Journal of Computer Science and Cybernetics, V.38, N.4 (2022), 327–346
DOI no. 10.15625/1813-9663/38/4/17548

EMPIRICAL STUDY OF FEATURE EXTRACTION APPROACHES FOR IMAGE CAPTIONING IN VIETNAMESE

KHANG NGUYEN

VNU-HCM University of Information Technology, Quarter 6, Linh Trung Ward,
Thu Duc City, Ho Chi Minh City, Viet Nam

Abstract. Image captioning is a challenging task that is still being addressed in the 2020s. The problem has the input as an image, and the output is the generated caption that describes the context of the input image. This study focus on the image captioning problem in Vietnamese. In detail, an empirical study of grid-based and region-based feature extraction approaches using current state-of-the-art object detection methods is investigated to explore the suitable way to represent the images in the model space. Each feature type represents images, and the image captioning task is trained using the Transformer-based model. The effectiveness of different feature types is explored on two Vietnamese datasets: UIT-ViIC and VieCap4H, the two standard benchmark datasets. The experimental results show crucial insight into the feature extraction task for image captioning in Vietnamese.

Keywords. Grid features, region features, image captioning, Viecap4h, Uit-viic, faster R-CNN, cascade R-CNN, grid R-CNN, Vinvl.

1. INTRODUCTION

Image captioning is a problem that describes the visual content in an image using natural language. The mentioned problem is a fascinating topic because it intersects two well-known research directions in artificial intelligence: Natural Language Processing and Computer Vision. In recent years, many studies have tried to increase this problem's performance. The goal is to find an optimal method to process an input image, represent its semantics, and transform the image into a sequence of words by connecting visual and linguistic objects while preserving the naturalness of the language. Simply put, image description is an image-to-sequence problem, converting an image to a sequence, where the input is a sequence of pixels of the image. The sequence of pixels will be encoded into one or more feature vectors in the visual coding step to generate input for the next step, called the language model. The language model takes the visual encoding vectors as input and then generates a sequence of words, where these words would generally be available in an existing vocabulary set.

Although research groups worldwide are still paying much attention to this problem, the research situation on the problem of generating image description sentences in Vietnam has not been strongly promoted. Therefore, this study focuses on the image captioning problem in the Vietnamese language.

Corresponding author.
E-mail address: khangnttm@uit.edu.vn (K. Nguyen).

In detail, the way of extracting visual features of input images is considered the key component that directly impacts the effectiveness of the primary captioning model. There are two approaches for extracting features: 1) extract the grid features from a CNN-based network; 2) extract the region features, which are embedding vectors of regions of interest in the input image as visual features. This study mainly focuses on the two mentioned approaches. About the grid features approach, two ResNeXt-based backbones provided in the study [1], and [2] are used as grid feature extractors. Furthermore, our experiments also use the pre-trained CLIP model as a grid feature extractor. Regarding the region features approach, 03 recent object detection models have been adopted. Including Faster R-CNN [3], Cascade R-CNN [4], and Grid R-CNN [5]; they are all categorized as two-stage object detection models. Then, the obtained visual features representing the images are then trained with the Transformer-based model to generate the hypothesis captions. In brief, our contributions can be listed as follows:

- 02 backbones in two studies [1, 6] are used to trained on the large-scale dataset for extracting grid features of input images.
- 03 state-of-the-art object detection models and 01 pre-trained model, including Faster R-CNN [3], Cascade R-CNN [4], Grid R-CNN [5], and VinVL [2] are adopted to extract region features from input images.
- To evaluate the performance of different types of region features, Transformer-based [6] model is used to train to generating captions. Two benchmark datasets for image captioning in Vietnamese are used to evaluate the effectiveness: UIT-ViIC [7] and VieCap4H [8].
- The experimental results are crucial insights for extracting visual features for the image captioning problem in the Vietnamese language.

The rest of this paper: Section 2 presents our quick survey on the image captioning problem in the world as Ill as in Vietnamese. Section 3 describes more profound about our feature extraction strategy. Section 4 describes in more detail the main Transformer-based model for training image captioning. Section 5 presents experimental results on two benchmark datasets: UIT-ViIC and VieCap4H. Section 6 summarizes the paper and presents some directions for future research.

2. RELATED WORKS

2.1. Problem definition

First, I provide the basic problem definition of image captioning for better illustration in the rest of the paper. Given **I** is the three-channel input image, the process from extracting features to decoding to the predicted output **Y** can be formulated as Equation (1), (2)

$$\mathbf{X} = \text{Encoder}(\mathbf{I}), \tag{1}$$

$$P\left(y_1, y_2, ..., y_n \mid \mathbf{X}\right) = \prod_{i=1}^{n} P\left(y_i \mid y_1, y_2, ..., y_n, \mathbf{X}\right), \tag{2}$$

where Encoder is the visual feature extractor, commonly CNN-based models or Faster R-CNN-based variants. $\mathbf{X} = \{x_1, x_2, ..., x_k\}$, where x_i is an object feature vector or a grid feature. Equation (2) describes the language model which is used to decode the visual feature $\mathbf{X}$ to the predicted sequence, where $\mathbf{Y}$ includes probabilities of predicted words and y_n is the probability of the end-of-sequence token. The language model can be variants of Long short-term memory and Transformer.

2.2. Overview of the current research situation in image captioning

Currently, in the world, there are about 05 famous datasets on this problem, which can be mentioned as COCO Caption [9], VizWiz [10], TextCaps [11], Conceptual Captions [12], Fashion Captioning [13]. Among them, the studies on image captioning problems often focus on the MS-COCO COCO Caption dataset [9]. The development sequence of the methods can be divided into two phases: before and after the Transformer [14] was born. Before Transformer was born, several approaches using RNN, LSTM, or Bi-LSTM and attention mechanisms were proposed to solve the problem, such as Show and Tell [15], Show, Attend and Tell [16], Bottom-Up Top-Down [17]. After Transformer was born with a powerful Self-Attention mechanism, almost all new methods are based on this architecture, which can be mentioned as AoANet [18], Object Relation Transformer [19], Meshed-Memory Transformer [20], X-Transformer [21], RSTNet [6]. In addition, several research directions have recently combined image and semantic features to embed in sequence generation models, such as Unified VLP [22] and VinVL [2].

2.3. Previous studies of image captioning in Vietnamese

Although research teams around the world continue to focus a lot of emphasis on this problem, the research scenario regarding the difficulty of coming up with image captioning phrases in Vietnam has not received much attention. At the ICCCI 2020 conference, Lam et al. published the UIT-ViIC dataset [7]. The dataset includes images taken from the COCO Caption dataset and labeled in Vietnamese; the image data domain is mainly sports context. To evaluate the performance of captioning models in the UIT-ViIC dataset, Lam et al. chose common baselines including Show, Attend and Tell [16] for training image captioning. The 8th International Conference on Language and Sound Processing hosted the VieCap4H competition in 2021, drawing teams to work on the challenge of creating Vietnamese image description sentences for photos in the healthcare domain. More than 90 different players participated in the competition. [8], which is essential for domestic research groups to study this problem in Vietnamese. The top -1 team UIT AI [8] chose the data augmentation approach to outperform all other teams. By comparing the semantics of images with terms in annotated captions from the VieCap4H dataset, they crawled external images. Furthermore, they also changed the structure of annotated captions in the dataset. They used the PhoBERT [23] model to ensure the quality of new captions and then added it to the original dataset for training. Finally, they model the image captioning problem as a sequence-to-sequence problem, which predicts the new word at time step t by the classical classification problem, and trained the captioning model with Cross-entropy loss. The top-2 GPT-Team [8] used the Clip-Clap model [24] to solve the problem by performing the image captioning and language translation tasks simultaneously. The top-3 Fruit AI teams additionally practiced multi-tasking. In more detail, they developed a model that can identify tampered sentences with false labels while predicting the following word. They built enhanced image-caption pairs by swapping out some of the original captions' characters with random characters in order to teach the model to recognize the corrupted words. These augmented image-caption pairs had bogus labels put on them.

Thus, up to now, only 02 datasets, UIT-ViIC and VieCap4H, have been officially published for the problem of generating Vietnamese image description sentences, which are also used to conduct the experiments in this study.

3. FEATURE EXTRACTION APPROACHES

3.1. Region features

3.1.1. General formulation for extracting region features

For feature extraction approaches, the concept of "region features," is followed, which takes the embedding vectors of the region of interest on an image as the input, which is fit into the captioning model. Given **I** is an RGB input image, the process of extracting region features, which is used in this study, can be formulated as the following Equation (3), (4), (5), (6), (7)

$$\mathcal{F}_t = \text{Backbone}(\mathbf{I}), \tag{3}$$

$$\mathcal{F}_b = \text{Neck}(\mathcal{F}_t), \tag{4}$$

$$\mathcal{R} = \text{RPN}(\mathcal{F}_b), \tag{5}$$

$$\mathcal{F}_r = \text{BoxExtractor}(\mathcal{R}, \mathcal{F}_b), \tag{6}$$

$$\mathcal{F}'_r = \text{FC}(\mathcal{F}_r), \tag{7}$$

where $\mathcal{F}_t \in \mathbb{R}^{C \times H \times W}$ is the top-down grid features obtained from the CNN-based backbone network. $\mathcal{F}_b$ is the bottom-up grid features obtained from the feature pyramid network (which is denoted as Neck). $\mathcal{R}$ is the set of N coordinates of the Region of Interests which possibly includes the visual objects which are proposed by the Regional Proposal Network (RPN). Then, based on these coordinates, the BoxExtractor is used to extract the embedding vectors of the regions of interest. The method to extract embedding vectors in BoxExtractor is RoIAlign [25], in which the size of each region is set as 7 × 7. After that, features $\mathcal{F}_r \in \mathbb{R}^{N \times C \times 7 \times 7}$ are obtained. I continue to fit them into the Fully Connected layers (FC) to transform the representation of each region from 3D space to 1D space; Finally the features $\mathcal{F}'_r \in \mathbb{R}^{N \times 1024}$ are obtained.

3.1.2. Experimental object detection models

The above process is the standard pipeline for recent two-based detection methods. Therefore, to extract region features, 04 state-of-the-art two-stage object detection methods are considered:

Faster R-CNN. Faster R-CNN was proposed by Ren et al. [3] based on the prior Fast R-CNN [26] and the addition of Region Proposal Network (RPN), being the well-established two-stage object detector. The proposed Region Proposal Network receives image features and outputs the region proposals for further bounding box regression and label classification. The authors train the RPN via binary class label assignment by minimizing the multi-task loss function which can be formulated as Equation (8)

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*), \tag{8}$$

where i is the index of an anchor in a mini-batch; p_i is the objectness score of an anchor; p_i^* label $\{0,1\}$ represents whether or not the anchor contains an object; t_i associates with a vector representing the horizontal bounding boxes coordinates; t_i^* is the ground-truth corresponding to the positive anchor.

Cascade R-CNN. Cai and Vasconcelos [4] proposed the high-quality detection model Cascade R-CNN. Multi-stage object detection architecture with set detectors trained in turn with the current detector's output as the input to the next detector. This scheme is used to solve the mismatch in quality between the output and the detector and the overfitting problem caused by the sensitive IoU threshold (when the IoU is large). However, creating a high-quality detector is not simply increasing the IoU during the training phase. In case the IoU threshold is increased, it also means that a significant decrease is witnessed in the number of active training samples. Different heads in the architecture designed for a particular IoU threshold, from small to large, are used at different stages (H1, H2, H3). Cascade regression is a resampling process that provides positive samples for further processing stages. This process can be formulated as Equation (9)

$$f(x,b) = f_T \circ f_{T-1} \circ ... \circ f_1(x,b), \tag{9}$$

T: total number of refining bounding box stages. Each f_T regressor in the cascade is optimized for the respective distribution b_T.

Grid R-CNN. Lu et al. [5] proposed the novel Grid R-CNN for a more accurate object detection. It adopts the well-known two-stage object detection pipeline; However, in the R-CNN head, the authors used the grid-guided mechanism instead of a normal fully-connected layer offset prediction for high-quality localization. The head outputs a probability heatmap via the FCN, which then later, the model can derive the grid points in the bounding boxes corresponding to the objects. Finally, by having high-level feature information extracted from the previous stage, it can easily determine the final bounding boxes using the feature fusion model. In the inference phase, the pixel with the highest confidence score will be used to map for location on the original image. As regards the point (H_x, H_y) in the heat map will be mapped with the point (I_x, I_y) on the image

$$I_x = P_x + \frac{H_x}{w_o} w_p, \tag{10}$$

$$I_y = P_y + \frac{H_y}{h_o} h_p. \tag{11}$$

3.2. Grid features

3.2.1. Overview

Besides region features, grid features are also used to represent images in the model space to investigate their effectiveness. While region features are embedding vectors of regions of interest proposed by the Regional Proposal Network (RPN), grid features are simply the features obtained from the last convolution block in the CNN-based models. Commonly, recent studies [6] used the ResNet-based or ResNeXt-based models for extracting grid features.

3.2.2. Experimental backbones

ResNeXt-based pre-trained model 1 (Jiang et al.). The study [1] is used to extract grid features; in detail, Jiang et al. use bottom-up, top-down architecture [17] to compute feature maps from lower blocks of ResNet to block C_4. However instead of using 14×14 `RoIPooling` to compute C_4 output features, then feeding to C_5 block and applying `AveragePooling` to compute per-region features, they convert the detector in [17] back to the CNN-based feature extractor to compute grid features at the same C_5 block. Through experiments, they observe that using converted C_5 block directly helps reduce computational time but achieves surprising results. Their `X152` pre-trained models is used for grid feature extraction.

ResNeXt-based pre-trained model 2 (VinVL) [2]. The pre-trained model VinVL [17] is also adopted, which is also based on the Bottom-Up model and uses ResNeXt as the backbone. This pre-trained model may produce richer feature maps because it was trained on the larger dataset leading to better pattern recognition ability. In detail, the VinVL used the ResNeXt backbone from the 1^{st} to the 4^{th} residual block as a feature extractor. After that, the feature maps fit into Regional Proposal Network (RPN) to produce region features. Each region feature is then applied to the 5^{th} residual block to obtain the final embedding vectors of regions of interest. The VinVL is a modified version of Bottom-up, which uses different classification heads to predict two more pieces of information: objects' attributes and objects' categories. Therefore, the VinVL model should be transformed into a unified backbone to obtain the grid features. For more detail, the feature maps obtained from the 4^{th} residual block are directly passed into the 5^{th} residual block to obtain the final grid features. Grid features obtained from both above backbones have the shape of $(H, W, 2048)$. Then, `AdaptiveAvgPool2D (7,7)` is applied to reshape from $(H, W, 2048)$ to $(7, 7, 2048)$. A single grid is flattened, then the final output has the shape of $(49, 2048)$. Besides using the VinVL pre-trained model as a grid features extractor, using it as a region features extractor is also experimented. In detail, its Regional Proposal Network produces regions of interest and applies fully connected layers to regress bounding boxes and classify the categories of predicted objects.

CLIP features [27]. CLIP is a robust model that is trained on a large corpus, including image-text pairs. In detail, CLIP is trained to predict $N \times N$ image-text pairs that happen given N input image-text pairs. Because of being trained to predict the occurrence of image-text pairs, the features of images obtained from CLIP include much valuable information. Therefore, in this study, CLIP features `ViT-L/14` pre-trained version is used to extract grid features. The pre-trained Vision Transformer provided in CLIP is considered as an image encoder for more detail. Then, the high-level representation $Z \in \mathrm{R}^{(1+g\times g)\times d}$ is obtained, where g is the grid size and 1 indicates the [CLS] token in Vision Transformer. the [CLS] token is removed and only use $Z' \in \mathrm{R}^{g\times g\times d}$ as the grid features. However, the features obtained from Vision Transformer `ViT-L/14` version have a grid size equal to 16. In this study [28], Wu et al. proved that a larger grid size might also obtain the same performance compared to a smaller grid size. Therefore, the grid size is reduced by applying `AdaptiveAvgPool2D (7,7)`.

4. CAPTIONING MODEL

4.1. Transformer-based model

As the primary architecture for training to generate captions, the Transformer [14] is used. Transformer architecture includes two crucial components: The Encoder and the decoder are two essential parts of the transformer architecture. The decoder is used to translate this presentation into a list of words, while the Encoder is used to learn the high-level latent space of the input feature (target predicted captions). In this section, the Transformer Encoder is revisited. Given $\mathbf{F} = \{f_1, f_2, ..., f_n\}$ denotes the feature vectors of the input image, it is first fitted into the feed-forward network to get the $\mathrm{d}_{\mathrm{model}}$ dimensional representation $\mathbf{Z} \in \mathbb{R}^{N \times d_{model}}$, where N is the number of feature vectors. They are then applied to multi-head attention mechanisms, skip connections, feed-forward networks, and normalization layers to produce the representation that includes the relation information between feature vectors. The encoder output is then passed into the decoder with the annotated caption (in the training stage) or the previously predicted words (in the inference stage) to generate the whole caption that describes the context of the input image. The architecture of the Transformer Decoder is similar to Encoder, but the future annotated words are masked. The architecture of the Transformer can shortly be formulated as Equation (12), (13)

$$\mathbf{Z} = \mathrm{Encoder}(\mathbf{F}), \tag{12}$$

$$\mathbf{Y} = \mathrm{Decoder}(\mathbf{Z}, \mathbf{H}), \tag{13}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times d_{model}}$ is the encoder output; $\mathbf{H}$ is the set of previously predicted words (inference) or the ground-truth caption (training); $\mathbf{Y}$ is the predicted caption.

The key idea of the Transformer is fitting all sequence components into the model, therefore self-attention mechanism is proposed to obtain the relation between each component and the rest. The self-attention (SelfAttention) can be formulated as Equation (14), (15)

$$Q = \mathbf{U} \times W^Q; K = \mathbf{U} \times W^K; V = \mathbf{U} \times W^V, \tag{14}$$

$$\mathbf{Z} = \mathrm{Softmax}(\frac{QK^T}{\sqrt{d_k}}) \times V, \tag{15}$$

where Q, K, and V are the Queries, Keys, and Values; W^Q, W^K and W^V are learned weights matrices; $\mathbf{Z}$ is the high-level presentation produced by the self-attention mechanism.

In practice, there are N_h self-attention heads for learning different aspects of the relationship between components and fused by the Concatenate operator. This process can be formulated as the Equation (16), (17)

$$\mathrm{head}_k = \mathrm{SelfAttention}_k(\mathbf{U}), k \in \{1, 2, ..., N_h\}, \tag{16}$$

$$\mathrm{MHSA}(\mathbf{U}) = \mathrm{Concatenate}(\mathrm{head}_1, \mathrm{head}_2, ..., \mathrm{head}_{N_h}). \tag{17}$$

In Transformer Encoder, $\mathbf{U}$ is the visual representation obtained from visual features $\mathbf{X}$. In Transformer Decoder, the $\mathbf{U}$ used as Queries is the input sequence (previous words or ground-truth caption), and the $\mathbf{U}$ used as Keys and Values is the encoder output.

4.2. Vietnamese adaptive decoding

Inspired by the study [6], in the decoding process, the multi-head self-attention mechanism in the Transformer Decoder is provided with the linguistic features of annotated or previous words, which helps the model generate non-visual words more naturally. First, the scheme in the study [6] is followed to train a BERT-based model. The pre-trained model PhoBERT-base [23] is used to adapt to the Vietnamese language. Then, language features of words are produced, concatenated with the visually encoded features from the encoder output, then fits into the multi-head self-attention mechanism in the Transformer Decoder. This process can be briefly formulated as Equation (18), (19), (20)

$$lf = \text{PhoBERT}(W), \tag{18}$$

$$Z_{enc} = \text{Concatnate}(Z_{enc}, lf), \tag{19}$$

$$Z_{dec} = \begin{cases} \text{MHSA}(W^E, Z_{enc}, Z_{enc}) & \text{if } l = 0 \\ \text{MHSA}(h_t, Z_{enc}, Z_{enc}) & \text{otherwise,} \end{cases} \tag{20}$$

where lf are the language features. Z_{enc} is the encoder output. W^E are word embedding vectors of input words. h_t is the high-level representation of words produced in the middle layers of the Transformer Decoder. l denotes the index of Transformer Decoder layers.

4.3. Multi-representation inference

The Multi-representation Inference scheme (MRI) is presented to enhance the ability to represent input images in the model space. In detail, in each type of region or grid feature, all models at the inference stage are used, and they are fused to obtain the final generated captions.

Given $\Theta = \{\theta_1, \theta_2, ..., \theta_N\}$ is the set of learned parameters of all trained models on different types of features; $\mathcal{F} = \{f_1, f_2, .., f_N\}$ is the set of different features representing the input images, the MRI scheme can be formulated as Equation (21)

$$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} \text{Transformer}_{\theta_i}(f_i), \tag{21}$$

where $\hat{Y}$ is the average of probabilities of words predicted from all models, it will be decoded to get the final captions.

Especially with region features, Θ is the set of learned parameters of models trained using region features extracted from Faster R-CNN, Cascade R-CNN, and Grid R-CNN extractor. Similarly, with grid features, Θ is the set of learned parameters of models trained using grid features extracted from ResNeXt-152 (Jiang et al.) and VinVL ResNeXt-152.

5. EXPERIMENTS

5.1. Benchmark datasets

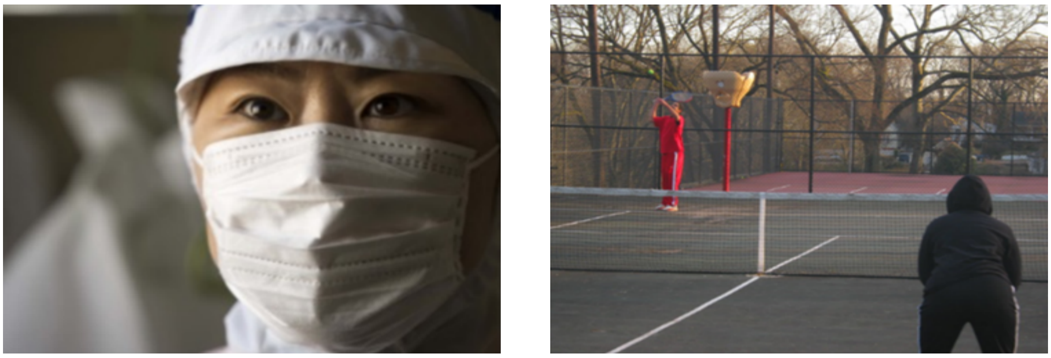
To evaluate the effectiveness of region features extracted from different object detection methods, two benchmark datasets are used for image captioning in Vietnamese: UIT-ViIC and VieCap4H.

Table 1: Data statistics of the VieCap4H dataset [8]

	Images	Captions	captions/image	Avg. length
Train	8,032	9,429	1.17	11.88
Public test	1,002	1,039	1.04	11.86
Private test	1,034	1,095	1.05	11.97
All	10,068	11,563	1.15	11.89

Table 2: Data statistics of the UIT-ViIC dataset [7]

	Images	Captions	Captions/image	Avg. length
Train	2,695	13,475	5	12.15
Validation	924	4,620	5	12.11
Test	231	1,155	5	12.32
All	3,850	19,250	5	12.15



GT Caption: Một người phụ nữ đội mũ và đeo khẩu trang màu trắng. *(A woman wearing a hat and a white mask.)*

GT Caption: Một người đàn ông và một người khác đang chơi tennis ở trên sân. *(A man and another person are playing tennis on the court.)*

Figure 1: An example image of VieCap4H and UIT-ViIC dataset

UIT-ViIC: This dataset was published by Lam et al., which contains 3,850 images from the MS-COCO Caption dataset, annotating captions in Vietnamese. The average length of each caption is approximately 10 to 15 words. The context of images in the dataset is almost related to sports balls. The statistic of the UIT-ViIC dataset is reported in Table 4.

VieCap4H: This dataset was released under the Vietnamese Language and Signal Processing club (VLSP) competition in 2021. This dataset includes 10,068 images with 11,563 captions in total, where 8,032 images are used for the public train set, 1,002 images are used for the public test set, and 1,034 images are used for the private test set. All images in the VieCap4H dataset are related to the healthcare domain, specially COVID-19. The statistic of the VieCap4H dataset is reported in Table 2.

Figure 1 illustrates two example images corresponding to two benchmark datasets.

5.2. Metrics

UIT-ViIC. Four standard metrics are used to evaluate the performance of the different feature extraction approaches in the UIT-ViIC dataset including BLEU (B) [29], METEOR (M) [30], ROUGE_L (R) [31], and CIDEr (C) [32]. BLEU is a well-known metric that is used to evaluate machine translation tasks. The METEOR score uses a fragmentation measure to assess the order of unigrams between hypothesis and reference captions, while the BLEU scores only consider the matching. The ROUGE-L score is the ROUGE metric that considers L-grams matching; this is designed as a recall-related metric because the denominator used to calculate the correct percentage is the total sum of the number of L-grams appearing in the reference captions. In contrast, this number in the BLEU score is the total number of n-grams that occur in hypothesis captions. SPICE score is the new metric that evaluates the reference and hypothesis captions based on the scene graph. The CIDEr score considers distinguishing the n-grams which are rare or common in the vocabulary via vectorizing them using TF-IDF [33] to obtain their frequency weights. Since the CIDEr score evaluates the diversity of words in generated captions, it is considered the most crucial metric in previous studies.

VieCap4H. On the VieCap4H dataset, The online evaluation is used on both the public and private-test sets, which the VLSP2021 VieCap4H challenge's organizer provided. METEOR, ROUGEL, and CIDEr are not used to evaluate this dataset because the ground truth is unavailable to the research community. Instead, the average of four BLEU scores Vedantam, Lawrence Zitnick, and Parikh (BLEU@1, BLEU@2, BLEU@3, BLEU@4) is used to measure the performance of different feature extraction approaches.

5.3. Implemental details

5.3.1. Feature extraction

For extracting region features, the pre-trained models of Faster R-CNN, Cascade R-CNN, and Grid R-CNN provided in the MMDetection toolbox [34] are used. The backbone config is ResNeXt-101 64x4d, and the checkpoints loaded into the models are obtained from the best during 24 training epochs. By default, 1000 regions of interest are detected by models on the image. However, only 49 regions with the highest objectness scores are used to represent the input images in the captioning model space. The dimension of each embedding vector is 1024. Therefore, with a single input image, I obtain the region features $\mathcal{F} \in \mathbb{R}^{49\times1024}$.

About the grid features, the pre-trained models on the ResNeXt-152 backbone provided in the studies (Jiang et al.) [1] and [2] (VinVL) are used. Because of obtaining the features from the last residual block of ResNeXt, each grid embedding vector has a dimension of 2048. After all, with a single input image, I obtain the grid features $\mathcal{F} \in \mathbb{R}^{49\times2048}$.

5.3.2. Transformer hyperparameters

In the Transformer model, there are 3 encoder layers to learn the high-level relationship latent space of the input features, and 3 decoder layers will decode these representations to the final generated captions. The critical component of the encoder and decoder is the multi-head attention mechanism, which includes h self-attention heads to learn the high-level relationship latent space. In this study, the number of self-attention heads is set as 8. The dimension of learned latent space in the encoder is set as 512.

5.3.3. Training scheme

To train the captioning model, the study [35] is followed, which proposed the training scheme consists of two phases. In phase 1, the model is updated with the learned parameters using the Cross-entropy loss function, which can be formulated as follows

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log\left(p_\theta\left(w_t^*|w_{1:t-1}^*\right)\right), \tag{22}$$

where θ denotes the learned parameters, and $w_{1:T}^*$ is the ground truth caption.

After that, Self-Critical Sequence Training (SCST) is applied, which uses the Reinforce policy algorithm [36] to directly refine the generated captions using the model trained with the Cross-entropy loss function

$$L_{RL}(\theta) = -E_{w_{1:T}\ p_\theta}\left[r(w_{1:T})\right], \tag{23}$$

where the reward $r(\cdot)$ denotes the CIDEr score.

5.4. Experimental results

The experimental results on the VieCap4H public test and private test in Table 3, and the UIT-ViIC test set are reported in Table 4. First, I discuss the results of two test sets of the VieCap4H dataset. Generally, the model trained using the grid features as image representations seems to perform much better than region features. The highest results on both test sets using a single type of feature are recorded in VinVL pre-trained model (public test: 29.2057% and private test: 27.5677%). In contrast, region features' highest results are recorded at Grid R-CNN (ResNeXt-101) (public test: 25.098% and private test: 24.3787%). This observation can be explained by the fact that the object detection models used to extract region features were trained only MS-COCO dataset, including 2,500,000 annotated objects, while the grid extractors are trained on the Visual genome dataset, which includes 3,843,636 annotated objects. Therefore, grid extractors see and learn objects in the training stage, which helps extracted grid features that are richer than region features, and contain more valuable information that can effectively represent the images. Moreover, the research community commonly would not train the feature extraction task directly on the dataset used to evaluate the captioning task because it would cost much time and resources. Therefore, pre-trained models are usually used to extract features, and image captioning is considered a downstream task. Consequently, when using object detection models trained on the MS-COCO dataset, they could not perform well when inferencing the VieCap4H dataset. The detected regions may include lots of false-positive ones. In this case, grid features will perform better because using grid features does not force the model to pay attention to any specific regions on the image; the model will learn the patterns of the global information of images instead. These observations reverse when using VinVL ResNeXt-152 to extract region features. The VinVL model is trained on large-scale datasets, leading to better pattern recognition. The features from ResNeXt-152 include lots of valuable information, which helps the Regional Proposal Network recognize regions of interest effectively. However, when using CLIP as the grid features extractor, the obtained results can not be compared to other types of grid features. The reason that can be explained is that the lacking of semantic features, such as class names of objects that appeared in the scene, affects the performance of CLIP because it is trained on the dataset, including image-caption pairs. Another reason that can be mentioned is the significant mismatch between data domains lead to the bad performance of pattern recognition.

Table 3: The experimental results of the VieCap4H public test and private test. The best performance is marked in boldface.

Visual Features	R	G	Avg. BLEU@[1:4] (%)	
			Public-test	Private-test
Faster R-CNN (FR)	✓		24.0025	24.0915
Cascade R-CNN (CR)	✓		24.7853	23.2285
Grid R-CNN (GR)	✓		25.098	23.9652
MI (FR + CR + GR)	✓		25.5304	24.3787
VinVL ResNeXt-152 (V-X152)	✓		**29.593**	28.3472
CLIP ViT-L/14		✓	25.5846	24.0066
ResNeXt-152 (X152)		✓	29.176	27.336
VinVL RexNeXt-152 (V-X152)		✓	29.2057	27.5677
MI (X152 + V-X152)		✓	29.2331	**28.8211**

Table 4: The experimental results on the UIT-ViIC test set. The best performance is marked in boldface.

Visual Features	R	G	BLEU@4 (%)	METEOR (%)	ROUGE-L (%)	CIDEr (%)
Faster R-CNN (FR)	✓		40.7201	34.9301	60.9917	112.7668
Cascade R-CNN (CR)	✓		41.2815	34.7708	61.1504	114.0168
Grid R-CNN (GR)	✓		38.9171	33.9351	59.8029	107.9405
MI (FR + CR + GR)	✓		41.5974	35.4054	61.5079	117.8272
CLIP ViT-L/14	✓		43.6317	35.03	62.8088	117.5339
ResNeXt-152 (X152)		✓	**47.0248**	36.3433	63.9701	128.7717
VinVL RexNeXt-152 (V-X152)		✓	44.3001	35.7373	62.9395	120.9649
MI (X152 + V-X152)		✓	46.9374	**36.7348**	**65.1932**	**133.2868**
VinVL ResNeXt-152		✓	47.2961	**37.0732**	**64.8361**	**137.1243**

The performance of different features on the UIT-ViIC dataset shows similar results with the VieCap4H. Using grid features as image representations helps the captioning model perform better. The highest results using a single type of feature are recorded using ResNeXt-152 (Jiang et al.) pre-trained model as feature extractor (BLEU@4 47.0248%, METEOR 36.3433%, ROUGE-L 63.9701%, CIDEr 128.7717%).

The results are boosted when the Multi-representation Inference scheme (MI) is used. On the VieCap4H dataset, the MI scheme, which considers predictions on all models trained on region features extracted from Faster R-CNN, Cascade R-CNN, and Grid R-CNN, helped obtain the results of 25.5304% on the public test and 24.3787% on private-test, which performs better than the single model trained on Grid R-CNN region features (+0.4324% on public-test and +0.4135$ on private-test). The combination of predictions from models trained on-grid features extracted from ResNeXt-152 (Jiang et al.) and the VinVL pre-trained model outperforms the single model trained on VinVL grid features by 1.2534% higher.

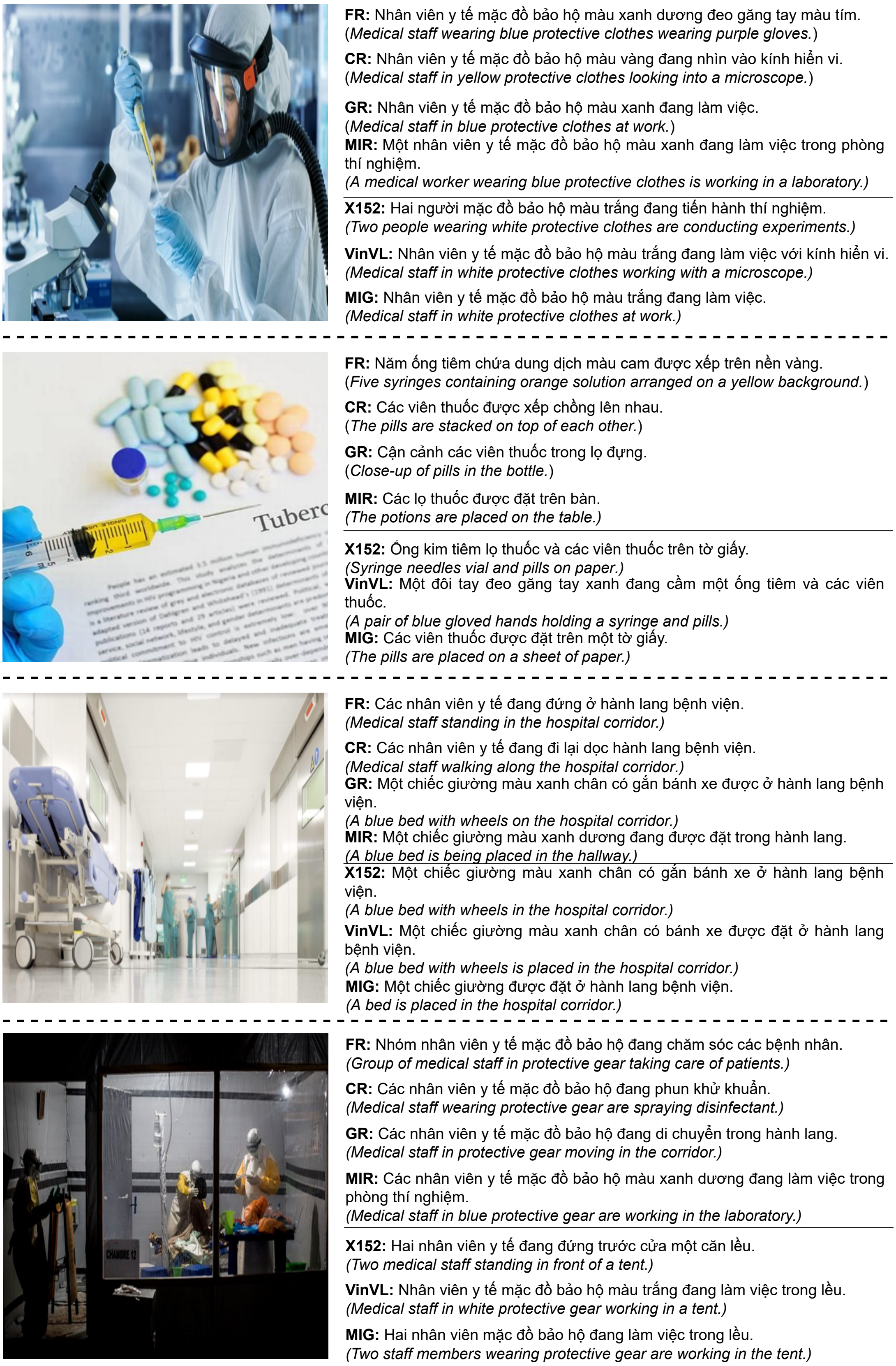
MI scheme also performs well on the UIT-ViIC test set. On region features, the results are +0.3159%, +0.6346%, +0.3575%, and +3.8104% higher than the single model trained on Cascade R-CNN region features on BLEU4, METEOR, ROUGE-L, and CIDEr metrics, respectively. The combination of grid features performs better than the single model trained on ResNeXt-152 grid features, whose results are +0.3915%, +1.2231%, and +4.5151% higher on BLEU4, METEOR, ROUGE-L, and CIDEr metrics, respectively. The obtained results prove that the MI scheme takes advantage of all valuable information in all grid or region features. The average probability of predicted words allows the model to generate more natural and high-quality captions.

5.5. Qualitative results

5.5.1. Generated captions

To analyze deeper the Qualitative results more, the generated captions on example images of both VieCap4H (Figure 2) and UIT-ViIC (Figure 3) datasets are provided. I denote FR, CR, and GR for the models trained on region features extracted from Faster R-CNN, Cascade R-CNN, and Grid R-CNN, respectively. X152 denotes the model trained on-grid features extracted from the pre-trained model of Jiang et al., while VinVL denotes the VinVL pre-trained model. On VieCap4H, it can be observed that models trained on-grid features are aware of the color better than region features. In the first sub-figure, FR and CR models recognized the color of protective clothes as "blue," the GR model recognized it as "yellow," while the person is wearing the white protective clothes, which is correctly recognized by the X152 model and the VinVL model. Moreover, the grid features seem to represent the information around space better. In the second sub-figure, FR recognized that "syringes" are "arranged on a yellow background", GR mislook medicines are included in "bottle". In contrast, X152 recognized exactly that "medicines" are "on paper", and VinVL stated that there are "a pair of blue gloved hands holding a syringe".

Now, let's move to discuss the generated captions of some example images in the UIT-ViIC dataset illustrated in Figure 3. The first sub-figure shows that generated captions from models trained on region features are more similar to ground-truth captions than those trained with grid features. It can be explained that in the context of including lots of apparent visual objects, the region features can perform better. Because many people are wearing the competition uniforms and standing on the field with a few-noise background, the object detectors recognize the objects quite well, leading to high-quality captions. In the second sub-figure, the FR model correctly recognized that people are "riding" elephants and playing "football", which is an action hardly seen in the dataset. However, the number of people is detected wrongly; the exact number is two. A similar problem also appears in the generated captions from models trained using grid features. In the third sub-figure, generated captions from models trained using grid features recognized the correct action of the boy, which is "swinging the bat".



FR: Nhân viên y tế mặc đồ bảo hộ màu xanh dương đeo găng tay màu tím.
(*Medical staff wearing blue protective clothes wearing purple gloves.*)

CR: Nhân viên y tế mặc đồ bảo hộ màu vàng đang nhìn vào kính hiển vi.
(*Medical staff in yellow protective clothes looking into a microscope.*)

GR: Nhân viên y tế mặc đồ bảo hộ màu xanh đang làm việc.
(*Medical staff in blue protective clothes at work.*)

MIR: Một nhân viên y tế mặc đồ bảo hộ màu xanh đang làm việc trong phòng thí nghiệm.
(A medical worker wearing blue protective clothes is working in a laboratory.)

X152: Hai người mặc đồ bảo hộ màu trắng đang tiến hành thí nghiệm.
(Two people wearing white protective clothes are conducting experiments.)

VinVL: Nhân viên y tế mặc đồ bảo hộ màu trắng đang làm việc với kính hiển vi.
(Medical staff in white protective clothes working with a microscope.)

MIG: Nhân viên y tế mặc đồ bảo hộ màu trắng đang làm việc.
(Medical staff in white protective clothes at work.)

FR: Năm ống tiêm chứa dung dịch màu cam được xếp trên nền vàng.
(*Five syringes containing orange solution arranged on a yellow background.*)

CR: Các viên thuốc được xếp chồng lên nhau.
(*The pills are stacked on top of each other.*)

GR: Cận cảnh các viên thuốc trong lọ đựng.
(*Close-up of pills in the bottle.*)

MIR: Các lọ thuốc được đặt trên bàn.
(The potions are placed on the table.)

X152: Ống kim tiêm lọ thuốc và các viên thuốc trên tờ giấy.
(Syringe needles vial and pills on paper.)
VinVL: Một đôi tay đeo găng tay xanh đang cầm một ống tiêm và các viên thuốc.
(A pair of blue gloved hands holding a syringe and pills.)
MIG: Các viên thuốc được đặt trên một tờ giấy.
(The pills are placed on a sheet of paper.)

FR: Các nhân viên y tế đang đứng ở hành lang bệnh viện.
(Medical staff standing in the hospital corridor.)

CR: Các nhân viên y tế đang đi lại dọc hành lang bệnh viện.
(Medical staff walking along the hospital corridor.)
GR: Một chiếc giường màu xanh chân có gắn bánh xe được ở hành lang bệnh viện.
(A blue bed with wheels on the hospital corridor.)
MIR: Một chiếc giường màu xanh dương đang được đặt trong hành lang.
(A blue bed is being placed in the hallway.)
X152: Một chiếc giường màu xanh chân có gắn bánh xe ở hành lang bệnh viện.
(A blue bed with wheels in the hospital corridor.)
VinVL: Một chiếc giường màu xanh chân có bánh xe được đặt ở hành lang bệnh viện.
(A blue bed with wheels is placed in the hospital corridor.)
MIG: Một chiếc giường được đặt ở hành lang bệnh viện.
(A bed is placed in the hospital corridor.)

FR: Nhóm nhân viên y tế mặc đồ bảo hộ đang chăm sóc các bệnh nhân.
(Group of medical staff in protective gear taking care of patients.)

CR: Các nhân viên y tế mặc đồ bảo hộ đang phun khử khuẩn.
(Medical staff wearing protective gear are spraying disinfectant.)

GR: Các nhân viên y tế mặc đồ bảo hộ đang di chuyển trong hành lang.
(Medical staff in protective gear moving in the corridor.)

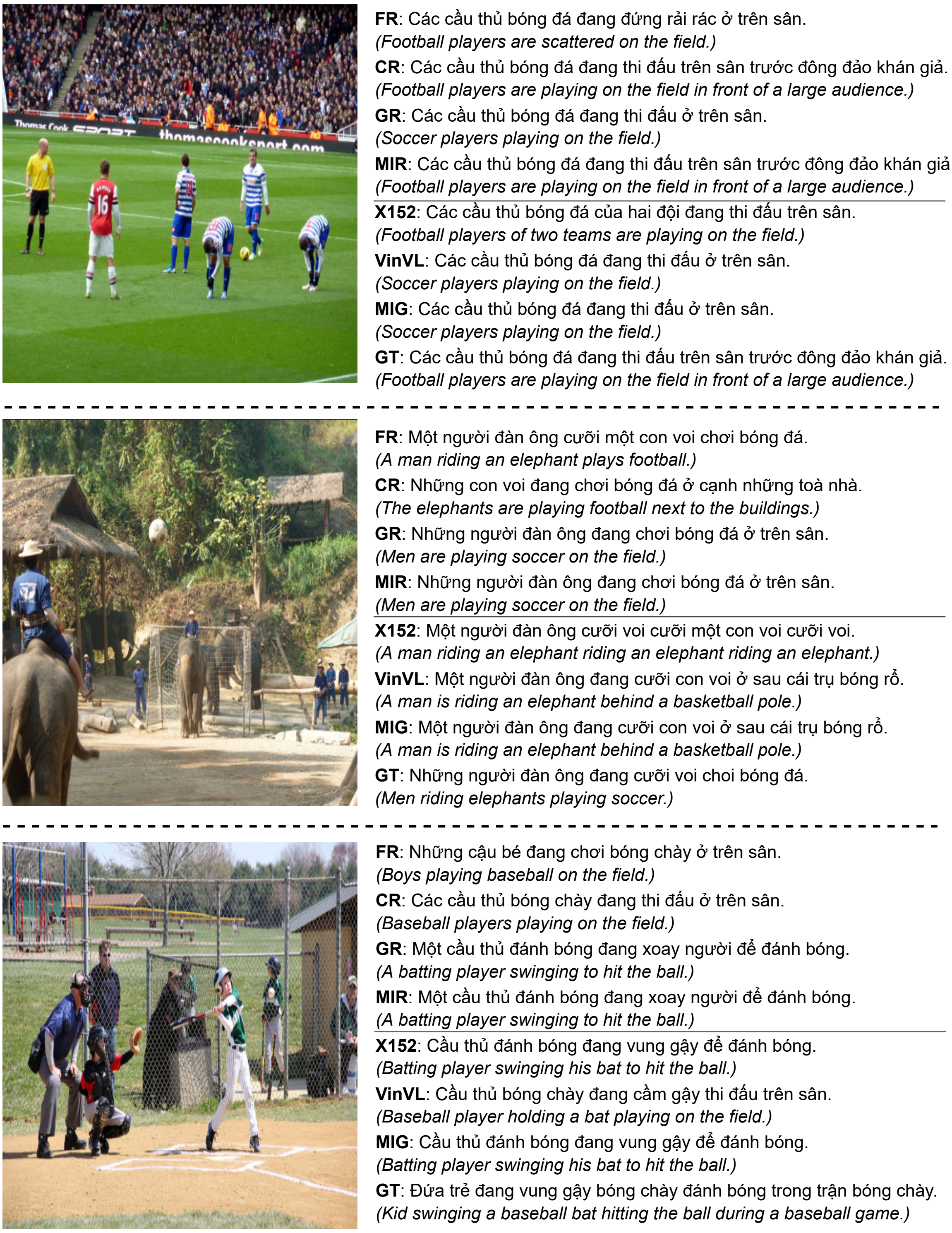
MIR: Các nhân viên y tế mặc đồ bảo hộ màu xanh dương đang làm việc trong phòng thí nghiệm.
(Medical staff in blue protective gear are working in the laboratory.)

X152: Hai nhân viên y tế đang đứng trước cửa một căn lều.
(Two medical staff standing in front of a tent.)

VinVL: Nhân viên y tế mặc đồ bảo hộ màu trắng đang làm việc trong lều.
(Medical staff in white protective gear working in a tent.)

MIG: Hai nhân viên mặc đồ bảo hộ đang làm việc trong lều.
(Two staff members wearing protective gear are working in the tent.)

Figure 2: The qualitative comparison between generated captions between feature extraction approaches on the VieCap4H private-test set (GT is not available).



FR: Các cầu thủ bóng đá đang đứng rải rác ở trên sân.
(Football players are scattered on the field.)
CR: Các cầu thủ bóng đá đang thi đấu trên sân trước đông đảo khán giả.
(Football players are playing on the field in front of a large audience.)
GR: Các cầu thủ bóng đá đang thi đấu ở trên sân.
(Soccer players playing on the field.)
MIR: Các cầu thủ bóng đá đang thi đấu trên sân trước đông đảo khán giả.
(Football players are playing on the field in front of a large audience.)

X152: Các cầu thủ bóng đá của hai đội đang thi đấu trên sân.
(Football players of two teams are playing on the field.)
VinVL: Các cầu thủ bóng đá đang thi đấu ở trên sân.
(Soccer players playing on the field.)
MIG: Các cầu thủ bóng đá đang thi đấu ở trên sân.
(Soccer players playing on the field.)
GT: Các cầu thủ bóng đá đang thi đấu trên sân trước đông đảo khán giả.
(Football players are playing on the field in front of a large audience.)

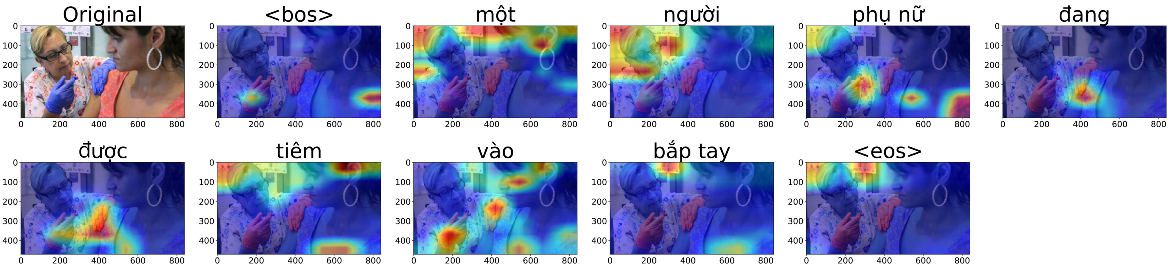
FR: Một người đàn ông cưỡi một con voi chơi bóng đá.
(A man riding an elephant plays football.)
CR: Những con voi đang chơi bóng đá ở cạnh những toà nhà.
(The elephants are playing football next to the buildings.)
GR: Những người đàn ông đang chơi bóng đá ở trên sân.
(Men are playing soccer on the field.)
MIR: Những người đàn ông đang chơi bóng đá ở trên sân.
(Men are playing soccer on the field.)

X152: Một người đàn ông cưỡi voi cưỡi một con voi cưỡi voi.
(A man riding an elephant riding an elephant riding an elephant.)
VinVL: Một người đàn ông đang cưỡi con voi ở sau cái trụ bóng rổ.
(A man is riding an elephant behind a basketball pole.)
MIG: Một người đàn ông đang cưỡi con voi ở sau cái trụ bóng rổ.
(A man is riding an elephant behind a basketball pole.)
GT: Những người đàn ông đang cưỡi voi choi bóng đá.
(Men riding elephants playing soccer.)

FR: Những cậu bé đang chơi bóng chày ở trên sân.
(Boys playing baseball on the field.)
CR: Các cầu thủ bóng chày đang thi đấu ở trên sân.
(Baseball players playing on the field.)
GR: Một cầu thủ đánh bóng đang xoay người để đánh bóng.
(A batting player swinging to hit the ball.)
MIR: Một cầu thủ đánh bóng đang xoay người để đánh bóng.
(A batting player swinging to hit the ball.)

X152: Cầu thủ đánh bóng đang vung gậy để đánh bóng.
(Batting player swinging his bat to hit the ball.)
VinVL: Cầu thủ bóng chày đang cầm gậy thi đấu trên sân.
(Baseball player holding a bat playing on the field.)
MIG: Cầu thủ đánh bóng đang vung gậy để đánh bóng.
(Batting player swinging his bat to hit the ball.)
GT: Đứa trẻ đang vung gậy bóng chày đánh bóng trong trận bóng chày.
(Kid swinging a baseball bat hitting the ball during a baseball game.)

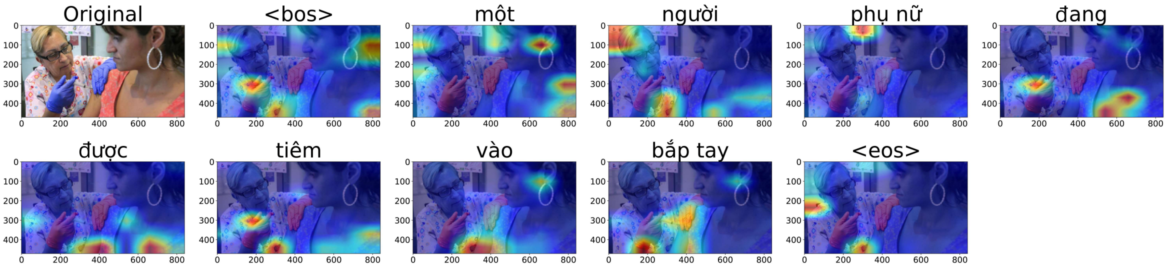
Figure 3: The qualitative comparison between generated captions between feature extraction approaches on the UIT-ViIC test set (GT is available).



Original <bos> một người phụ nữ đang

được tiêm vào bắp tay <eos>

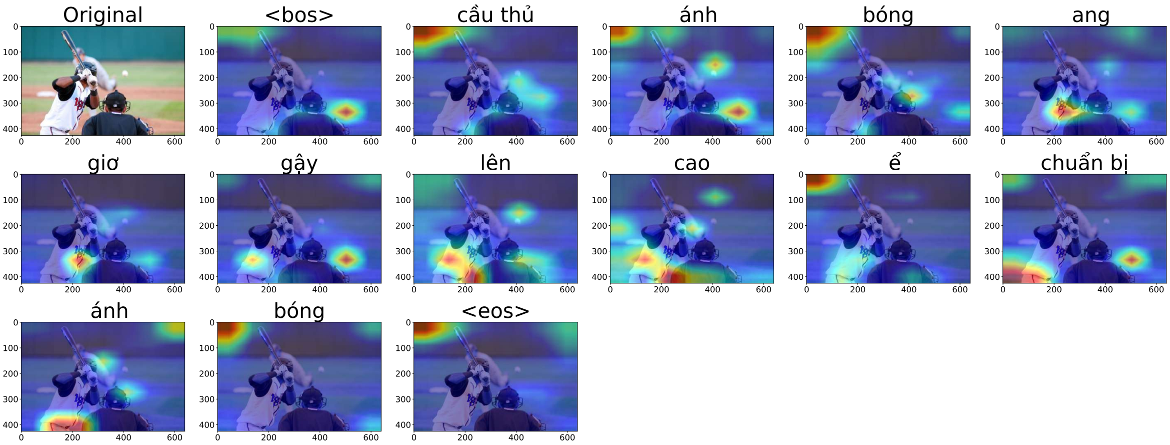
Figure 4: Visualization of attention scores at each decoding step using grid features extracted from ResNeXt-152 (Jiang et al.) on an example image in VieCap4H private-test set. Please 4× zoom out for better observation. (English translation: A woman is being injected into her biceps)



Original <bos> một người phụ nữ đang

được tiêm vào bắp tay <eos>

Figure 5: Visualization of attention scores at each decoding step using grid features extracted from VinVL ResNeXt-152 on an example image in VieCap4H private-test set. Please 4× zoom out for better observation.. (English translation: A woman is being injected into her biceps)



Original <bos> cầu thủ ánh bóng ang

giơ gậy lên cao ể chuẩn bị

ánh bóng <eos>

Figure 6: Visualization of attention scores at each decoding step using grid features extracted from ResNeXt-152 (Jiang et al.) on an example image in UIT-ViIC test set. Please 4× zoom out for better observation. (English translation: A woman is being injected into her biceps)

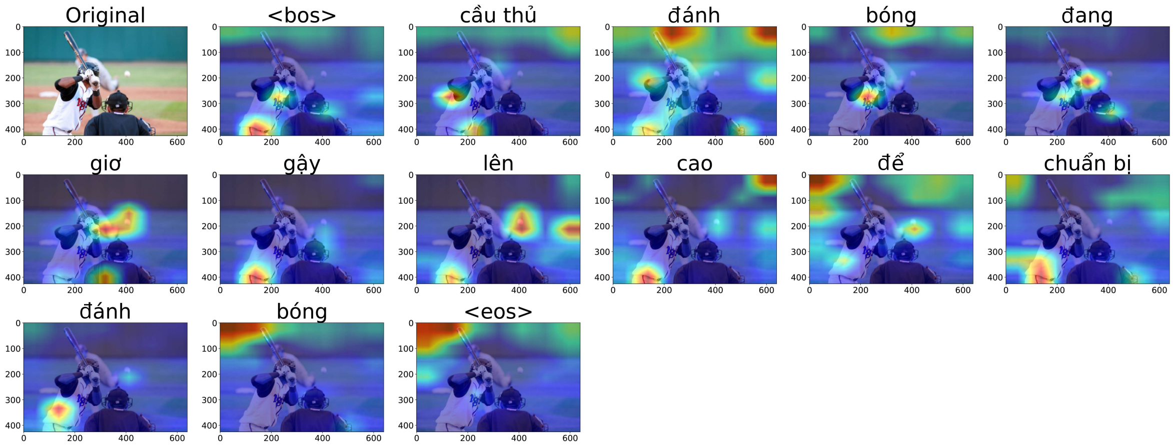


Figure 7: Visualization of attention scores at each decoding step using grid features extracted from VinVL ResNeXt-152 on an example image in VieCap4H private-test set. Please 4× zoom out for better observation. (English translation: A woman is being injected into her biceps)

5.5.2. Attention scores

As the results reported in Tables 4 and 3, the grid features better represent the input images in the model space. Therefore, the visualization of attention score at each decoding time step of models trained using grid features is provided in Figures 4, 5, 6 and 7. Generally, the pre-trained VinVL ResNeXt-152 [2] model guides the models on where to focus better than the ResNeXt-152 (Jiang et al.) [1]. Figure 4 shows that the positions with high attention scores are unrelated to the generated words. It seems that the grid features extracted from ResNeXt-152 (Jiang et al.) [1] did not represent the images well; the crucial features of images are not emphasized. Figure 5 proves that the pre-trained VinVL model [2] performs the feature extraction better, and the high scores' positions are well related to the generated words. The same observation also appeared in the example images from the UIT-ViIC dataset, which are illustrated in Figure 6 and 7.

6. CONCLUSION

In conclusion, the research goal of this study is to explore the effectiveness of representations of images in the model space. Therefore, the empirical study of feature extraction approaches for image captioning is presented. Two approaches used in this study are extracting grid features and region features. A Transformer-based model was used to train the captioning task on different features. Moreover, I focus on image captioning in Vietnamese; therefore, a series of experiments are conducted on two Vietnamese benchmark datasets: VieCap4H and UIT-ViIC. The obtained results showed that the grid features help the primary captioning model perform better in all metrics. In qualitative results, it can be investigated that region features may achieve better in some specific cases. Finally, the attention scores also strengthen the effectiveness of using grid features as image representations in the model space.

ACKNOWLEDGMENT

This work was supported by the Multimedia Processing Lab (MMLab) at the University of Information Technology, VNUHCM.

REFERENCES

[1] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller, and X. Chen, "In defense of grid features for visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 267–10 276.

[2] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5579–5588.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[4] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162.

[5] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7363–7372.

[6] X. Zhang, X. Sun, Y. Luo, J. Ji, Y. Zhou, Y. Wu, F. Huang, and R. Ji, "Rstnet: Captioning with adaptive attention on visual and non-visual words," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 465–15 474.

[7] Q. H. Lam, Q. D. Le, V. K. Nguyen, and N. L.-T. Nguyen, "Uit-viic: A dataset for the first evaluation on vietnamese image captioning," in *International Conference on Computational Collective Intelligence.* Springer, 2020, pp. 730–742.

[8] T. M. Le, L. H. Dang, T.-S. Nguyen, T. M. H. Nguyen, and X.-S. Vu, "VLSP 2021 - VieCap4H Challenge: Automatic image caption generation for healthcare domain in Vietnamese," in *Proceedings of the 8th International Workshop on Vietnamese Language and Speech Processing.* Ho Chi Minh, Vietnam: VNU Journal of Science: Computer Science and Communication Engineering, 12 2021.

[9] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[10] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3608–3617.

[11] O. Sidorov, R. Hu, M. Rohrbach, and A. Singh, "Textcaps: A dataset for image captioning with reading comprehension," in *European Conference on Computer Vision.* Springer, 2020, pp. 742–758.

[12] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565.

[13] X. Yang, H. Zhang, D. Jin, Y. Liu, C.-H. Wu, J. Tan, D. Xie, J. Wang, and X. Wang, "Fashion captioning: Towards generating accurate descriptions with semantic rewards," in *European Conference on Computer Vision.* Springer, 2020, pp. 1–17.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning.* PMLR, 2015, pp. 2048–2057.

[17] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6077–6086.

[18] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4634–4643.

[19] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 578–10 587.

[21] Y. Pan, T. Yao, Y. Li, and T. Mei, "X-linear attention networks for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 971–10 980.

[22] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and VQA," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 041–13 049.

[23] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," *arXiv preprint arXiv:2003.00744*, 2020.

[24] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[26] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning.* PMLR, 2021, pp. 8748–8763.

[28] M. Wu, X. Zhang, X. Sun, Y. Zhou, C. Chen, J. Gu, X. Sun, and R. Ji, "Difnet: Boosting visual information flow for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 020–18 029.

[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 of the Association for Computational Linguistics*, 2002, pp. 311–318.

[30] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 65–72.

[31] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004, pp. 74–81.

[32] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4566–4575.

[33] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF," *Journal of Documentation*, 2004.

[34] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[35] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008–7024.

[36] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992.

Received on October 01, 2022
Accepted on November 21, 2022