VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

REPORT

COMPUTER VISION

Instructor: **PhD. PHAM VAN HUY**

Student: **BUI HAI DUONG - 521H0220**

PHAN MINH HOANG – 521H0501

LA NGUYEN QUOC THINH – 521H0513

Class: **21H50302**

HO CHI MINH CITY, 2025

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

REPORT

COMPUTER VISION

Instructor: **PhD. PHAM VAN HUY**

Student: **BUI HAI DUONG - 521H0220**

PHAN MINH HOANG – 521H0501

LA NGUYEN QUOC THINH – 521H0513

Class: **21H50302**

HO CHI MINH CITY, 2025

ACKNOWLEDGEMENT

To complete this essay, besides our own efforts, we have received a lot of help in terms of knowledge, experience, and skills from the school and teachers. First and foremost, we would like to express my special gratitude to PhD Pham Van Huy - the lecturer of Introduction to Computer Vision course who has taught us valuable knowledge of the subject. That knowledge is the foundation for us to continue learning and effectively apply it to this essay. Additionally, We would like to thank the teacher for allowing us to complete this essay, which has helped us to further develop my understanding of the subject. Thank you for your guidance and support in helping us to complete this essay to the best of my ability. Moreover, we would also like to express our gratitude to the school and the teachers who have compiled the Introduction to Computer Vision materials, providing me with useful resources for research and essay writing. Thank you sincerely!

COMPLETION OF THESIS
AT TON DUC THANG UNIVERSITY

We here by certify that this thesis is my/our own work and was conducted under the guidance of PhD. Pham Van Huy. The research and results presented in this thesis are truthful and have not been published previously in any form. The data presented in tables and figures used for analysis, comments, and evaluations were collected by the author from various sources and are clearly cited in the reference section.

Moreover, this thesis includes some comments, evaluations, and data from other authors and organizations, which are properly cited and referenced.

If any misconduct is detected, I fully take responsibility for the content of my thesis. Ton Duc Thang University is not liable for any copyright infringement that may occur during the thesis completion process.

Ho Chi Minh City, January 5, 2025
Author
(signature and full name)

ACKNOWLEDGEMENT AND EVALUATION SECTION BY INSTRUCTOR

Instructor's Acknowledgement Section

Ho Chi Minh City, 2025

(signature and full name)

Instructor's Evaluation Section

Ho Chi Minh City, 2025

(signature and full name)

SUMMARY

The task of this documents is about how to deal with the Image captioning task, introducing two main approaching ways are Merging and Injection architecture to solve the problems. This document separated into 5 main chapters:

Chapter 1: Introduction – This chapter covers main general ideas to deal with the image captioning task.

Chapter 2: Preprocessing – Describe about how raw images and captions is preprocessed

Chapter 3: Model architecture – Showing the main architecture used in the implementation and the mathematical behind.

Chapter 4: Environmental setup – Showing the base setup before training and evaluating the model.

Chapter 5: Result – Showing some result after the training process.

Table of Contents

TABLE OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long-short Term Memory
NLP	Natural Language Processing

CHAPTER 1: INTRODUCTION

1.1 What is image captioning

Image captioning is a crucial task in the field of computer vision and natural language processing, where the goal is to generate descriptive and meaningful captions for a given image. This process involves analyzing the visual content of the image and translating it into coherent text that captures the essence of the scene. By bridging the gap between visual data and language, image captioning enables a wide range of applications, such as improving accessibility for visually impaired individuals, enhancing search and retrieval systems, and supporting automated content generation. The underlying methodology typically leverages deep learning techniques, combining convolutional neural network (CNN) for image feature extraction with recurrent neural network (RNN) or transformers for language generation. This interdisciplinary approach highlights the synergy between visual understanding and linguistic representation, showcasing advancements in artificial intelligence.

1.2 Approaching ways

1.2.1 Injection architecture

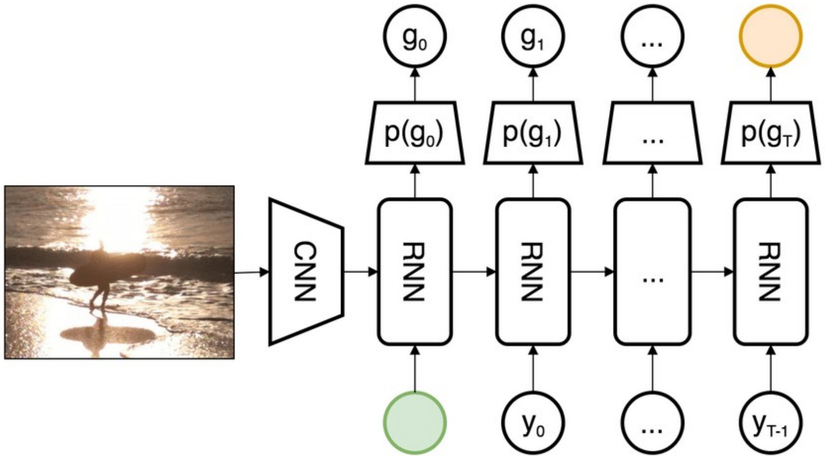


Figure 1. Injection architecture

Injection architecture refers to a design framework in which the input to the model is a combination of images and their associated captions, enabling a synergistic integration of visual and textual data. In this architecture, the image plays a central role, maintaining its relevance throughout each step of the RNN processing. By blending visual features extracted from the image with linguistic elements from the caption, this approach ensures that the model can effectively leverage both modalities to generate accurate and context-aware outputs.

Typically, the image features are extracted using a CNN and then injected into the RNN at various stages of processing, either at the input layer, intermediate layers, or recurrently at each time step. This continuous involvement of the image ensures that the visual context is preserved and directly influences the generation of each word in the caption. Such an architecture is particularly advantageous for tasks requiring a deep understanding of the interplay between visual content and language, enhancing the model's ability to produce coherent and contextually rich captions.

1.2.2 Merging architecture

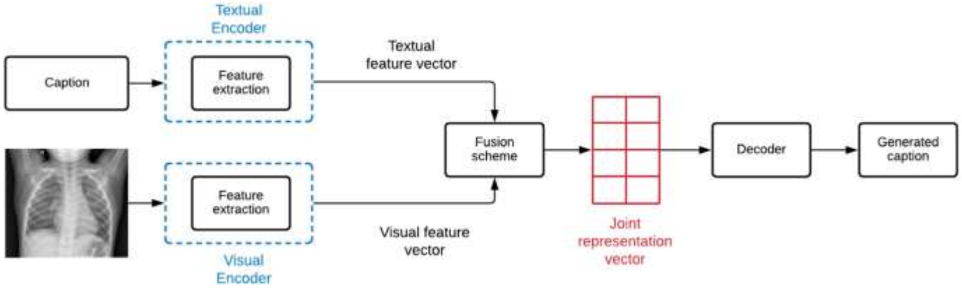


Figure 2. Merging architecture

In contrast to injection architecture, merging architecture processes images and captions independently before combining their outputs to generate the final caption. This approach treats visual and textual data as separate streams, allowing each modality to be processed using specialized models or techniques optimized for their respective characteristics. For instance, image features are typically extracted using a CNN, while captions are processed using natural language models, such as RNN or transformers. Once the independent processing is complete, the outputs from both streams are merged, often through concatenation, attention mechanisms, or other fusion techniques. This merging step integrates the information from both modalities, enabling the model to generate a caption that reflects the combined understanding of the visual and textual inputs. While this architecture ensures modularity and flexibility in handling different types of data, it may lack the continuous interaction between modalities that characterizes injection architectures, potentially making it less effective for tasks requiring deep interdependence between images and captions. Nonetheless, merging architectures remain a robust choice for applications where independent feature extraction and subsequent integration are sufficient to achieve accurate results.

CHAPTER 2: PREPROCESSING

The preprocessing step is a critical step in the pipeline of building an Image Captioning model. This step makes sure that the raw data (image and captions) will be transformed into a structured format, which is suitable for the model to learn.

3.1 Data description

The dataset includes:

- Images - These images are stored in a directory.
- Captions - Each image has one or more related textual descriptions, stored in a CSV file with 2 columns: image (image filename) and caption (sentences describe the image).

3.2 Vocabulary Construction

To process caption effectively, we need to convert text into numerical representations. This requires a vocabulary that maps words to unique numbers.

- Vocabulary Class
    - Special Tokens - Includes these special tokens:
        - <PAD>: Padding token for sequences.
        - <SOS>: Star of sequences.
        - <EOS>: End of sequences.
        - <UNK>: Unknown words.
    - Tokenization - Using spacy to split captions into each word.
    - Frequency Threshold - Words must appear at least a predefined number of times to be added to the vocabulary.
    - Mapping - Includes mapping words to their index and mapping indexes back to words.
- Vocabulary Building

    - Captions will be tokenized, and the frequency counter will determine which words meet the threshold and add it to the vocabulary. The less frequent words are replaced by <UNK>.

3.3 Image Preprocessing

The following transformations are applied:

- Resizing: All images are resized to 226x226 pixels.
- Random Cropping: Crops a 224x224 section from the resized image.
- Normalization: Pixel values are normalized using mean and standard deviation values taken from the ImageNet dataset:
    - Mean: (0.485, 0.456, 0.406)
    - Std: (0.229, 0.224, 0.225)

CHAPTER 3: MODELS ARCHITECTURE

In this chapter, we present the architecture of the sequence-to-sequence (Seq2Seq) model employed to address the image captioning task. The Seq2Seq framework has proven effective for generating textual descriptions from visual data, leveraging distinct components for feature extraction and sequence generation. The model consists of an encoder, which processes input images, and a decoder, which generates coherent captions. Additionally, we examine the integration of the attention mechanism and its impact on performance, particularly in handling long-range dependencies.

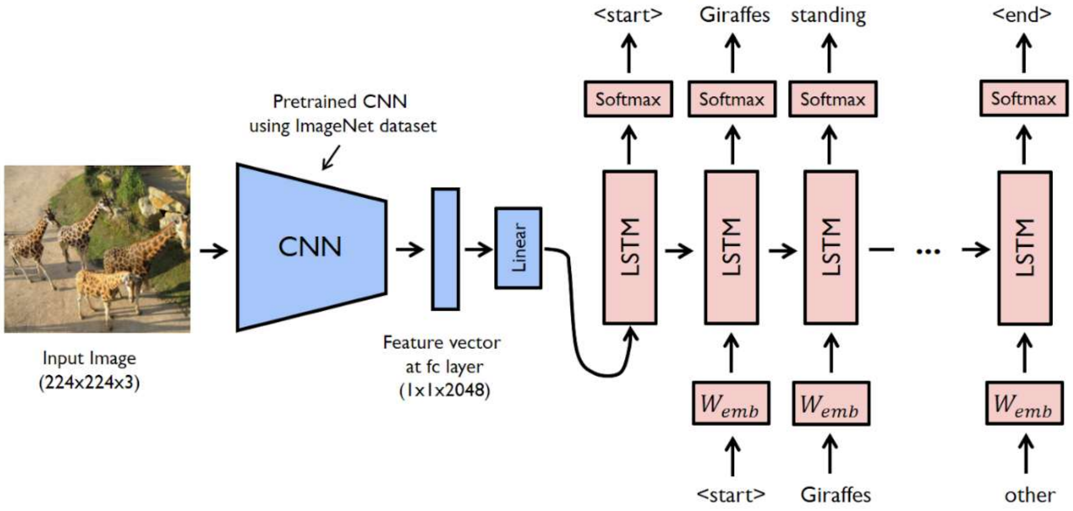


Figure 3. Injection architecture workflow

4.1 Encoder

The encoder in our Seq2Seq model is designed to extract high-level feature representations from images. For this purpose, we employ a CNN architecture, specifically ResNet. ResNet is known for its residual learning capability, which alleviates the vanishing gradient problem and allows for the construction of deeper networks. Given an input image I. The encoder outputs a feature map $F \in R^{WxHxC}$, where W, H and C are the width, height, and number of channels of the feature map, respectively. The feature extraction process can be mathematically expressed as:

$$F \; = \; ResNet(I)$$

Here, F encapsulates the spatial and semantic characteristics of the image, which are essential for generating contextually accurate captions.

4.2 Decoder

The decoder transforms the encoded image features into a sequence of words forming the caption. In this project, we utilize a LSTM network due to its proficiency in modeling sequential data and mitigating the vanishing gradient issue over long sequences.

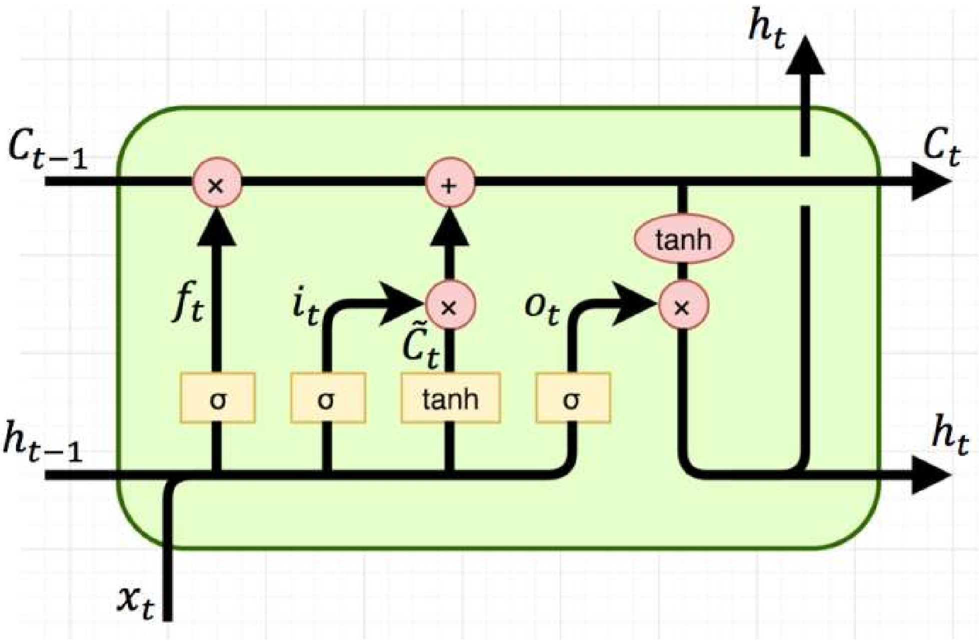


Figure 4. LSTM architecture

f_t, i_t, o_t correspond to the forget gate, input gate and output gate.

- Forget gate: $f_t = \sigma\left(U_f . x_t + W_f . h_{t-1} + b_f\right)$
- Input gate: $i_t = \sigma(U_i . x_t + W_i . h_{t-1} + b_i)$
- Output gate: $o\_t = \sigma(U_o . x_t + W_o . h_{t-1} + b_o)$

$c\_t = tanh(U_c . x_t + W_c . h_{t-1} + b_c),$ similar to compute s_t in RNN

The decoding process begins by projecting the feature map F into a fixed-dimensional h and c which serves as the initial hidden state of the LSTM:

$$h, c \ = \ f_{(proj)}(F)$$

At each timestep t, the decoder predicts the next word y_t based on the current hidden state h_t, the previous word y_{t-1}, and the context vector c_t. The process is governed by the following equations:

$$c_t = f_t . c_{t-1} + i_t . c_t$$

$$h_t = o_t . tanh(c_t)$$

$$yt = sofmax(W_o h_t + b_o)$$

Here, W_o and b_o are trainable parameters of the output layer. The decoder generates the caption iteratively until the end-of-sequence token is produced.

4.3 Attention

The attention mechanism significantly enhances the Seq2Seq model by dynamically focusing on specific regions of the image when generating each word in the caption. Unlike traditional approaches that rely on a single global context vector, attention computes a context vector c_t for each time step t by weighting the contributions of different parts of the image feature map F. The attention weights $t_{i,j}$ are computed as:

$$t_{i,j} = sofmax(e_i^t{}_{,j}) = \frac{exp\left(e_i^t{}_{,j}\right)}{\sum exp\left(e^{t_i'}{}_{,}{}^{j'}\right)}$$

where $e_{i,j}^t = f(h_{t-1}, F_{i,j})$ is the alignment score between the decoder's previous hidden state h_{t-1} and the feature vector $F_{i,j}$ at spatial location. The context vector c_t is then derived as:

$$c_t = \sum \alpha_i^t{}_{,j} \, F_{i,j}$$

By enabling the decoder to focus on relevant portions of the image, the attention mechanism improves the model's capacity to generate detailed and contextually accurate captions. Empirical evidence from natural language processing (NLP) tasks, such as machine translation, supports the efficacy of attention in addressing long-range dependencies, and similar benefits are observed in image captioning.

CHAPTER 4: ENVIRONMENTAL SETUP

5.1 Dataset

Dataset name: Flickr8k

Detail about this dataset:

A new benchmark collection for sentence-based image description and search, consisting of 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events, etc... The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations.

5.2 Code implementation

5.2.1 Model

Encoder model: Resnet-50

Decoder model: LSTM

Decoder model: LSTM-Attention

5.2.2 Configuration

Word embedding dimensions: 300

Attention dimensions: 256

Decoder dimensions: 512

Dropout: 0.5

Encoder learning rate: 1e-4

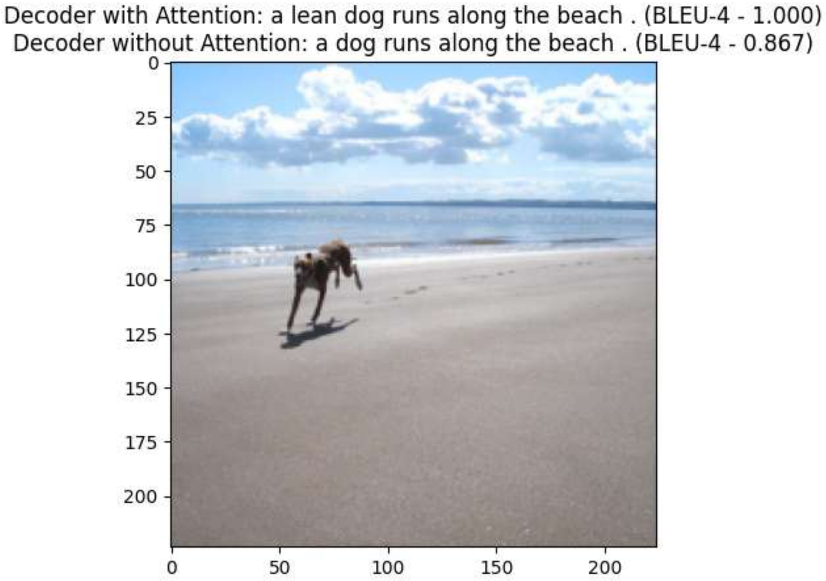
Decoder learning rate: 3e-4

Numbers of Epoch: 40

Optimizer: Adam

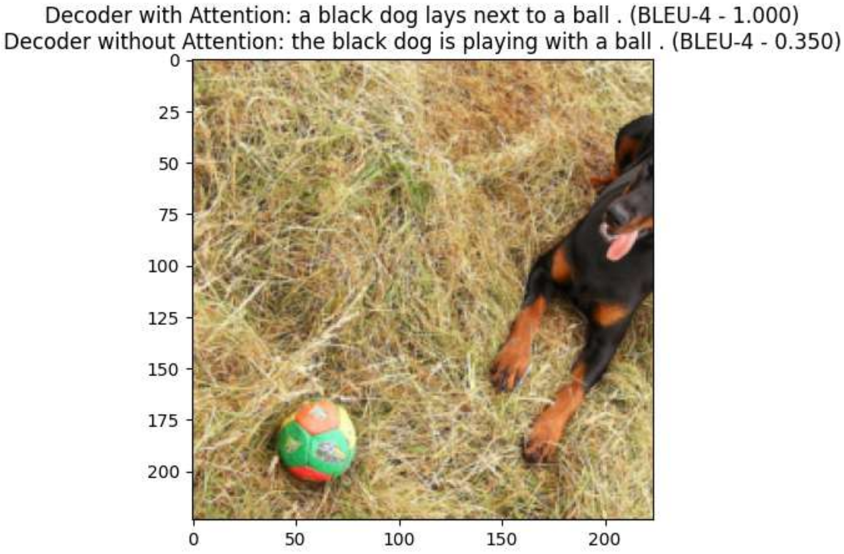
Loss function: Cross Entropy Loss

CHAPTER 5: RESULT



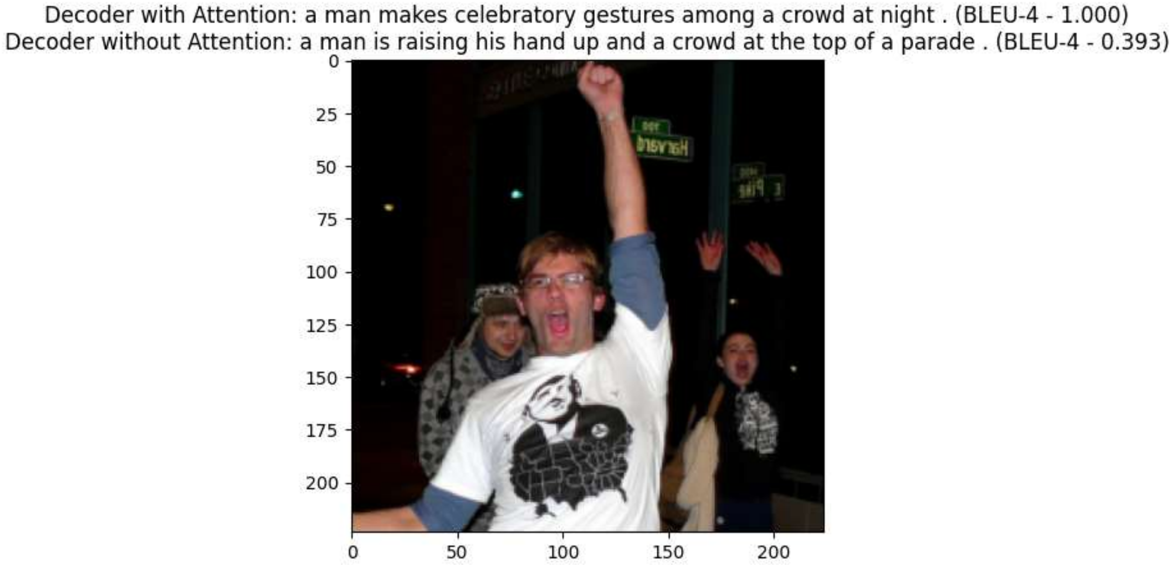
Decoder with Attention: a lean dog runs along the beach . (BLEU-4 - 1.000)
Decoder without Attention: a dog runs along the beach . (BLEU-4 - 0.867)

Figure 5. Output 01



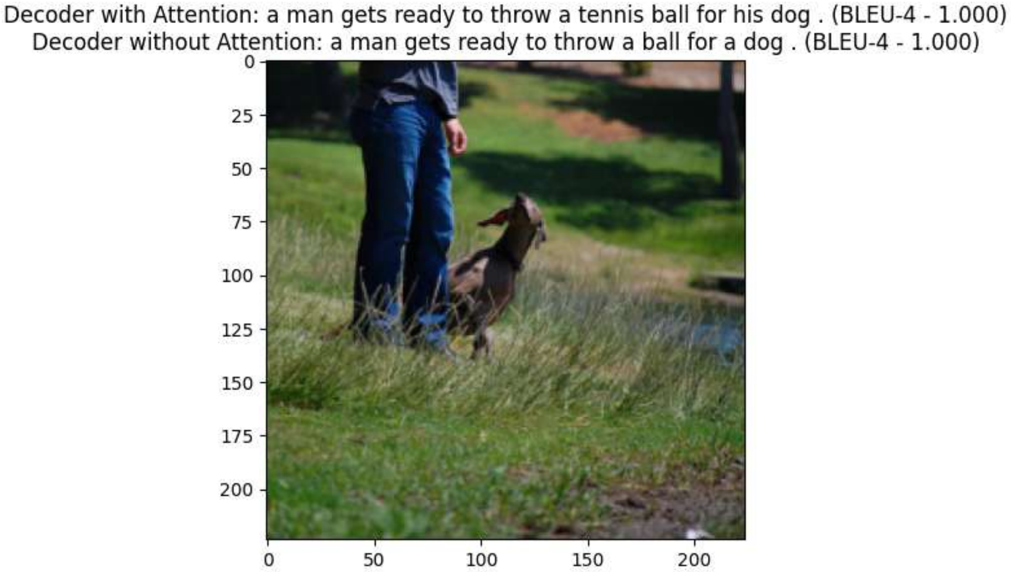
Decoder with Attention: a black dog lays next to a ball . (BLEU-4 - 1.000)
Decoder without Attention: the black dog is playing with a ball . (BLEU-4 - 0.350)

Figure 6. Output 02



Decoder with Attention: a man makes celebratory gestures among a crowd at night . (BLEU-4 - 1.000)
Decoder without Attention: a man is raising his hand up and a crowd at the top of a parade . (BLEU-4 - 0.393)

Figure 7. Output 03



Decoder with Attention: a man gets ready to throw a tennis ball for his dog . (BLEU-4 - 1.000)
Decoder without Attention: a man gets ready to throw a ball for a dog . (BLEU-4 - 1.000)

Figure 8. Output 0s4

REFERENCES

[1] Nguyen Thanh Huyen, A Guide to Image Captioning (2021), Viblo.

[2] Trung Đức, Image Captioning – Chú thích dữ liệu hình ảnh bằng công nghệ học sâu (2022), VinBigdata.

[3] G. Sairam, M. Mandha, P. Prashanth, P. Swetha et al, Image Captioning using CNN and LSTM (2023), ieeexplore.