head attention .....	14
2.2.4. Transformer Architecture .....	15
2.3. Evaluation methods .....	18
CHAPTER 3. METHODS OF IMPLEMENTATION .....	20
3.1. Description of the problem .....	20
3.2. Build descriptions for images .....	20
CHAPTER 4. EXPERIMENT .....	23
4.1. Dataset .....	23
4.2. Data preprocessing .....	23
4.3. Training .....	23
4.4. Accuracy rating .....	24
4.5. Experimental results .....	24
CHAPTER 5. CONCLUSION .....	26
5.1. Achieved results .....	26
5.2. Development direction .....	26
REFERENCES .....	27
APPENDIX .....	28

## LIST OF FIGURES

Figure 1 . <i>Mini-network replacing the <math>5 \times 5</math> convolutions. [6]</i> .....	9
Figure 2 . <i>Inception modules where each <math>5 \times 5</math> convolution is replaced by two <math>3 \times 3</math> convolution [6]</i> .....	10
Figure 3 . <i>Replace <math>3 \times 3</math> convolution with asymmetric[6]</i> .....	10
Figure 4 . <i>Inception modules after the factorization of the <math>n \times n</math> convolutions.[6]</i> ....	11
Figure 5 . <i>The classifier acts as a regularization mechanism [6]</i> .....	11
Figure 6 . <i>Grid size reduction diagram [6]</i> .....	12
Figure 7 . <i>Brief self-attention model</i> .....	13
Figure 8 . <i>Masked Self-Attention mechanism</i> .....	14
Figure 9 . <i>Multi-head attention mechanism</i> .....	14
Figure 10 . <i>Transformer model architecture [8]</i> .....	15
Figure 11 . <i>General diagram of the computational process of the Encoder model</i> .....	17
Figure 12 . <i>The process of creating descriptions for image</i> .....	21
Figure 13 . <i>Inception-V3 model extracting image features</i> .....	21
Figure 14 . <i>Flickr8k dataset</i> .....	23

## LIST OF TABLE

Table 1 . <i>Table of BLEU evaluation results of the descriptive sentence build model</i> .	24
Table 2 . <i>Some test results of the model</i> .....	25

### LIST OF ABBREVIATIONS

No.	Abbreviation	Origin word
1	AI	Artificial Intelligence
2	CV	Computer Vision
3	NLP	Natural Language Processing
4	RNN	Recurrent Neural Network
5	CNN	Convolutional Neural Network
6	FFN	Feed-Forward Networks
7	YOLO	You Only Look Once

## **ABSTRACT**

Building descriptive sentences for images is one of the important problems in the field of computer vision and natural language processing. In this thesis, deep learning models based on merge architecture are used to generate descriptive sentences for images. The merge Architecture combines the image features extracted from the Inception-v3 model with the description sentences encoded by the Transformer model. The model uses images and descriptive sentences from the Flickr8k dataset to train. The results of the model evaluation using the Flickr8k dataset (BLEU-1: 24.94, BLEU-2: 12.43, BLEU-3: 5.40, BLEU-4: 5.27) are lower than those of research with the same method of Shah, et al [1] (BLEU-1: 54.30, BLEU-2: 44.50, BLEU-3: 36.20, BLEU-4: 29.40)

## **CHAPTER 1. INTRODUCTION**

### **1.1. Statement of the problem**

Nowadays, Artificial Intelligence (AI) is increasingly popular and contributes to profoundly changing many aspects of life. In particular, Computer Vision (CV) and Natural Language Processing (NLP) are two important areas of AI including methods of image acquisition, processing, analysis, speech recognition, decomposition words, analyze sentences and generate automatic description sentences for images.

Deep Learning Network is the field of study of algorithms, computer programs for computers to learn and make predictions like humans. It is applied in many different fields such as science, engineering, other areas of life, as well as applications in classification, object detection and descriptive sentence generation.

Creating descriptive sentences for images is the process of combining CV and NLP to recognize the context and content of the image and describe it through the object and location. From there, create descriptive sentences in natural language. Some typical applications in creating descriptive sentences for photos are image indexing, assisting people with disabilities, implementing human-computer interaction, applying to virtual assistants.

In recent studies, the problem of creating descriptions for images has received a lot of attention and is applied in many different languages. However, the use of Inceptionv3 and Transformer models for model prediction is not yet widespread. Therefore, in this thesis, Inceptionv3 will be used to extract features of images and Transformer predicts description based on Flickr8k dataset.

### **1.2. Related work**

Here are some studies related to creating descriptive sentences for image.

Xinlei Chen et al. [2] use a cyclic neural network to bi-directional mapping between images and their sentence-based descriptions. This method is allow to create a new question with 1 image. Uses a Recurrent Neural Network (RNN) to feed new light information from when a word is generated or read an updated live projection. The model was developed and evaluated on all PASCAL 1K, Flickr8k, Flicke30k and MS COCO files. Data usage is done in 2 steps: Using CoreNLP tool Stanfoed to encrypt sentences and lowercase all letters. When compared to human-generated subtitles, auto-generated subtitles are 19.8% more popular.

Khang Nhut et al. [3] discusses a facial expression recognition model and a description generation model to build descriptive sentences for images and facial expressions of people in images. Experimental results on the Flickr8k dataset in



Vietnamese achieve BLEU-1, BLEU-2, BLEU-3, BLEU-4 scores of 0.628; 0.425; 0.280; and 0.174, respectively.<sup>i</sup>

Xiong et al. [4] proposes a hierarchical Transformer based medical imaging report generation model. Our proposed model consists of two parts: (1) An Image Encoder extracts heuristic visual features by a bottom-up attention mechanism; (2) a non-recurrent Captioning Decoder improves the computational efficiency by parallel computation. The former identifies regions of interest via a bottom-up attention module and extracts top-down visual features. Then the Transformer based captioning decoder generates a coherent paragraph of medical imaging report. The proposed model is trained by using a self-critical reinforcement learning method. The proposed model on publicly available datasets of IU X-ray. The experiment results show that proposed model has improved the performance in BLEU-1 by more than 50% compared with other state-of-the-art image captioning methods.

Wenhao Jiang et al. [5] use an input image is encoded by a convolutional neural network (CNN) and then translated into natural language with a recurrent neural network (RNN). The existing models counting on this framework employ only one kind of CNNs, extit{e.g.}, ResNet or Inception-X, which describes the image contents from only one specific view point. Thus, the semantic meaning of the input image cannot be comprehensively understood, which restricts improving the performance. To exploit the complementary information from multiple encoders, they proposed a novel recurrent fusion network (RFNet) for the image captioning task. The fusion process in their model can exploit the interactions among the outputs of the image encoders and generate new compact and informative representations for the decoder. Experiments on the MSCOCO dataset demonstrate the effectiveness of their proposed RFNet, which sets a new state-of-the-art for image captioning.

### **1.3. The purpose of the topic**

The goal of the project is to study a method to build descriptive sentences for images automatically based on the object displayed on the image.

### **1.4. Object and scope research**

❖ Research object:

The object of research is the Inception-V3 and Transformer models.

❖ Research scope:

The Flickr8k dataset is used to build description sentences for images.

### **1.5. Research approach and methods**

❖ General approach

Based on references. Then proceed to build and test the model on google colaboratory

❖ Research methods

- Research theory from journals, articles, websites in related fields.
- Implement and evaluate the results achieved.

**1.6. Topic outline**

- Chapter 1: Introduction - overview of the year thesis.
- Chapter 2: Literature review - describes the theory used in the thesis.
- Chapter 3. Methods of implementation
- Chapter 4. Experiment
- Chapter 5. Conclusion of achieved results and development direction

## CHAPTER 2: LITERATURE REVIEW

### 2.1. Inception-V3 model

Inception-v3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for Googlenet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge.

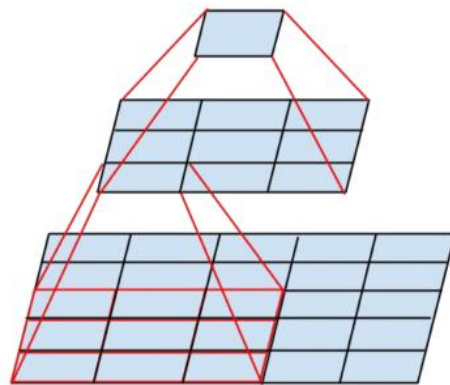
Compared to VGG, the Inception Network is more computationally efficient, both in terms of the number of parameters generated by the network and the resulting cost savings (memory and other resources). If any changes are made to the Inception network, it is necessary to ensure that the computational advantages are not lost. In the Inception-V3 model, several techniques for network optimization have been proposed to loosen the constraints for easier model adaptation. The techniques include: factorized convolutions, regularization, dimension reduction, and parallelized computations. The Inception-V3 network architecture is presented as follows:

#### 2.2.1. Factorized convolutions

This reduces the number of parameters involved in a network. It also checks the efficiency of the network.

#### 2.1.2. Factorization into smaller convolutions

Replacing larger convolution methods with smaller convolutions will certainly lead to faster training. Use two 3x3 convolutions instead for 5x5 convolutions. The 5x5 convolution has 25 parameters in size instead 2 3x3 filters have only 18 ( $3*3+3*3$ ) parameters. The simulation architecture is shown in Figure 1 .



*Figure 1. Mini-network replacing the  $5 \times 5$  convolutions. [6]*

With this technique, one of the new inception modules will be replaced as shown in Figure 2.

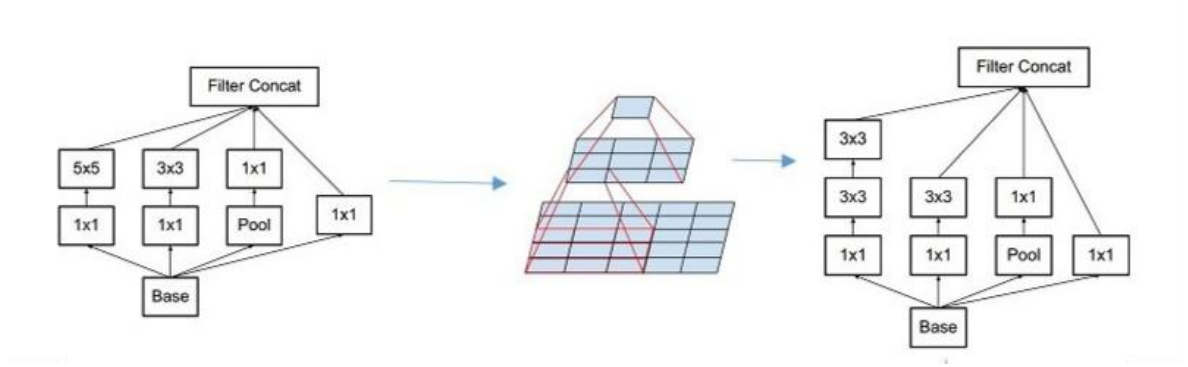


Figure 2. Inception modules where each  $5 \times 5$  convolution is replaced by two  $3 \times 3$  convolution [6]

### 2.1.3 Dataization of asymmetric filter

The  $3 \times 3$  convolution will be replaced by a  $1 \times 3$  and  $3 \times 1$  convolution

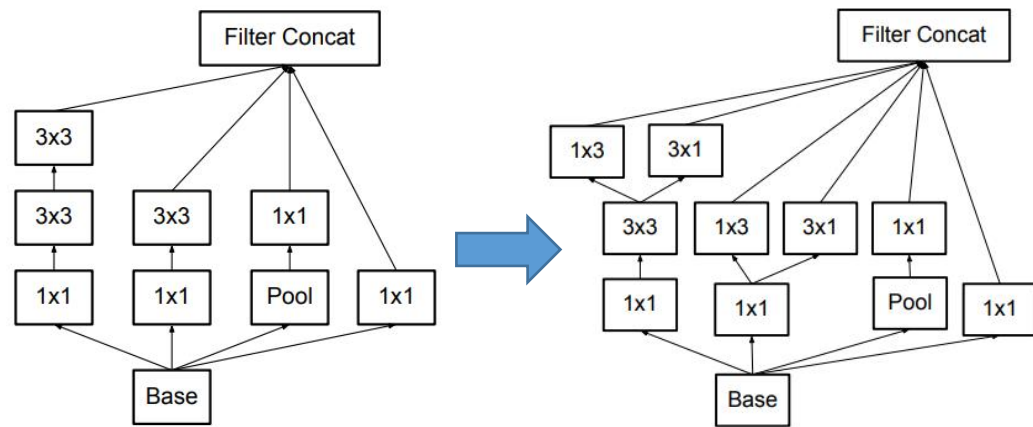


Figure 3. Replace  $3 \times 3$  convolution with asymmetric[6]

- Using  $3 \times 3$  filter, number of parameters:  $3 \times 3 = 9$ .
- Using  $1 \times 3$  and  $3 \times 1$  filters, number of parameters:  $1 \times 3 + 3 \times 1 = 6$
- The number of parameters is reduced by 33%

Theoretically, any  $n \times n$  convolution can be replaced by a  $1 \times n$  convolution followed by an  $n \times 1$  convolution. This saves the computational cost which increases significantly as  $n$  increases (Figure 4). In practice, that employing this factorization does not work well on early layers, but it gives very good results on medium grid-sizes

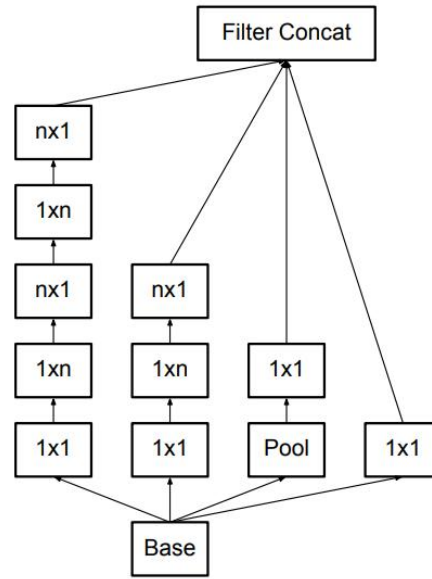


Figure 4. Inception modules after the factorization of the  $n \times n$  convolutions.[6]

#### 2.1.4. Auxiliary Classifier

Auxiliary Classifiers were already suggested in GoogLeNet / Inception-v1[4]. There are some modifications in Inception-v3. Only 1 auxiliary classifier is used on the top of the last  $17 \times 17$  layer, instead of using 2 auxiliary classifiers. (The overall architecture would be shown later.)

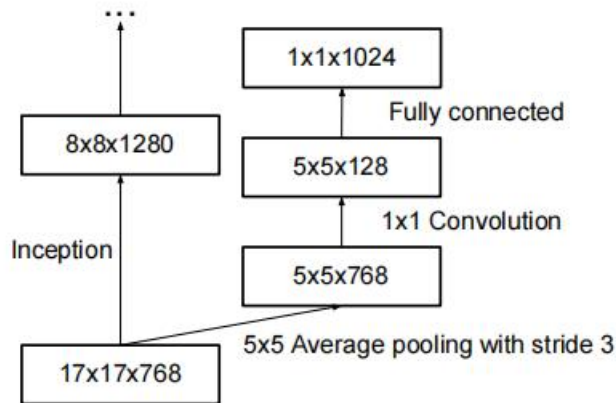


Figure 5. The classifier acts as a regularization mechanism [6]

The purpose is also different. In GoogLeNet / Inception-v1 [4], auxiliary classifiers are used for having deeper network. In Inception-v3, auxiliary classifier is used as regularizer. So, actually, in deep learning, the modules are still quite intuitive.

### 2.1.5. Grid Size Reduction

With the efficient grid size reduction, 320 feature maps are done by conv with stride 2. 320 feature maps are obtained by max pooling. And these 2 sets of feature maps are concatenated as 640 feature maps and go to the next level of inception module. Less expensive and still efficient network is achieved by this efficient grid size reduction.

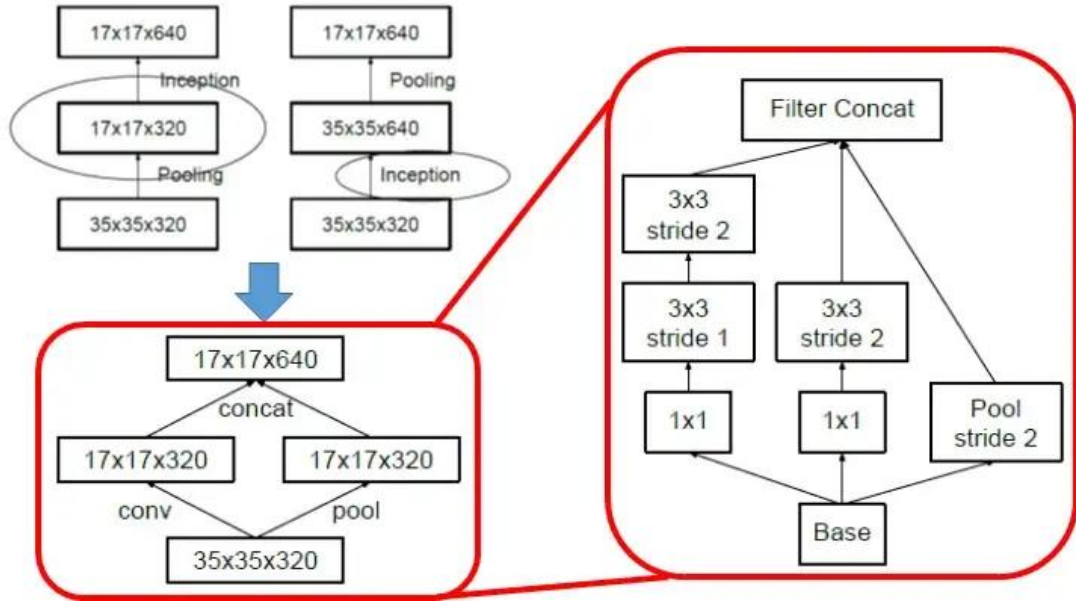


Figure 6. Grid size reduction diagram [6]

## 2.2. Transformer model

Transformer uses an Encoder and Decoder architecture, allowing for many parallel computations. So reduce training time.

### 2.2.1. Self-attention mechanism

Self-attention is the most important component of the transformer. While the attention mechanism will calculate based on the decoder's state at the current time-step and all the hidden states of the encoder. And self-attention can be understood as attention in a sentence, when each element in the sentence will interact with each other. Each token will "observe" the remaining tokens, gather the context of the sentence and update the representation vector.

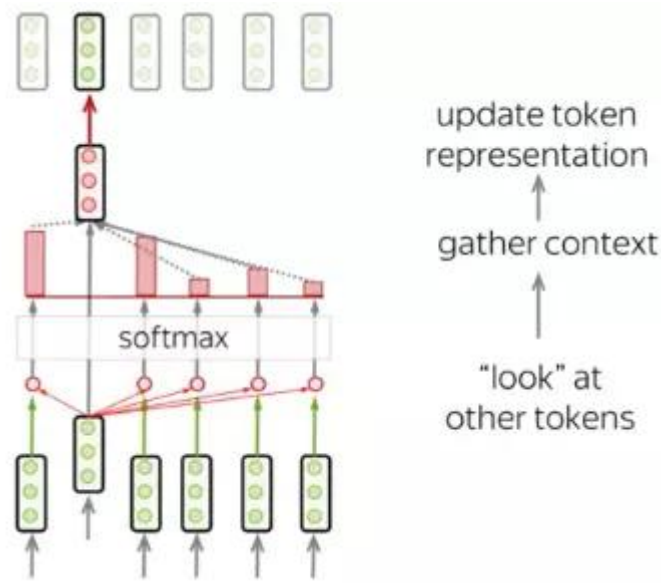


Figure 7. Brief self-attention model

Source: <https://viblo.asia/p/tim-hieu-ve-kien-truc-transformer-Az45byM6lxY>

To build a self-attention mechanism, it is necessary to pay attention to the operation of 3 vectors representing each word, respectively:

- Query: ask for information
- Key: reply that it has some information
- Value: return that information

Query is used when a token "observes" the remaining tokens, it will look around to understand the context and its relationship with the remaining tokens. The key will respond to the Query's request and is used to calculate the attention weight. Finally, Value is used with the attention weight just now to calculate the attention vector.

### 2.2.2. Masked Self-Attention Mechanism

This is the mechanism used for the Decoder in the transformer, specifically it performs the task of allowing the target token at the current time-step to be only allowed to use the tokens in the previous time-step. In terms of operation it is the same as introduced above, except that it does not take into account the attention of future tokens.

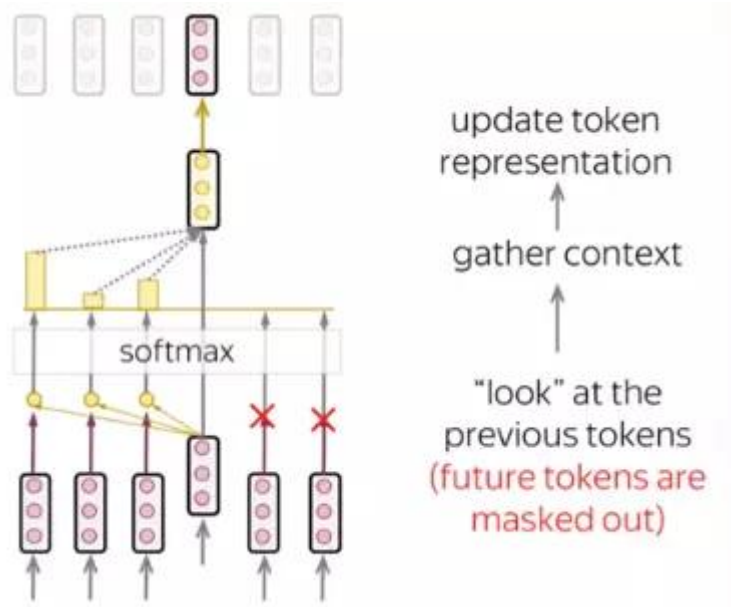


Figure 8. Masked Self-Attention mechanism

Source: <https://viblo.asia/p/tim-hieu-ve-kien-truc-transformer-Az45byM6lxY>

### 2.2.3. Multi-head attention

To understand the role of a word in a sentence, we need to understand the relationship between that word and the rest of the sentence. This is very important in the process of processing input sentences and also in the process of creating sentences. Therefore, the model needs to focus on many different things, namely, instead of having only one self-attention mechanism as introduced, also known as 1 "head", the model will have many "heads". Each head will focus on the aspect of the relationship between the word and the rest. That is multi-head attention.

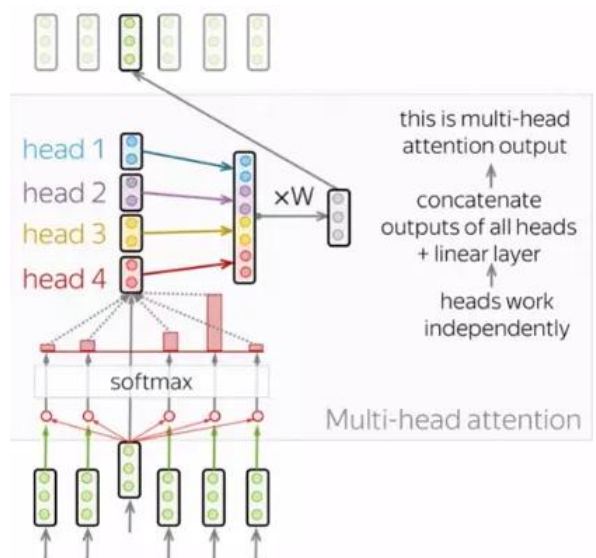


Figure 9. Multi-head attention mechanism

Source: <https://viblo.asia/p/tim-hieu-ve-kien-truc-transformer-Az45byM6lxY>



When implementing, we need to rely on query, key and value to calculate for each head. Then, concat the resulting matrices to obtain a matrix of multi-head attention. To get an output of the same size as the input, it is necessary to multiply by the matrix  $W$ .

#### 2.2.4. Transformer Architecture

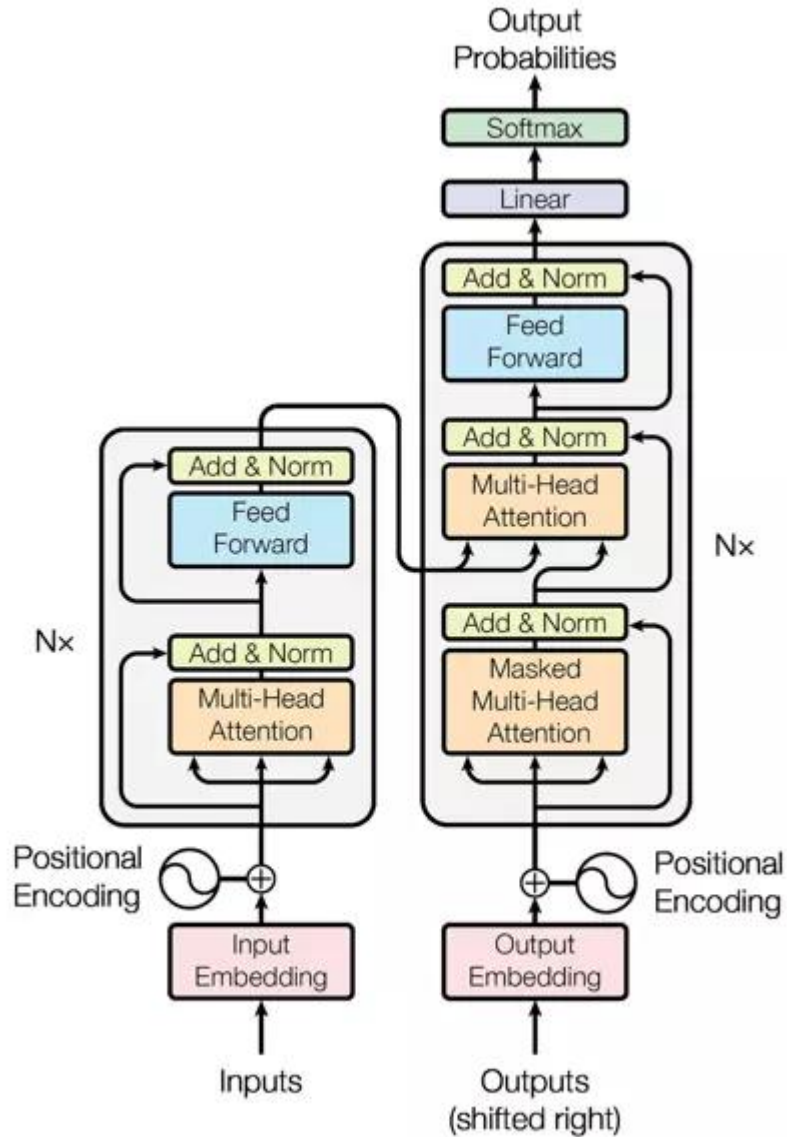


Figure 10. Transformer model architecture [8]

Figure 10 details the Transformer model architecture. On the left is the Encoder, usually with  $N \times$  layers stacked on top of each other. Each layer will have Multi-head attention and a Feed-forward block. Also Residual is the same as in Resnet. On the right is the Decoder, similarly there will be  $N \times$  layers overlapping. The architecture is the same as the Encoder but will have the Masked multi-head attention block in the first place.

Components in the model include:

- **Positional encoding:** Because the transformer has no feedback or convolutional networks, it will not know the order of the input tokens. Word embedding helps to represent the semantics of a word, but the same word in different positions will have different meanings. Transformer uses Positional encoding to encode the position of words in a sentence. The vector representing the position of the word in the sentence will be added to the tokens after the word vector embedding step.
- **Normalization class:** Figure 10 has the class "Add & Norm", Norm represents the Normalization class. This layer simply renormalizes the output of multi-head attention, effectively improving convergence.
- **Residual connection:** The residual connection essentially adds the input of a block to its output. With this connection, the network can overlap many layers. Residual connection also allows information to pass through sub-layers directly.
- **Feed-Forward block:** This is the basic block, after performing computation in attention block at each layer, the next block is Feed-Forward Networks (FFN). It can be understood that the attention mechanism helps to collect information from the input tokens, then the FFN is the block that processes that information.

### Transformers Encoder

Transformer uses Self attention mechanism for encryption. To speed up the execution as well as show the correlation between words, the words in the sentence are embedded into the model by parallel processing. Each word is embedded in a vector of size  $d=256$ . Use the Positional Encoding mechanism to encode the position of words in a sentence.

Positional Encoding uses the sin and cos formulas to calculate the positional encoding of the word. Now the input of Self attention is the word vector along with the encoded position.

$$PE_{(pos,2i)} = \sin\left(\frac{POS}{1000^{\frac{2i}{d_{model}}}}\right) \quad (1)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{POS}{1000^{\frac{2i}{d_{model}}}}\right) \quad (2)$$

Self attention calculates the relevance index of words in a sentence. From there, their relationship was born. First, create three vectors for each word including the

query vector (queries - Q), the key vector (keys - K), and the value vector (values - V). These vectors are generated by multiplying the embedding by 3 randomly initialized matrices but they are trained during execution. Their size is 64, while the input/output vectors embedded in the encoder are 256. The score is calculated by taking the dot product of the query vector with the corresponding word key vector. scoring. Divide the score by 8 then pass the result through the Softmax operation. Softmax normalizes the scores so that they are all positive and less than 1.

The Multi-head attention mechanism is used to improve the computation of the Self attention mechanism. Transformer uses 8 attention heads. Figure 11 shows the calculation process for the encryption model.

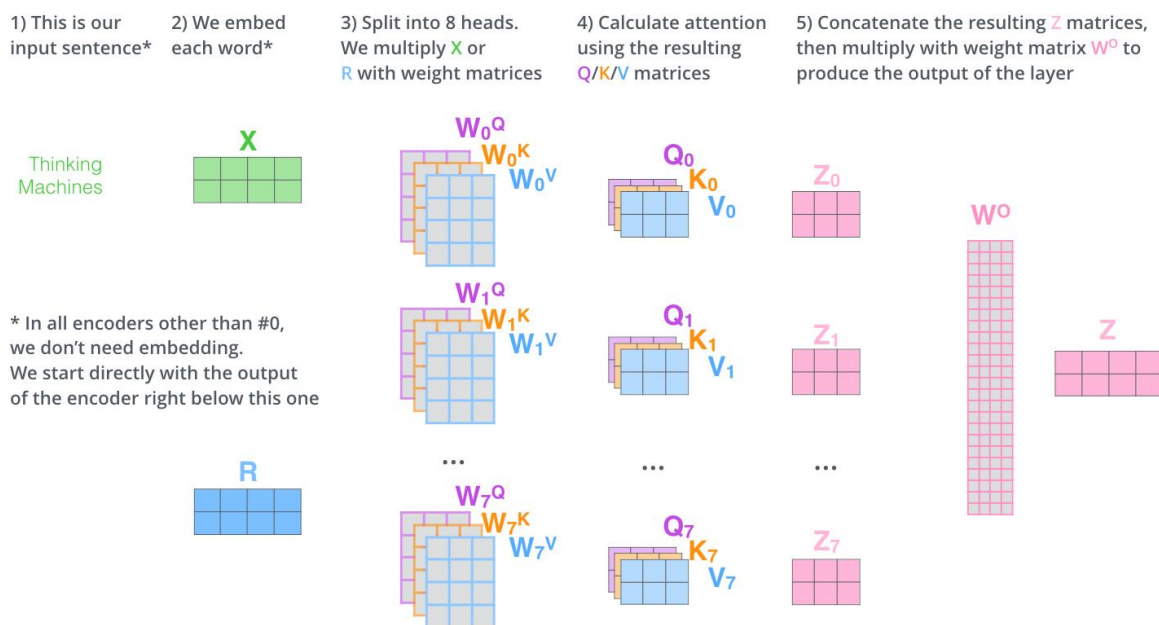


Figure 11. General diagram of the computational process of the Encoder model

Source: <https://buiminhptit.github.io/2020/03/10/gi%E1%BA%A3i-th%C3%ADch-m%C3%B4-h%C3%ACnh-transformer.html>

## Transformers decoder

The encoder starts by processing the input string. The output of the encoder is then transformed into a set of attention vectors K and V. These vectors will be used by each decoder in the "encoder-decoder attention" layer to help the decoder focus on those appropriate position in the input string. Performs the same computation as the encoder. In the same way as an encoder, embed and add a position encoding to the inputs of that decoder to indicate the position of each word.

In the decoder, the attention layer is only allowed to participate in earlier positions in the output sequence. This is done by masking the positions taken next before the softmax step in Self attention. The Encoder-Decoder attention class works like

Multi-head attention, except that it generates a query matrix from the underlying layer, which takes the key and value matrix from the output of the encoder stack. The decoder's Linear layer is a fully connected neural network that projects the vector generated by the decoder stack into a larger vector called the logits vector. Then pass a Softmax layer to divide their probabilities.

### 2.3. Evaluation methods

BLEU [9] is a widely used translation quality assessment method in machine translation. The main idea of the method is to compare the results of automatic machine translation with a standard translation used as a reference. The comparison is made through statistical matching of words in two translations taking into account their order in the sentence (word-by-grams method)[10]. This method is based on the correlation coefficient between machine translation and accurate human translation to evaluate the quality of a translation system.

The evaluation is done on the statistical results of the match of n-grams (character strings consisting of n words or characters) from the data store of translation results and the repository of high-quality reference translations[9]. IBM's algorithm evaluates the quality of the translation system by the matching of n-grams, and it is also based on comparing the lengths of the translations.

IBM's evaluation scoring formula is as follows [11]:

$$\text{Score} = \exp \left\{ \sum_{i=1}^N w_i \log(p_i) - \max \left( \frac{L_{\text{ref}}}{L_{\text{tra}}} - 1, 0 \right) \right\} \quad (3)$$

$$p_i = \frac{\sum_j NR_j}{\sum_j NT_j} \quad (4)$$

The parameters in the formula are as follows::

- $NR_j$ : is the number of n-grams in segment j of the translation for reference.
- $NT_j$ : is the number of n-grams in segment j of the machine translation.
- $w_i = N^{-1}$
- $L_{\text{ref}}$ : is the number of words in the reference translation, the length of which is usually close to the length of the machine translation.
- $L_{\text{tra}}$ : is the number of words in the machine translation.

Score value evaluates the degree of correspondence between two translations and it performs on each segment, where segment is understood as the minimum unit in translations, usually each segment is a sentence or a paragraph. The matching

statistics of n-grams is based on the set of n-grams on segments. First, is it calculated on each shard, then recalculate this value on all shards.

## CHAPTER 3. METHODS OF IMPLEMENTATION

### 3.1. Description of the problem

Building descriptions for images is a problem involving computer vision and natural language processing to recognize the context of images, then describe them in natural language such as English. The model performs feature extraction of the encoded image to create feature vectors for the image. And the model also encodes the descriptions into word vectors. Finally, merge image feature vectors and word vectors, and also decode them to get a complete description sentence.

### 3.2. Build descriptions for images

Quá trình thực hiện và đánh giá mô hình xây dựng mô tả cho ảnh được thực thi trên Flickr8k data theo các bước sau:

- Step 1: Data analyze
- Step 2: Data preprocessing
- Step 3: Word Embedding
- Step 4: Build the model
- Step 5: Evaluate the model

✧ Step 1 - Data analyze: The model is developed on Flickr8k data file with 8,091 images with each image including 5 different description sentences. The image name is unique and is considered a distinguishing identifier. All description sentences are saved in txt file with information including image name, index from 0 to 4 corresponding to 5 description sentences for that image and finally description sentence for image.

✧ Step 2 - Data preprocessing: Perform data cleaning to remove redundant characters to reduce the number of vocabularies while reducing storage space to improve the execution speed of the model.

✧ Step 3 - Word Embedding: is a class of techniques in which individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to a vector and the vector values are learned in a similar way to a neural network. In this thesis, use the Tokenizer of keras to separate words and phrases to form a dictionary of words with corresponding indexes.

✧ Step 4 - Build the model

To build a model to create descriptive sentences for images, it is necessary to complete two processes: encoding images into image feature vectors and encoding description sentences into word vectors. These two jobs are performed independently of each other, after the image and description encoding is completed, the merge process is started for training.

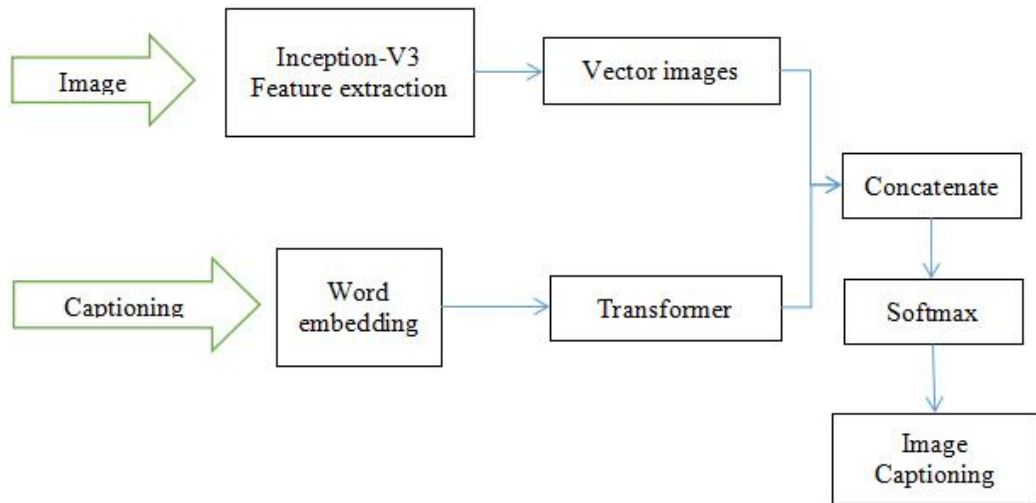


Figure 12. The process of creating descriptions for image

The model for constructing descriptive sentences for images is built based on 3 component models including: feature extraction model for image, encoder model and decoding model.

### Feature extraction model of the image

Use pre-trained Inception-V3 architecture to extract features on images. There is no need to classify the image so we just need to extract an image vector for the input image. Therefore, eliminate the softmax layer from the model. All images are resized to 299x299, adding a vector dimension to the image now the model input image size is 299x299x3. After feature extraction, the output of the model is a vector of size 8x8x2048. The architecture of Inception-V3 model to extract image features is presented in the following figure:

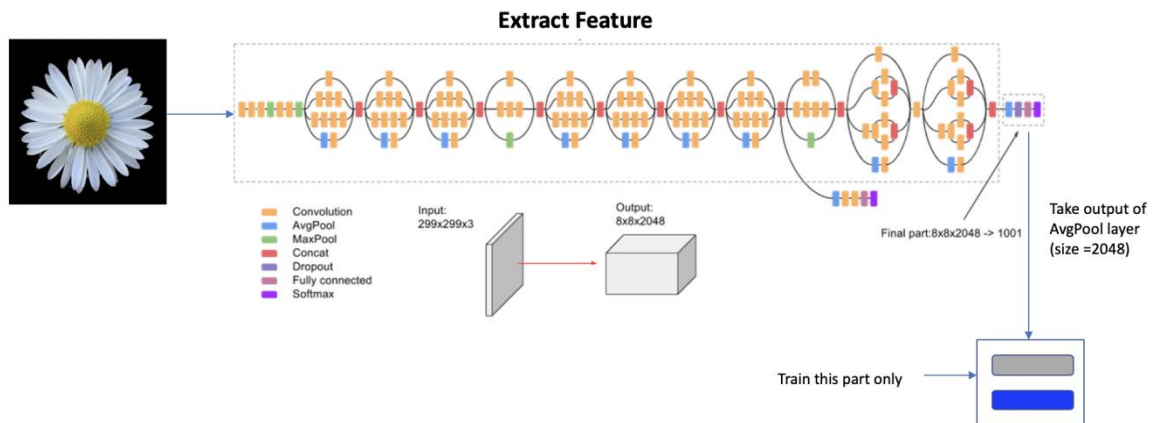


Figure 13. Inception-V3 model extracting image features

### **Transformer encoder model**

Each word entered into the Transformer model will be embedded in a vector of size  $d=256$ . Use the Positional Encoding mechanism to encode the position of the word in the sentence. Then, the Self Attention mechanism will calculate the related indexes to generate the relationship between the words through the query vectors (queries -  $q$ ), key vectors (keys- $k$ ) and value vectors (values -  $v$ ). The Multi-head attention mechanism will be used to improve the calculation of the self attention mechanism.

### **Transformer decoder model**

The output of the encoder model is transformed into a set of vectors  $k$  and  $v$ . These vectors will be used by the decoder in the “encoder-decoder attention” layer. Performs the same computation as the encoder. The linear layer of the decoding model implements the vector projection generated by the stack of decoding layers into the vector logist. And then, The softmax layer turns those scores into probabilities that help determine the output for the model.

✧ Step 5 - Evaluate the model

Conduct evaluation of the descriptive sentence construction model for the image by BLEU score.



## CHAPTER 4. EXPERIMENT

### 4.1. Dataset

Using the Flickr8k dataset taken from Kaggle to train and evaluate the model to build description sentences for images. The dataset contains 8,091 images and each image corresponds to 5 descriptive sentences in English. The image name is considered as the ID used to distinguish each other and is used to link between train and test files. Example of a data set is shown in the figure:



the white and brown dog is running over the surface of the snow .  
a white and brown dog is running through a snow covered field .  
a dog running through snow .  
a dog is running in the snow  
a brown and white dog is running through the snow .



man on skis looking at artwork for sale in the snow  
a skier looks at framed pictures in the snow next to trees .  
a person wearing skis looking at framed pictures set up in the snow .  
a man skis past another man displaying paintings in the snow .  
a man in a hat is displaying pictures next to a skier in a blue hat .



several climbers in a row are climbing the rock while the man in red watches and holds the line .  
seven climbers are ascending a rock face whilst another man stands holding the rope .  
a group of people climbing a rock while one man belays  
a group of people are rock climbing on a rock climbing wall .  
a collage of one person climbing a cliff .



large brown dog running away from the sprinkler in the grass .  
a dog is playing with a hose .  
a brown dog running on a lawn near a garden hose  
a brown dog plays with the hose .  
a brown dog chases the water from a sprinkler on a lawn .

*Figure 14. Flickr8k dataset*

### 4.2. Data preprocessing

Process and format descriptive sentences by removing redundant punctuation and words to speed up execution.

### 4.3. Training

Mô hình được huấn luyện trên google Colaboratory. Tập dữ liệu Flickr8k sử dụng 6.000 hình ảnh để huấn luyện tương ứng với 30.000 câu mô tả. Sử dụng mô hình InceptionV3 để rút trích đặc trưng hình ảnh và lưu dưới dạng mảng từ điển. Sau đó, sử dụng các đặc trưng vừa trích xuất được và các câu mô tả đã tiền xử lý để huấn luyện mô hình. Mô hình sử dụng thư viện chính là Tensorflow và Keras.

#### 4.4. Accuracy rating


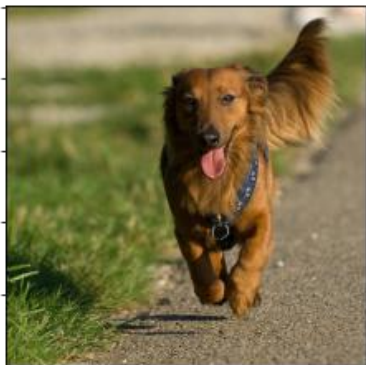
Evaluate the model for building image description sentences by calculating cumulative points for BLEU with the following weights: BLEU-1(1, 0, 0, 0), BLEU-2(0.5, 0.5, 0, 0), BLEU-3(0.3, 0.3, 0.3, 0) and BLEU-4(0.25, 0.25, 0.25, 0.25). The evaluated BLEU index is calculated by evaluating the average BLEU index on 100 images of the test set, comparing the description sentences predicted from the newly built model and 5 descriptive sentences in the original data set of that image. The experimental results are shown in Table 1.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Inception-V3 + Transformer	24.94	12.43	5.40	5.27

Table 1. Table of BLEU evaluation results of the descriptive sentence build model

#### 4.5. Experimental results

Some experimental results in the Flickr8k set are summarized in Table 2. The results include images, real caption, predicted caption and BLEU index corresponding to each result.

Test 1	
	<p>BLEU-1: 25.0  BLEU-2: 7.46  BLEU-3: 1.69  BLEU-4: 1.29  Real Caption: man is jumping into the surf  Predicted Caption: man in dark wetsuit is caught by wave</p>
Test 2	
	<p>BLEU-1 score: 28.57  BLEU-2 score: 7.97  BLEU-3 score: 1.76  BLEU-4 score: 1.33  Real Caption: &lt;unk&gt; brown dog running  Predicted Caption: the dog is running down the road</p>

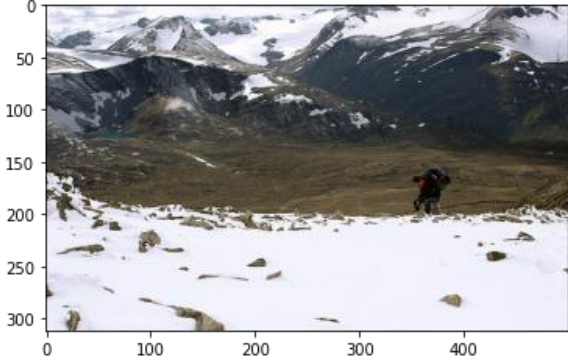


Test 3	
	<p>BLEU-1 score: 14.29  BLEU-2 score: 5.64  BLEU-3 score: 1.43  BLEU-4 score: 1.12  Real Caption: man descends snowy mountain  Predicted Caption: the person is in the snowy mountains</p>
Test 4	
	<p>BLEU-1 score: 28.57  BLEU-2 score: 21.82  BLEU-3 score: 2.03  BLEU-4 score: 6.97  Real Caption: miami basketball player shooting  Predicted Caption: basketball player preparing to shoot the ball</p>
Test 5	
	<p>BLEU-1 score: 39.77  BLEU-2 score: 21.09  BLEU-3 score: 1.90  BLEU-4 score: 6.48  Real Caption: man in aerodynamic gear riding professional mountain bike through forest  Predicted Caption: man in blue helmet and safety suit riding bike</p>

Table 2. Some test results of the model

## **CHAPTER 5. CONCLUSION**

### **5.1. Achieved results**

The basic model has built description sentences with Inception-v3 and Transformer models based on the training data set. However, the BLEU-1 index was only 24.94, lower than that of the study with the same method of Shah et al [1], which achieved the BLEU-1 score of 54.30.

### **5.2. Development direction**

The current model has some limitations in accuracy and has not been able to create a complete description. In the future, research will be conducted to improve the accuracy of the model. Or, it is possible to apply new methods for feature extraction of models such as YOLO, Inception-v4 to enhance the quality of input data to create better recognition capabilities for recognizing objects in images. Help predictive descriptive sentences stick to the image content. Moreover, is to create a website "Image captioning" with the model of creating the most perfect description.

## REFERENCES

- [1] Shah, Faisal Muhammad, et al. "Bornon: Bengali Image Captioning with Transformer-based Deep learning approach." *arXiv preprint arXiv:2109.05218* (2021).
- [2] Xinlei Chen, C. Lawrence Zitnick; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2422-2431
- [3] Khang Nhut, L. A. M., et al. "Facial Expression Recognition and Image Description Generation in Vietnamese." *Fuzzy Systems and Data Mining VII: Proceedings of FSDM 2021* 340 (2021): 63.
- [4] Xiong, Yuxuan, Bo Du, and Pingkun Yan. "Reinforced transformer for medical image captioning." *International Workshop on Machine Learning in Medical Imaging*. Springer, Cham, 2019.
- [5] Jiang, Wenhao, et al. "Recurrent fusion network for image captioning." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
- [6] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
- [7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- [8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [9] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). 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).
- [10] Hovy, E. H. (1999). Toward finely differentiated evaluation metrics for machine translation. In *Proceedings of the EAGLES Workshop on Standards and Evaluation* Pisa, Italy, 1999.
- [11] Hùng, V. T. (2007). Phương pháp và công cụ đánh giá tự động các hệ thống dịch tự động trên mạng. *Tạp chí Khoa học và Công nghệ Đại học Đà Nẵng*, 37-42.

## **APPENDIX**

Instructions for installation and use follow the following link:  
<https://github.com/ntmkhanh>