Ho Chi Minh city - University of Science
Faculty of Information Technology

Final Report

TEXT MINING

LECTURER

- Lê Thanh Tùng
- Nguyễn Trần Duy Minh

PREPARED BY

- 22127220 - Nguyễn Anh Kiệt
- 22127275 - Trần Anh Minh
- 22127280 - Đoàn Đặng Phương Nam
- 22127353 - Cao Minh Quang
- 22127398 - Nguyễn Văn Minh Thiện

Ho Chi Minh city, 2025

Contents

1. Abstract

In recent years, Visual Question Answering (VQA) has emerged as a key task at the intersection of computer vision and natural language processing. However, existing datasets and methods largely focus on general natural images and English-language contexts, leaving a significant gap in supporting more complex visual domains and low-resource languages. In this paper, we introduce ViInfographicsVQA, the first large-scale Vietnamese VQA dataset designed specifically for infographics, a domain rich in both textual and visual elements. The dataset comprises over 34,000 infographic images and more than 139,000 question-answer pairs, automatically generated and categorized into text-based and non-text-based questions to reflect the multimodal nature of real-world infographics. To address the challenges posed by ViInfographicsVQA, we propose two model architectures. The first is a lightweight baseline that relies solely on visual features. The second is a more advanced pipeline that dynamically determines whether to incorporate embedded text from OCR into the reasoning process. Our proposed framework integrates EfficientNet and BLIP-2 for visual representation, BARTpho for Vietnamese language understanding, and a novel Bi-directional Cross Attention mechanism for multimodal fusion. Experimental results show that the second approach significantly outperforms the baseline across all evaluation metrics (BLEU-4, ROUGE, and BERTScore), particularly on text-based questions, demonstrating the critical role of OCR-aware reasoning in this domain. We believe that ViInfographicsVQA offers a valuable benchmark for future research in infographic understanding and Vietnamese multimodal learning. The dataset[1], code[2], and model checkpoints[3] for our experiments are publicly available to encourage further research on Vietnamese infographic-based VQA and multimodal learning.

Keywords: Visual Question Answering, Infographics, OCR, Vietnamese, Multimodal Fusion, Low-resource Language

2. Introduction

Visual Question Answering (VQA) is a multifaceted task at the intersection of computer vision and natural language processing [1]. It requires models to comprehend and reason over visual content to answer questions posed in natural language. This task has garnered significant attention due to its wide-ranging applications, including assistive technologies, educational tools, and information retrieval systems.

Over the years, numerous VQA datasets have been developed to advance research in this domain. Notable among these are VQAv2.0, which provides a diverse set of images and questions to challenge models' understanding of visual scenes. However, a majority of these datasets focus on natural images and general scenes, leaving specialized domains like infographics underrepresented. Infographics, characterized by their rich combination of textual and visual elements, present unique challenges for VQA systems, necessitating the integration of multimodal information and complex reasoning capabilities.

Furthermore, the landscape of VQA research is predominantly centered around the English language, with limited resources available for other languages. Vietnamese, in particular, is considered a low-resource language in this context. While there have been efforts to develop Vietnamese VQA datasets, such as ViVQA [2], OpenViVQA [3], ViTextVQA [4], ViOCRVQA [5], ViCLEVR [6], etc.

[1] *https://huggingface.co/datasets/Namronaldo2004/ViInfographicsVQA*

[2] *https://github.com/Namronaldo08102004/CSC15105-ViInfographicsVQA*

[3] *https://huggingface.co/Namronaldo2004/ViInfographicsVQA*

These resources primarily focus on general or text-based images and do not address the unique challenges posed by infographics.

Recognizing this gap, we introduce ViInfographicsVQA, the first Vietnamese VQA dataset specifically designed for infographics. Our motivation stems from the observation that infographics often contain dense information distributed across various regions, requiring models to perform localized reasoning and counting tasks. Counting, in particular, remains a challenging aspect of VQA, as it demands precise identification and enumeration of relevant elements within an image. By focusing on counting questions within infographics, we aim to push the boundaries of current VQA systems and encourage the development of models capable of nuanced visual reasoning.

In addition, within this study, we conduct experiments using two distinct model pipelines on the ViInfographicsVQA dataset to demonstrate its applicability and challenge level. The first pipeline integrates BARTpho, a Vietnamese language model, for processing input questions, with a novel combination of a CNN-based model for capturing local features and a Transformer-based model for extracting global features, inspired by recent advancements in multimodal architectures [7]. This pipeline is further enhanced by the integration of a Hierarchical Co-Attention mechanism [8], which enables the model to jointly reason about the attention across both the question and the visual content, facilitating a deeper and more contextualized understanding. The second pipeline focuses on adapting the first architecture to effectively address non-text-based tasks, while introducing a specialized mechanism for text-based tasks by incorporating OCR tools. These tools extract embedded textual information from the infographic, which is then processed as an auxiliary input to support the answering of questions that rely on textual content. This dual-pipeline approach demonstrates the flexibility of ViInfographicsVQA in accommodating a range of question types and highlights the necessity for models to simultaneously reason over both structured visual data and natural language, a hallmark challenge of infographic-based VQA.

While our dataset is synthetically generated, we have incorporated human oversight to ensure the quality and relevance of the data. We acknowledge the limitations inherent in synthetic data but believe that ViInfographicsVQA serves as a valuable benchmark for exploring the complexities of VQA in the context of infographics and the Vietnamese language. We hope this resource will spur further research and innovation in this underexplored area.

3. Related Works

In this section, we present an overview of existing datasets and methods related to Visual Question Answering (VQA) in multimodal contexts. First, we examine representative datasets that aim to combine textual and visual modalities, and discuss their shortcomings, particularly in scenarios requiring tightly coupled reasoning between image regions and embedded textual content. We then shift focus to Vietnamese-language VQA datasets, as well as datasets targeting the specific task of counting, which remains a challenging yet underexplored area. Finally, we survey prominent methods proposed to tackle VQA tasks, with attention to both general approaches and those specialized for infographic-style images in both English and Vietnamese settings.

3.1. VQA Datasets

3.1.1. Former VQA Datasets in a Multimodal Context

Mathew et al. [9] (2021) introduced the InfographicsVQA benchmark, specifically designed for visual question answering on infographic-style images. As part of their study, they reviewed a wide range of previous VQA datasets and identified several limitations that their proposed dataset sought to overcome. For instance, Textbook Question Answering (TQA) [10] and RecipeQA [11] were created for multimodal QA in structured domains such as educational content and cooking procedures, but these datasets present textual information separately from images in a machine-readable format. This separation reduces the need for models to jointly reason over visual and textual elements. In contrast, ST-VQA [12] and TextVQA [13] address scene text understanding in natural images, yet their sparse text content and uncontrolled visual settings limit their relevance to scenarios where text-rich and well-organized layouts are essential. OCR-VQA [14] centers on book covers and generates questions from structured metadata like author names and titles, which often minimizes the necessity of genuine visual reasoning. Meanwhile, datasets such as DVQA [15], FigureQA [16], and LEAF-QA [17] focus on charts and plots but rely heavily on synthetic imagery and templated question formats, lacking the complexity and variability seen in real-world infographic layouts. DocVQA [18] deals with scanned business and industrial documents and remains strictly extractive, requiring answers to be pulled directly from visible text elements without broader reasoning. VisualMRC [19] adopts an abstractive approach to VQA using webpage screenshots, introducing higher-level inference demands, though it is still limited in domain diversity. Collectively, while these datasets have significantly advanced multimodal VQA research, Mathew et al. [9] pointed out that they often simplify the visual-textual interaction or focus on narrow domains, making them insufficient for modeling the rich, structured, and densely interwoven content typical of infographics.

3.1.2. VQA Datasets in Vietnamese

Vietnamese VQA is an emerging field, with several datasets exploring diverse aspects of the task. However, it is important to note that, to date, no Vietnamese VQA dataset has focused specifically on infographic-based content, making ViInfographicsVQA the first to address this unique domain. Among notable efforts, Tran et al. [2] (2021) introduced ViVQA, which leverages 10,328 images from the MS COCO dataset along with 15,000 Vietnamese question-answer pairs to benchmark general-purpose visual question answering. Focusing on textual understanding, Nguyen et al. [4] (2024) proposed ViTextVQA, which includes over 16,000 images and around 50,000 questions to evaluate models' ability to comprehend embedded text. Viet-Doc-VQA [20] , released by 5CD-AI (2024), is constructed from 51,856 Vietnamese textbook pages and provides 310,952 question-answer pairs generated via Gemini 1.5 Flash, targeting educational and document-based contexts. Similarly, Pham et al. [5] (2024) developed ViOCRVQA, an OCR-centric VQA dataset featuring 28,000 images and 120,000 question-answer pairs focused on Vietnamese textual content in visual scenes. For open-ended QA, Nguyen et al. [3] (2023) introduced OpenViVQA, offering 11,000 images and 37,000 question-answer pairs to support more flexible answer generation. Viet-OCR-VQA [21], also curated by 5CD-AI (2024), is currently the largest of its kind, with over 137,000 images and 822,679 questions generated using Gemini 1.5 Flash. These datasets collectively advance Vietnamese VQA research by addressing challenges in optical character recognition, cultural and linguistic diversity, and multimodal reasoning. Nevertheless, none have yet explored the infographic modality, which presents distinctive challenges in layout understanding, dense text regions, and visual-textual interaction—gaps that ViInfographicsVQA is specifically designed to fill.

3.1.3. Counting-related VQA Datasets

Counting within the realm of Visual Question Answering (VQA) presents unique challenges, necessitating models to accurately enumerate objects amidst complex visual scenes. Several datasets have been curated to address this facet. TallyQA [22] stands out as one of the largest open-ended counting datasets, encompassing approximately 287,000 questions across 165,000 images. It distinguishes between simple counting tasks, which require basic object detection, and complex ones that demand reasoning over object relationships and attributes. Similarly, HowMany-QA [23] amalgamates counting-specific queries from VQA 2.0 and Visual Genome [24], emphasizing the need for interpretable counting mechanisms in VQA models. LEAF-QA [17] introduces a comprehensive dataset derived from real-world charts and figures, comprising around 250,000 annotated visualizations and nearly 2 million question-answer pairs. Its focus lies in challenging models to interpret and reason over structured data representations. While these datasets have propelled advancements in counting within VQA, they predominantly center on natural images or specific domains like charts. Consequently, they may not fully encapsulate the intricacies of infographics, which often intertwine dense textual information with complex visual layouts, posing additional hurdles for accurate counting and reasoning.

3.2. VQA Methods

3.2.1. General VQA Methods

Visual Question Answering (VQA) methods generally consist of three main components: visual and textual feature extraction, multimodal fusion, and answer prediction. Early approaches, such as those by Antol et al. [1], used pre-trained convolutional neural networks (for example, VGGNet [25] or ResNet [26]) to extract image features and LSTM-based encoders [27] with word embeddings like GloVe [28] or Word2Vec [29] for question representation. These features were then fused into a joint vector and passed through a classifier to predict answers. Subsequent models, such as the Bottom-up Top-down architecture by Anderson et al. [30], introduced object-level reasoning by using object detectors like Faster R-CNN [31] to extract region-based features, thereby enabling finer-grained attention over image content. Other works explored the use of grid-based features and found comparable performance [32], highlighting the importance of selecting appropriate visual representations. With the rise of attention-based architectures, the focus shifted to improved multimodal fusion strategies, where attention mechanisms—especially multi-hop attention—were used to model complex reasoning across visual and textual modalities. This development culminated in the adoption of co-attention modules and transformer-based architectures such as ViLBERT [33], LXMERT [34], VisualBERT [35], and UNITER [36], which jointly pre-train on large-scale vision-language data to learn aligned multimodal representations. These models demonstrated superior performance across various benchmarks. Additionally, the field expanded to address VQA on document images and scene-text-rich content, with models like M4C [37] and TAP [38] incorporating scene text tokens into the attention mechanism. Transformer variants like LayoutLM [39] and LAMBERT [40] further enriched this paradigm by integrating 2D positional embeddings of text within the visual layout, enabling document-level understanding. More recent architectures, including LayoutLMv2 [41], TILT [42], DocFormer [43], and StrucText [44], extend this direction by leveraging large-scale pretraining objectives tailored for document image understanding, marking a significant step forward in both accuracy and generalization for VQA across diverse visual domains.

3.2.2. Infographic-specific VQA Methods

While general VQA has been widely explored, methods tailored to infographics remain relatively scarce. Early efforts by Bylinskii et al. [45] and Madan et al. [46] focused on generating descriptive tags—textual or visual—from infographic content. Landman [47] later employed a summarization approach that generated captions solely based on recognized text, omitting layout or visual structure. These works, however, mostly used the Visually29K dataset, which is limited in diversity and originates from a single web source. MASSVIS [48] offered a more structured set of infographic designs but was designed for cognitive analysis and heavily skewed toward scientific illustrations, limiting its generalizability. As such, these early models lacked the capability to reason over the complex spatial-textual interactions and diverse semantics typical of real-world infographics.

3.2.3. Counting-related VQA Methods

Beyond infographics-focused VQA, counting-based VQA has also emerged as a distinct sub-task, requiring models to handle numeracy and object quantification beyond standard classification. Early efforts like SoftCount [23] estimated object counts via soft attention over image regions, but often suffered from imprecision [49]. To overcome this, approaches such as Zhang et al.'s counting module [49] and Trott et al.'s Interpretable Reinforcement Learning Counter (IRLC) [23] introduced object-centric and sequential counting strategies, improving both accuracy and interpretability. MoVie [50], a model leveraging modulated convolutions for local reasoning, further advanced counting performance by aligning visual and linguistic features more effectively. At the dataset level, TallyQA [22] was proposed to better evaluate complex counting scenarios involving object relationships and attributes. More recently, as large vision-language models (LVLMs) have demonstrated strong general performance, researchers found that they still struggle with counting, particularly for high object counts. To address this, the LVLM-COUN framework [51] applied a divide-and-conquer method by segmenting the image and aggregating per-region predictions, enhancing performance without additional training. These innovations in counting-based VQA serve as a crucial bridge between general-purpose VQA systems and domain-specific applications such as Vietnamese VQA.

3.2.4. Vietnamese VQA Methods

Vietnamese VQA research has also progressed through several notable developments. Tran et al. [2] introduced a model employing Hierarchical Co-Attention [8], surpassing traditional RNN baselines on the ViVQA dataset. Building upon this, Nguyen-Tran et al. [52] proposed a bidirectional cross-attention architecture to better align visual and textual features. Nguyen et al. [53] further advanced the field with the Parallel Attention Transformer (PAT), effectively integrating linguistic structure and contextual cues. Tran et al. [54] later combined BEiT-3 [55] for multimodal fusion with BARTpho [56] for Vietnamese question encoding, achieving state-of-the-art results on ViVQA. Most recently, Nguyen et al. [7] (2024) introduced a model that integrates BLIP-2 [57] and EfficientNet [58] to capture both global and local image features. By freezing these pre-trained components, they reduced computational costs while maintaining high performance. Their approach achieved a 71.04% accuracy on the ViVQA test set, marking a significant advancement in Vietnamese VQA.

4. ViInfographicsVQA Dataset

The ViInfographicsVQA dataset is specifically designed to address the visual question answering (VQA) task with a focus on counting-based questions. Our dataset targets the challenging and practical problem of counting objects or elements presented in infographic images. These infographics are carefully collected from reputable Vietnamese news sources, ensuring high-quality visual content and contextual relevance. All infographic content is entirely in Vietnamese, which allows us to formulate questions and answers grounded in the Vietnamese language.

By anchoring the dataset in native-language infographics, we aim to bridge visual reasoning with language understanding in the context of Vietnam, while also promoting research in Vietnamese-language VQA tasks. The counting nature of the questions introduces an additional layer of complexity, making this dataset a valuable resource for studying quantitative reasoning in vision-language models.

In this section, we begin by describing the data creation pipeline, including infographic collection, question-answer pair generation, question-type categorization, and preprocessing steps. Then, we present an in-depth analysis of the dataset, including statistical summaries and finally, we will make some comparisons with other existing VQA datasets.

4.1. Data Creation

The data creation pipeline of our VQA dataset is illustrated in Figure 1. The process begins with collecting a large number of infographic images from the web. These images are then organized into structured folders to facilitate subsequent processing. Next, we design a set of rules and annotation guidelines to ensure consistency and clarity in the question-answer (QA) generation process. Based on these guidelines, both textual (scene text-based) and non-textual (purely visual) questions are created. To support this, text information is extracted from images using OCR techniques, while visual features are used for non-text-based understanding. A large language model (LLM) is then employed to assist in generating diverse and high-quality QA pairs following the designed rules. These QA pairs are categorized into Text QA or Non-text QA depending on the type of information they target. Finally, the dataset is validated, cleaned, and split into training, validation, and test sets, ready to be used in VQA benchmarking experiments.

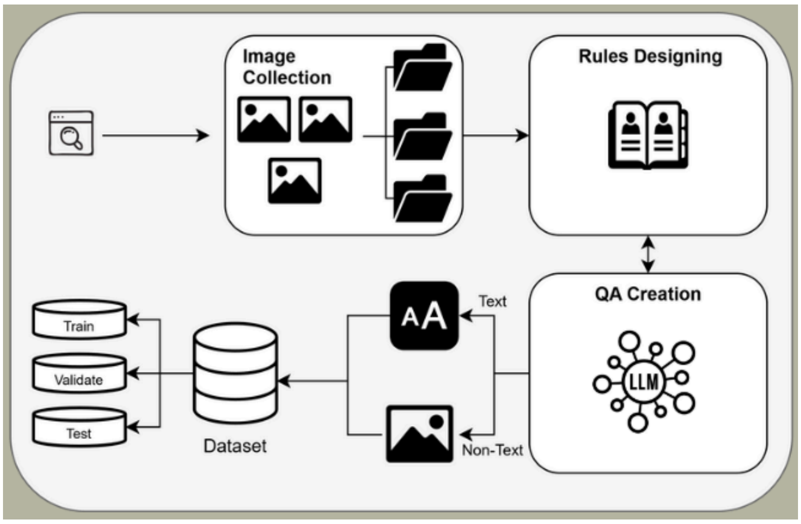


Figure 1: Overall process for the creation of the ViInfographicsVQA dataset.

4.1.1. Infographics Collection

To construct a culturally and linguistically relevant dataset, we curated a collection of infographic images sourced exclusively from Vietnamese news media. In contrast to prior VQA datasets in English, our approach seeks to represent the distinctive cultural, societal, and linguistic characteristics of Vietnam. Infographics found in Vietnamese media often depict data-driven visual storytelling about the country's socio-economic indicators, public health, education, or political events - frequently incorporating Vietnamese text, national symbols, and unique stylistic conventions. This design choice aligns with our goal to generate VQA questions and answers that are deeply rooted in the Vietnamese language and context.

We began by identifying reputable online news outlets in Vietnam that regularly publish infographic content. These news platforms often include dedicated sections for infographics, making it feasible to collect a large volume of high-quality visual data. Using custom-built crawlers, we automatically scraped infographic images from these sections. However, during the collection process, we encountered two major challenges.

- First, several news sites tend to republish infographics originally produced by larger media agencies. This led to a significant number of duplicate images across different sources. To address this, we implemented a robust deduplication process using image hashing techniques to ensure that only unique infographics were retained.

- Second, the scraped content often included irrelevant or noisy images that were embedded alongside the infographics on the same webpages, such as decorative banners, unrelated article thumbnails, or icons. To filter out such noise, we applied a combination of manual review and automated heuristics, such as leveraging OCR to detect the presence of meaningful Vietnamese text and verifying typical infographic layout patterns.

After cleaning and refining the dataset, we successfully compiled a collection of infographic images from 26 different Vietnamese news outlets. The resulting corpus represents a diverse and authentic visual archive of Vietnamese media. The distribution of collected infographics by news source was presented in Figure 2

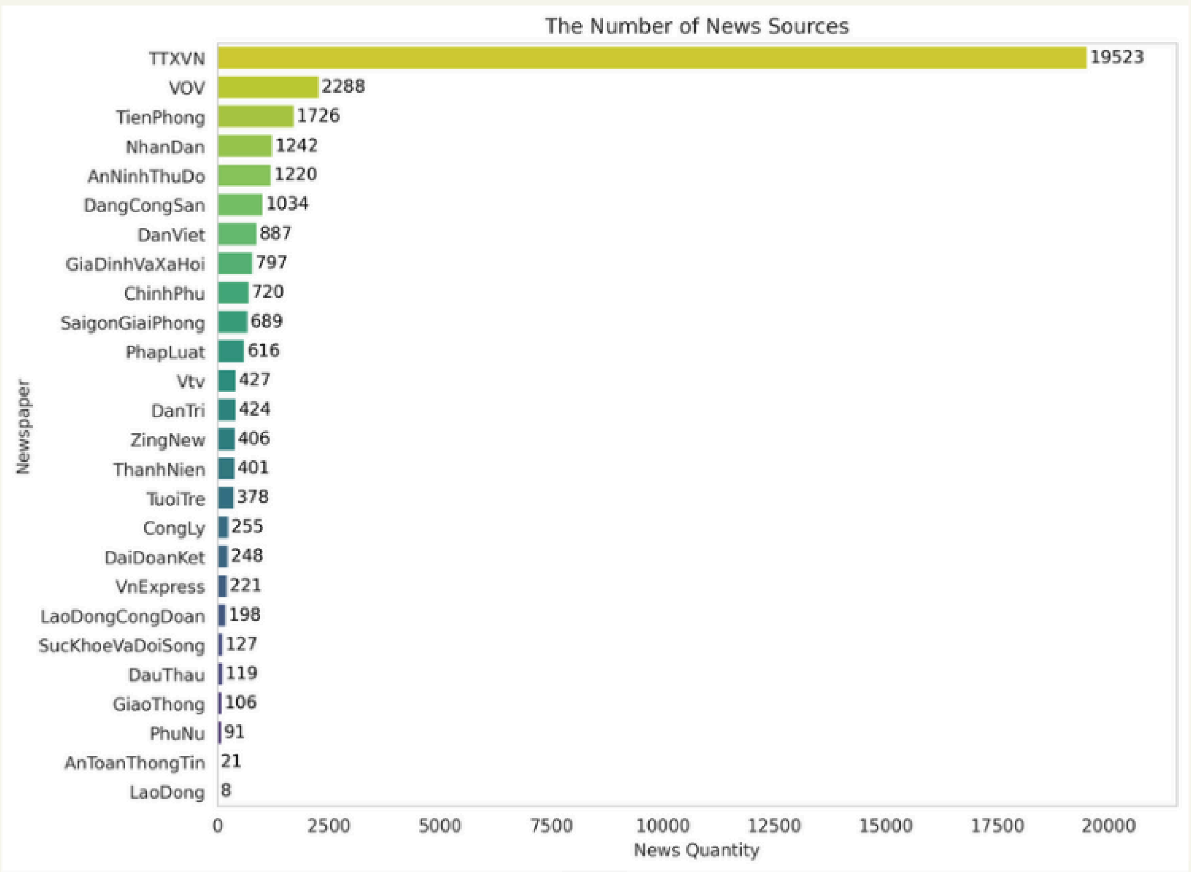


Figure 2: Distribution of collected infographic images across 26 Vietnamese news outlets

4.1.2. Question-Answer Pair Creation

To construct question–answer (QA) pairs for each infographic, we employ a synthesis-based approach using the Gemini-2.0 Flash model instead of relying on human crowdworkers. We chose Gemini-2.0 Flash for its strong multimodal capabilities, particularly in vision-language tasks such as visual question answering (VQA). Among currently available large vision-language models (LVLMs), Gemini-2.0 Flash offers a good balance of accuracy, efficiency, and accessibility. The decision was supported by public benchmark results, including the Open VLM Leaderboard[4], where Gemini-2.0 Flash demonstrates competitive performance across a wide range of multimodal tasks.

To ensure the generated QA pairs meet the desired quality and linguistic diversity, we designed a set of structured guidelines, which are presented in Table 1. These rules are embedded directly into the prompting process used to query Gemini. The generation pipeline is fully automated, and every QA pair is validated post-hoc for consistency and correctness.

[4] *https://huggingface.co/spaces/opencompass/open_vlm_leaderboard*

Criteria	Rules
Number of QAs	• Approximately 5 QA pairs are generated per infographic image.
QA Length	• Both the question and the answer should not exceed 30 words.
Color	• Only the following colors are allowed: black, white, red, orange, yellow, green, blue, sky blue, purple, pink, brown, gray. • Questions must ask about the color of specific objects, not the background or abstract regions. • If the object's color does not clearly fall into one of the allowed categories, color should be omitted.
Question Constraints	• Yes/no and choice-based questions are not permitted. • Questions should not require reasoning beyond the information explicitly shown in the infographic. • Avoid overly vague or general questions that can be answered without viewing the image. • Numerical questions must involve comparison (for example, "more than," "within a range"). • For counting tasks, specify a criterion (for example, "How many cities start with the letter T?").
Answer Constraints	• Answers should be full, grammatically correct sentences. • Each answer must be accompanied by a paragraph-length explanation (max 100 words), justifying the answer with reasoning or reference to visual elements in the infographic. Bullet points are not allowed.

Table 1: Guidelines for creating questions and answers in the ViInfographicsVQA dataset

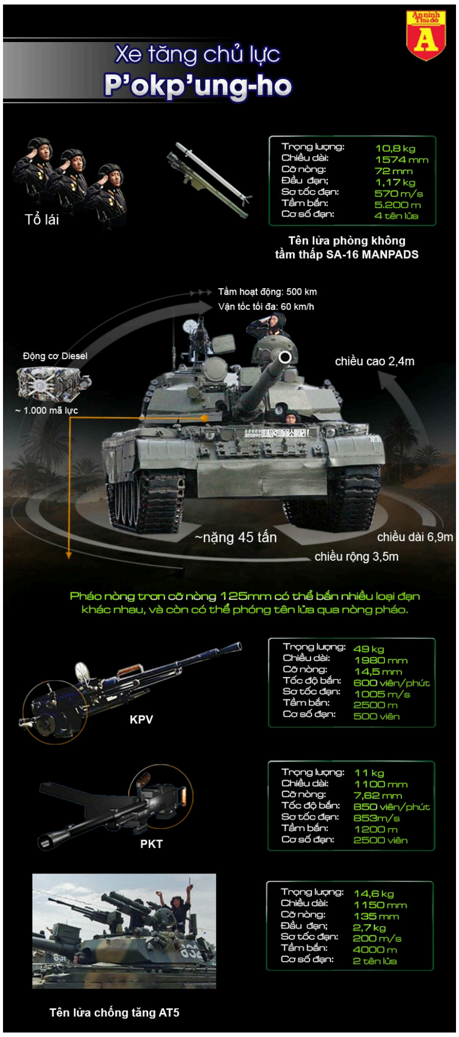
By embedding all these rules into the prompts, we are able to generate diverse and high-quality QA pairs through Gemini in a consistent and scalable manner. This methodology ensures that the dataset remains both linguistically rich and closely aligned with the unique properties of infographic-based visual question answering in Vietnamese.

4.1.3. QA Type Classification

To better analyze and evaluate the role of scene-text in our ViInfographicsVQA dataset, we categorize each question–answer (QA) pair into one of two types: **Text QA** or **Non-text QA**. This classification enables more targeted benchmarking of models' abilities to reason over both visual and textual elements commonly found in infographic-style images.

- **Text QA** involves questions that are grounded in textual elements present within the infographic. This includes questions based on numerical data, textual content, or any embedded scene text. Questions that require extracting information from text to support reasoning about other elements are also classified as Text QA.
- **Non-text QA**, by contrast, includes questions that can be answered without relying on any textual information. These questions typically focus on visual content such as colors, shapes of charts or graphs, positions on a map, or objects like people, animals, trees, and vehicles.

We present typical examples of Text QA and Non-text QA in Figure 3 using an infographic of the P'okp'ung-ho main battle tank. With the question in the Text QA category, it asks about the number of weapons listed with firing ranges greater than 2000 meters. Answering this requires extracting numerical range values from the weapon descriptions in the infographic and comparing them to a threshold, which reflects reasoning over scene text. This example highlights the need to read and reason over embedded text. In contrast, the Non-text QA example rely solely on visual understanding, which asks how many crew members are wearing helmets in the illustration labeled "Tổ lái" (crew). This requires identifying and counting objects in a specific visual region. These examples emphasize the distinction between textual and purely visual information grounding in multimodal question answering.



Text QA

Q: Có bao nhiêu loại vũ khí được liệt kê có tầm bắn lớn hơn 2000 mét?

A: Có 3 loại vũ khí có tầm bắn lớn hơn 2000 mét: Tên lửa phòng không tầm thấp SA-16 MANPADS (5200m), KPV (2500m), và Tên lửa chống tăng AT5 (4000m).

Trans. Q: How many of the listed weapons have a range greater than 2000 meters?

Trans. A: There are three types of weapons with ranges greater than 2000 meters: SA-16 MANPADS low-altitude air defense missiles (5200m), KPV (2500m), and AT5 anti-tank missiles (4000m).

Non-Text QA

Q: Có bao nhiêu người đang đội mũ bảo hiểm trong hình minh họa ‘Tổ lái’ ở góc trên bên trái của infographic?

A: Có 3 người đang đội mũ bảo hiểm trong hình minh họa ‘Tổ lái’.

Trans. Q: How many people are wearing helmets in the ‘Crew’ illustration in the upper left corner of the infographic?

Trans. A: There are three people wearing helmets in the ‘Crew’ illustration.

Figure 3: Typical examples of Non-text QA and Text QA

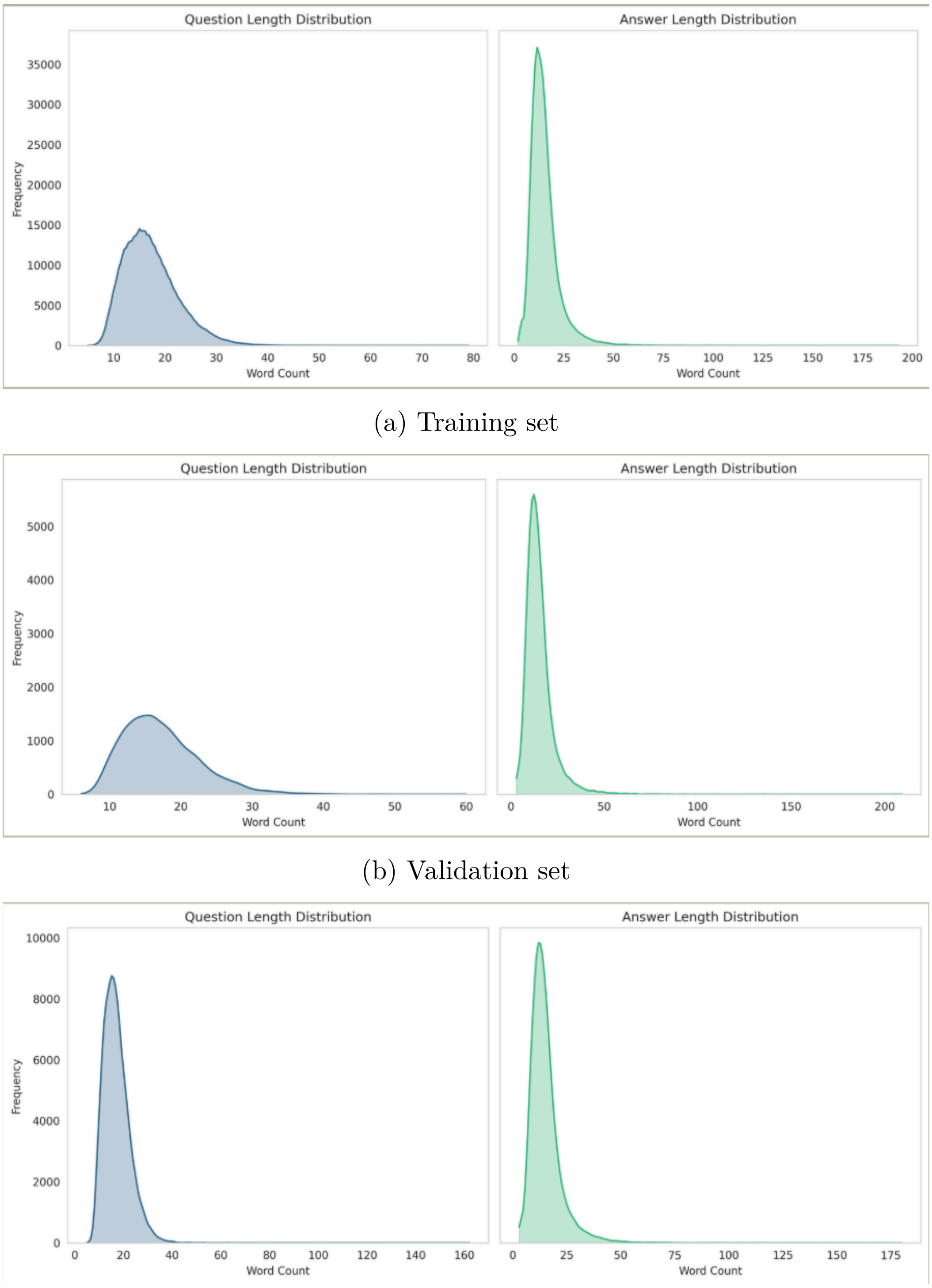
4.1.4. Data Preprocessing

To ensure the overall quality and consistency of the dataset, we performed several statistical analyses to identify and remove outliers, thereby improving the reliability of the data used in subsequent model training and evaluation. These preprocessing steps are essential to reduce noise, avoid training bias, and ensure that the dataset represents a reasonable range of question–answer (QA) pairs typically found in real-world infographics.

In particular, we examined the distribution of question and answer lengths across the train, validation, and test splits. Here, the length of a sentence is defined as the number of words it contains. The detailed statistics and distribution plots are shown in Table 2 and Figure 4. From the analysis, we observed that most questions have lengths ranging between 10 and 35 words, while most answers fall within the range of 10 to 40 words. Additionally, Table 2 reveals some edge cases where the QA pairs are either too short or excessively long. For instance, extremely short examples often occur in counting-type questions such as “Có bao nhiêu xem máy?” (How many motorbikes are there?) with a concise answer like “Có 2.” (There are two.) These are valid but sparse in the overall distribution. On the other hand, QA pairs with abnormally long lengths are typically generated from infographic sections that contain overly detailed descriptions. In such cases, both the question and the answer become unnecessarily verbose, leading to poor generalization. Therefore, these atypical cases were considered outliers and were subsequently removed from the dataset to maintain a more balanced and informative QA corpus.

Dataset	Question			Answer		
	min.	mean	max.	min.	mean	max.
Train Set	5	17.2	79	2	15.2	193
Validate Set	6	17.2	60	3	15.3	209
Test Set	5	17.2	162	3	15.1	180

Table 2: Statistic of questions and answers' length



(a) Training set

(b) Validation set

(c) Testing set

Figure 4: The distribution of questions and answers' length in three subsets

4.2. Data Analysis

	Infographics	Text	Non-Text
Train	23895	60060	37246
Validate	3404	8582	5308
Test	6876	17324	10677
Total	34175	85966	53231

Table 3: Statistic of images and QAs

Our ViInfographicsVQA dataset consists of 34,175 infographic images, from which we generated a total of 139,197 question-answer (QA) pairs in both Text and Non-text formats. The detailed breakdown of QA types and distributions is provided in Table 3. In addition to basic dataset statistics, we conducted an in-depth linguistic analysis to assess the complexity and diversity of ViInfographicsVQA in comparison to existing VQA datasets in both Vietnamese and English, including VQAv2, OCR-VQA [14], TextVQA [13], ViVQA [2], and OpenViVQA [3].

We define the linguistic characteristics of a VQA dataset based on two key factors: the number of semantic dependencies within each question or answer and the height of the corresponding semantic dependency tree. These two measures reflect how syntactically and semantically complex the sentences are. To quantify these aspects, we introduced the Linguistic Complexity Specification (LCS) algorithm [3]. LCS first determines the number of dependencies between tokens in a sentence using a dependency parser appropriate for the sentence's language. It then constructs the corresponding semantic tree and measures its height. A greater number of dependencies and a taller tree imply a more linguistically complex sentence, as illustrated in Figure 5.

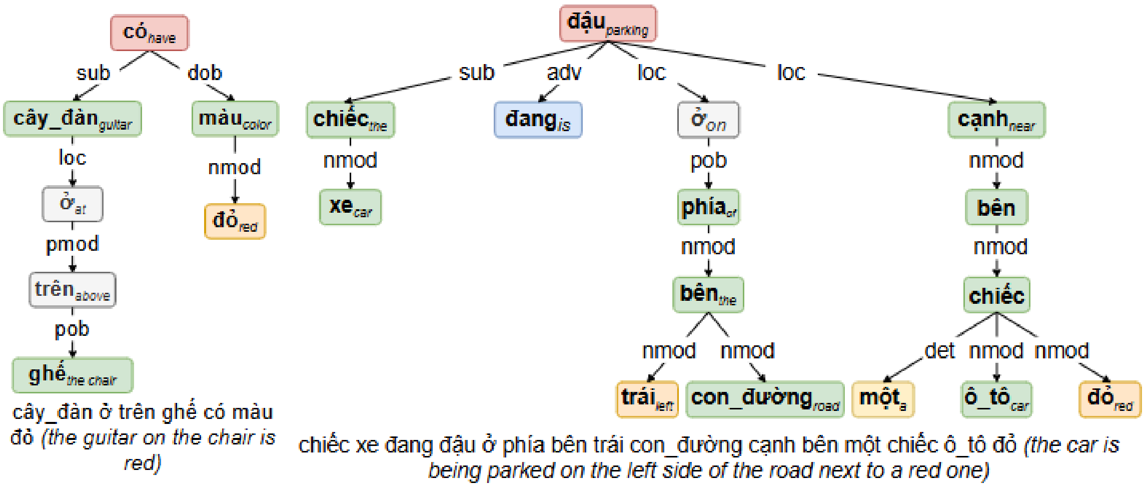


Figure 5: Trees of semantic dependencies between a simple sentence (left) and a complicated sentence (right). The simple sentence has 6 dependencies and its semantic tree has a height of 4 while the complicated one has 14 dependencies and its semantic tree has a height of 4. [3]

Before applying the LCS algorithm, we preprocessed all questions and answers in the VQA datasets. For Vietnamese datasets such as ViVQA [2] and OpenViVQA [3], we used VNCoreNLP [59] to perform word segmentation. This step is crucial due to the multi-syllabic nature of Vietnamese words, for example, "học sinh" (student) or "đại học" (university). Without proper segmentation, these compound words might be mistakenly treated as two separate words, which can lead to incorrect interpretations, for example, "học sinh" being split into "học" and "sinh" might be misunderstood as "studying biology". Thus, accurate word segmentation was performed before any linguistic parsing. After preprocessing, we applied PhoNLP [60] for dependency parsing of Vietnamese sentences to extract semantic relationships and build the semantic trees. For English datasets, preprocessing was simpler; we split the sentences into tokens using space characters and then parsed them using SpaCy[5] to obtain their semantic dependencies and tree structures.

	Dataset	Dependency			Height		
		min.	mean	max.	min.	mean	max.
Question	VQAv2 [1]	2	6.3	26	1	3.3	14
	TextVQA [13]	2	7.5	39	1	3.9	21
	OCR-VQA [14]	4	6.5	10	2	3.6	6
	ViVQA [2]	2	7.3	23	2	5.5	14
	OpenViVQA [3]	2	7.8	27	2	5.2	16
	ViInfographicsVQA (ours)	3	8.3	29	2	3.7	14
Answer	VQAv2 [1]	0	2.8	44	1	1.0	11
	TextVQA [13]	0	1.5	103	1	1.3	40
	OCR-VQA [14]	0	2.8	100	1	1.8	38
	ViVQA [2]	0	0.5	3	1	1.5	3
	OpenViVQA [3]	0	4.8	52	1	4.0	22
	ViInfographicsVQA (ours)	3	8.0	55	2	3.3	17

Table 4: Linguistic comparison on questions and answers among VQA datasets. Note that these results were obtained on train-dev sets. [3]

As presented in Table 4, the ViInfographicsVQA dataset shows notable linguistic complexity in both questions and answers compared to other VQA datasets. Specifically, our dataset achieves the highest mean number of dependencies in questions (8.3) and in answers (8.0), indicating that both question and answer texts tend to be more semantically rich and contextually detailed. Furthermore, the minimum number of dependencies for answers in ViInfographicsVQA is 3, while for most other datasets this value is 0, reflecting the presence of extremely short or simple answers in those datasets. In terms of the height of semantic trees, ViInfographicsVQA maintains moderate but consistent complexity, with an average tree height of 3.7 for questions and 3.3 for answers. This suggests that the syntactic structures in our dataset are deeper than those in typical English VQA datasets like VQAv2 or TextVQA, but still balanced enough to avoid extreme outliers. Although some English datasets (for example, TextVQA and OCR-VQA) have higher maximum heights (up to 40 for answers), their lower average values imply that such deep structures are rare and not representative of the dataset as a whole. These statistics highlight the richer and more consistent linguistic patterns in ViInfographicsVQA, which pose greater challenges for VQA models—particularly in capturing the nuances of Vietnamese language and infographic content. As a result, this dataset can serve as a valuable benchmark for developing and evaluating VQA systems that require deeper semantic understanding and reasoning.

Besides, to prepare vocabulary sets for both questions and answers, we constructed two separate vocabularies through a four-step preprocessing pipeline tailored specifically for Vietnamese. First, we converted all question and answer texts to lowercase to ensure case consistency. Next, since Vietnamese is a language with multi-syllable word formations, we employed the VnCoreNLP toolkit [59] to perform word segmentation, which is crucial to preserve the semantic integrity of tokens (for

[5] *https://spacy.io/*

example, distinguishing “học sinh” as a single unit meaning “student” rather than two unrelated words). After tokenization, we removed all non-alphanumeric characters, except for certain meaningful symbols such as commas (used for numerical formatting) and units like “%”, which carry semantic value in the context of infographics. Finally, we filtered out Vietnamese stopwords using a reference list from the repository [6]. However, rather than adopting this list verbatim, we carefully curated it by retaining words that might be considered stopwords in general but are contextually important for our dataset. This custom filtering helps preserve domain-relevant information in the vocabulary while removing irrelevant or semantically neutral terms.

As a result, we obtained two refined vocabulary sets, one for questions and another for answers. Each optimized for use in downstream VQA model training and evaluation. Figure 6 and Figure 7 visualize the frequency distributions of the resulting vocabularies using word clouds for questions and answers, respectively. In Figure 6, the question vocabulary exhibits strong prominence of interrogative and quantitative terms such as “bao nhiêu” (how many), “số lượng” (quantity), and “loại” (type), indicating that many questions revolve around counting, identifying quantities, or classifying entities. This suggests a significant presence of numeracy-based queries in our dataset. On the other hand, Figure 7 reveals that the answer vocabulary is dominated by terms like “màu” (color), as well as numerals such as 2, 3, 4, 5, reflecting the concise and often categorical nature of the answers. These characteristics highlight the specific linguistic patterns in Vietnamese VQA tasks and affirm the importance of having separate, task-specific vocabularies for questions and answers.

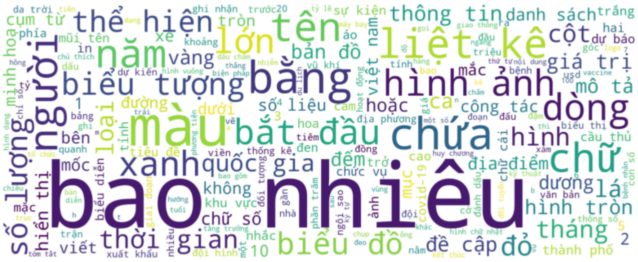


Figure 6: Wordcloud of questions’ vocabulary set

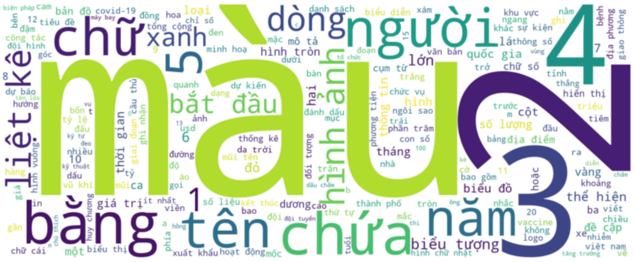


Figure 7: Wordcloud of answers’ vocabulary set

5. Our Proposed Methods

In the context of infographic Visual Question Answering (InfographicVQA), our objective is to develop a model that can accurately answer natural language questions based on the information presented within an infographic image. Given an infographic image I and a question Q, the model must reason over both the visual elements and the embedded textual content (potentially extracted via OCR) to produce the correct answer A. We pursue two main strategies: one that relies solely on the visual features of the infographic, and another that explicitly incorporates and reasons over the textual content within the image. We believe that for images rich in textual information, such as infographics, relying purely on visual cues may lead to a significant loss of critical information, especially in scenarios involving counting tasks or reasoning that depends heavily on text-based elements. This issue becomes even more pressing given that text is often central to understanding the infographic’s message. In this section, we provide a detailed description of the components and processing steps involved in both approaches, which are visually illustrated in Figures 8 and 9. Furthermore, in the Experiments section, we will demonstrate that the second strategy - enhanced with text reasoning - outperforms the purely visual approach across several evaluation metrics.

[6] *https://github.com/stopwords/vietnamese-stopwords*

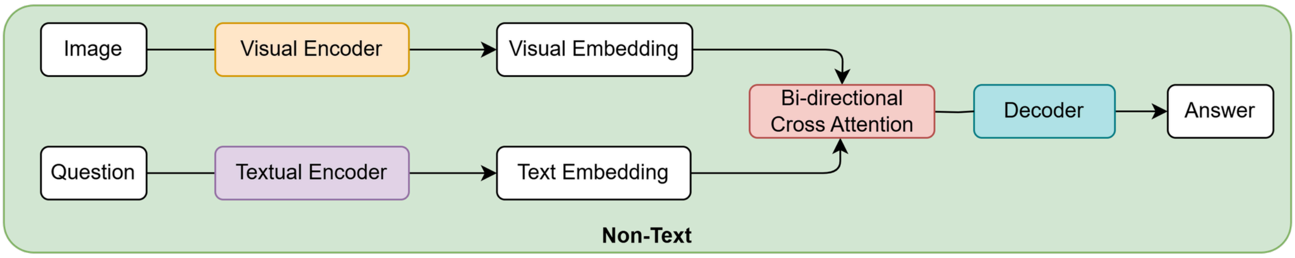


Figure 8: Diagram of a Non-Text-Based Visual Question Answering approach

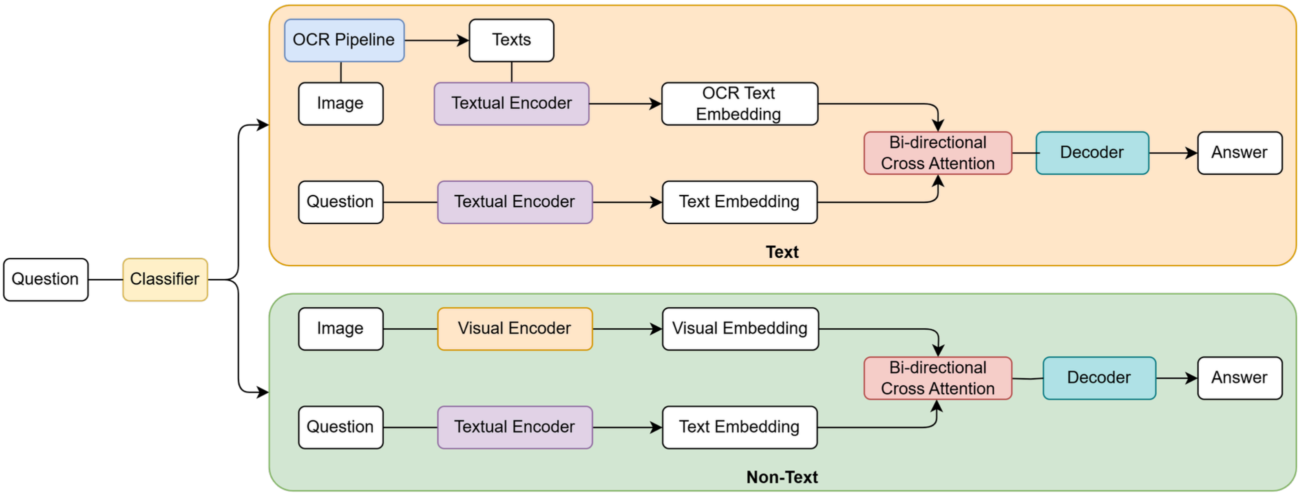


Figure 9: Diagram of Text-Based and Non-Text-Based Visual Question Answering approach

5.1. Classifier

Compared to the first approach, which relies entirely on visual features to answer all types of questions, the second strategy introduces a more structured design by splitting the processing pipeline into two specialized sub-models. One sub-model focuses on textual questions, leveraging the rich textual information often embedded in infographics. The other sub-model handles non-textual questions, which require understanding the visual elements of the image.

To effectively route each question to the appropriate sub-model, we introduce a **Question Classifier** module. The goal of this module is simple yet essential: given an input question Q, determine whether it is a Text Question or a Non-Text Question based on the definitions provided in section 4.1.3. Formally, this can be represented as a function:

$$f_{\text{cls}}(Q) \to \{\text{Text}, \text{Non-text}\} \tag{1}$$

This classification acts as a preprocessing step in the VQA pipeline and operates independently of the main training architecture. It serves as the branching mechanism that decides the appropriate reasoning path for each question.

There are many potential methods for implementing this classifier, ranging from traditional machine learning models to modern transformer-based architectures. However, in order to save development time and focus our efforts on improving the main models dedicated to text-based and visual-based reasoning, while still ensuring high classification accuracy, we opted to use Gemini - a powerful large language model. By leveraging Gemini's strong language understanding capabilities, we achieve reliable and efficient question classification without the need for additional training or fine-tuning.

5.2. Visual Encoder

The Visual Encoder in our framework is designed to extract semantically rich and comprehensive visual features by leveraging the complementary strengths of both convolutional and transformer-based models, which is presented in Figure 10. Inspired by the work of Nguyen et al. [7], who demonstrated the effectiveness of combining local features from CNN-based architectures with global features from Transformer-based models, we adopt a similar hybrid strategy for our visual encoding. Specifically, we integrate EfficientNet [58] for capturing fine-grained local visual details and BLIP-2 [57] for encoding high-level global semantics. This dual-encoder setup enables our model to form a holistic understanding of infographic images, which often contain both intricate visual patterns and broad layout structures.

Unlike Nguyen et al. [7], who chose to freeze the parameters of their visual encoders to reduce computational costs, we do not freeze the parameters in our setup. We argue that infographics are inherently diverse and information-dense, containing both textual and graphical components that vary significantly across samples. Therefore, we believe that fine-tuning both the CNN-based and Transformer-based encoders on our curated infographic dataset is crucial for achieving optimal performance and domain adaptation. This design choice allows our model to better specialize in understanding infographic-specific visual cues, thereby improving its ability to answer visual questions accurately.

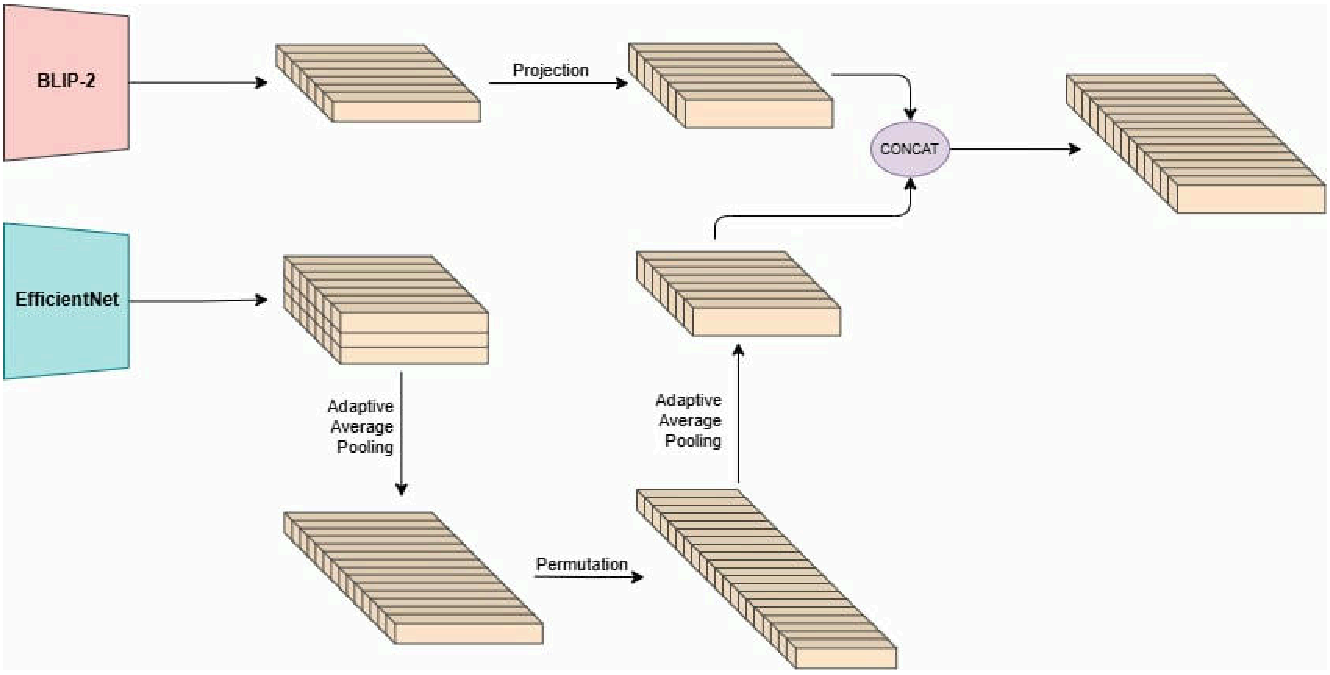


Figure 10: A visual representation of a multimodal feature fusion strategy. Features from BLIP-2 are directly used, while EfficientNet features undergo adaptive average pooling and permutation before being concatenated with the BLIP-2 features.

5.2.1. BLIP-2

BLIP-2 (Bootstrapping Language-Image Pretraining 2) is an advanced model in the field of vision-language pretraining, introduced by Li et al. [57] as a significant follow-up to the original BLIP framework. While the first BLIP model focused on training a vision-language system end-to-end on massive image-text datasets, BLIP-2 takes a more efficient and modular approach to achieve strong performance while reducing computational demands. Instead of training the entire pipeline from scratch, BLIP-2 leverages powerful frozen pre-trained models, including large-scale visual backbones like Vision Transformer (ViT) [61] and language models such as T5 [62], OPT [63], or FlanT5 [64]. These models remain fixed during training, allowing BLIP-2 to drastically cut down on training time and mitigate issues like catastrophic forgetting.

At the core of BLIP-2′s architecture is the Q-Former (Querying Transformer), a lightweight transformer module designed to bridge the gap between visual and textual representations (shown in Figure 11). Q-Former takes as input a set of learnable query embeddings, typically 32 tokens, each with a hidden size of 768—which interact with the image features extracted from the frozen ViT through a cross-attention mechanism. The output is a compact and semantically rich set of features that capture the global content of the image. Compared to the raw output of ViT (for example, a tensor of shape 257 × 1024 for ViT-L/14), the result of the Q-Former is significantly smaller (for example, 32 × 768) while preserving the essential information.

Structurally, the Q-Former consists of standard transformer blocks and operates in two branches: one that processes the visual input using the query tokens, and another that interfaces with textual data. In the BLIP-2 framework, the visual branch is especially important for applications like Visual Question Answering (VQA) systems, where the goal is to extract high-level image representations that can be used to reason about and answer text-based questions.

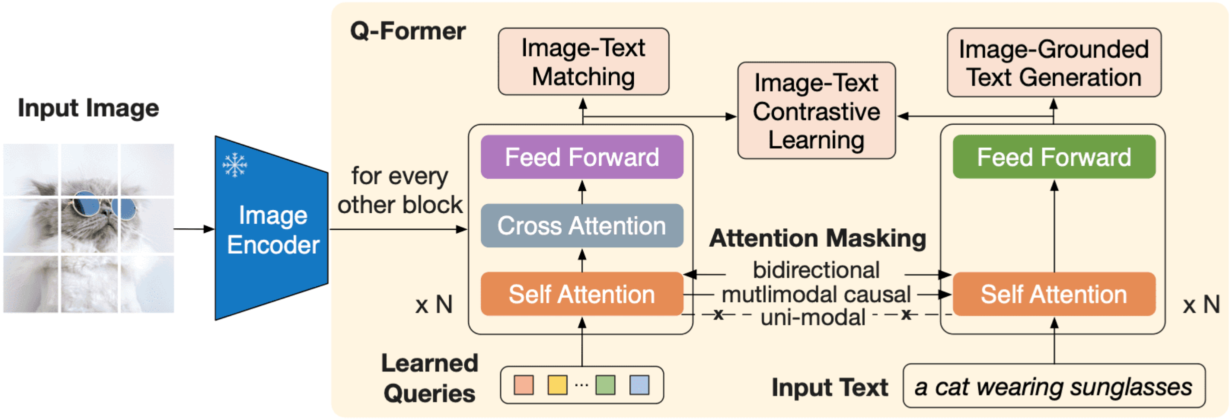


Figure 11: The model architecture of Q-Former in the BLIP-2 framework

The decision to freeze the large pre-trained models in BLIP-2 brings multiple benefits. It not only reduces the computational burden but also ensures the preservation of high-quality features learned during pretraining. Rather than modifying the backbone models, BLIP-2 focuses on adapting the interaction layer, which is the Q-Former, to learn how to effectively query the visual encoder and align the outputs with the language model. As a result, the model is able to generate strong cross-modal representations while maintaining efficiency and scalability.

Ultimately, the central goal of BLIP-2 is to enhance mutual understanding between images and language. The Q-Former enables the system to bridge the modalities by learning to translate and align visual features into a space that is meaningful to language models. This capability empowers BLIP-2 to perform not only traditional tasks like image captioning or VQA, but also more complex generative tasks such as image-to-text generation, visual dialogue, and multimodal reasoning. The final output of the image encoder - represented as a matrix of shape $\mathbb{R}^{32\times768}$, acts as a compact, global feature representation of the image, making it ideal for downstream use in text generation or classification modules.

$$V_G = \text{BLIP-2}(I) \in \mathbb{R}^{32\times768} \tag{2}$$

5.2.2. EfficientNet

EfficientNet is a family of convolutional neural networks (CNNs) introduced by Mingxing Tan and Quoc V. Le [58] in 2019. It was designed to achieve high accuracy while maintaining computational efficiency. The key innovation of EfficientNet lies in its compound scaling method, which uniformly scales the network's depth (number of layers), width (number of channels), and input resolution using a set of fixed scaling coefficients. This approach contrasts with traditional methods that scale these dimensions arbitrarily, often leading to suboptimal performance. By balancing these three dimensions, EfficientNet models achieve better accuracy and efficiency compared to previous CNN architectures .

The architecture of EfficientNet is built upon the Mobile Inverted Bottleneck Convolution (MBConv) blocks [[65], [66]], originally introduced in MobileNetV2. These blocks incorporate depthwise separable convolutions and inverted residual connections, allowing the network to capture complex features while reducing computational cost. Additionally, EfficientNet integrates Squeeze-and-Excitation (SE) modules, which adaptively recalibrate channel-wise feature responses, further enhancing the model's representational power .

EfficientNet models are denoted as B0 through B7, with each successive model scaling up the network's dimensions according to the compound scaling method. For instance, EfficientNet-B7 represents one of the largest models in the family, achieving state-of-the-art accuracy on the ImageNet dataset while being significantly more efficient than previous models. Specifically, EfficientNet-B7 attains 84.4% top-1 accuracy on ImageNet [67], while being 8.4 times smaller and 6.1 times faster on inference compared to the best existing ConvNets at the time of its release .

In practical applications, EfficientNet can be utilized for feature extraction in various computer vision tasks. For example, when processing an input image of dimensions $3 \times 224 \times 224$ (channels × height × width), EfficientNet-B7 can extract feature maps from the final convolutional layer before the classification head. These feature maps typically have dimensions $2560 \times 7 \times 7$, capturing rich local information essential for downstream tasks such as object detection or visual question answering.

$$V_L = \mathrm{EfficientNet}(I) \in \mathbb{R}^{2560\times7\times7} \tag{3}$$

5.2.3. Image Embedding Fusion

In this stage, we integrate the global and local visual features extracted from BLIP-2 and EfficientNet respectively, to construct a unified image representation. Each of these components captures distinct aspects of the visual input - BLIP-2 excels at encoding high-level global semantics, while EfficientNet is adept at extracting fine-grained local patterns. Fusing these complementary representations allows the model to benefit from both broad contextual understanding and detailed visual cues.

However, before this fusion can take place, a dimensional mismatch must be addressed. Specifically, the global visual features V_G from BLIP-2 have a shape of $\mathbb{R}^{32\times768}$, whereas the local visual features V_L from EfficientNet initially have a shape of $\mathbb{R}^{2560\times7\times7}$. To reconcile these formats, we apply a series of transformation steps to reshape V_L into a compatible form.

First, Adaptive Average Pooling is applied to reduce the spatial dimensions from 7×7 to 1×32, yielding:

$$V_L = \mathrm{AdaptiveAvgPool}(V_L) \in \mathbb{R}^{2560\times1\times32} \tag{4}$$

Next, the tensor is permuted to align the desired sequence dimension, resulting in:

$$V_L = \text{Permute}(V_L) \in \mathbb{R}^{32\times1\times2560} \tag{5}$$

Then, a second Adaptive Average Pooling operation is used to compress the channel dimension to 1024:

$$V_L = \text{AdaptiveAvgPool}(V_L) \in \mathbb{R}^{32\times1\times1024} \tag{6}$$

Flattening this gives us the final local feature representation:

$$V_L = \text{Flatten}(V_L) \in \mathbb{R}^{32\times1024} \tag{7}$$

In parallel, the global representation V_G from BLIP-2 is projected through a Linear layer to match the same shape as V_L, transforming it from$R^{32\times768}$ to:

$$V_G = \text{Linear}(V_G) \in \mathbb{R}^{32\times1024} \tag{8}$$

This alignment is essential for the subsequent fusion step. The choice of 1024 as the target dimensionality is deliberate and consistent with the textual encoder used in our second approach. In that design, a BART-based encoder (BARTpho) [56] processes OCR-extracted text with a maximum token length of 1024. Since this text encoder serves a role analogous to the image encoder in our non-text setup, standardizing the feature length across modalities ensures architectural symmetry and facilitates seamless downstream integration.

Finally, we perform the fusion of local and global image embeddings using a function F, which could represent concatenation, attention-based fusion, or another form of learned combination:

$$V = F(V_G, V_L) \in \mathbb{R}^{k\times1024} \tag{9}$$

where k depends on the nature of the fusion function.

5.3. OCR Pipeline

An essential component of our second approach, particularly in handling text-based questions, is the OCR pipeline. The central idea of this approach revolves around encoding textual information extracted from infographics and leveraging it as the primary context to answer questions. Therefore, the quality and structure of OCR-extracted text play a pivotal role in the overall system performance.

To perform Optical Character Recognition (OCR), we employ EasyOCR [68], a library-based tool that has proven through our internal experiments to be more effective for Vietnamese text recognition than alternative solutions such as Pytesseract [69] or PaddleOCR [70]. One of EasyOCR's strengths lies in its robustness and support for Vietnamese, which is crucial given the linguistic diversity in infographic datasets. However, like most OCR libraries, EasyOCR reads text in a fixed spatial sequence, typically from left to right and top to bottom, presented in Figure 12. This default behavior is suboptimal for infographics, where textual content is often grouped or arranged in a non-linear layout, potentially following column-based or clustered patterns. To address this, we incorporate a post-OCR refinement step using Gemini, a large-scale language model. Instead of using Gemini for recognition, we use it to enhance coherence: raw text segments from EasyOCR are passed through Gemini's sequence-to-sequence generation to produce fluent, semantically meaningful sentences that better capture the infographic's content. Due to limitations of the Gemini free-tier API, this step is conducted offline prior to training. The refined text is then pre-encoded and fed into the model, ensuring stable and efficient training while benefiting from Gemini's language capabilities.

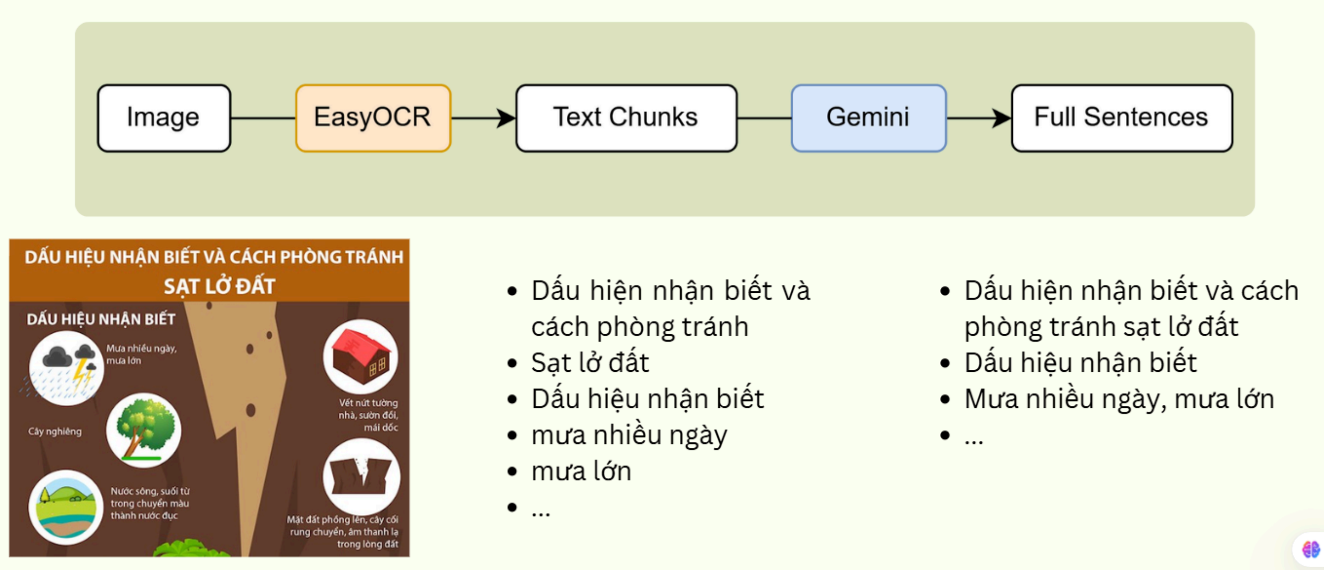


Image
EasyOCR
Text Chunks
Gemini
Full Sentences
DẤU HIỆU NHẬN BIẾT VÀ CÁCH PHÒNG TRÁNH SẠT LỞ ĐẤT
DẤU HIỆU NHẬN BIẾT
Mưa nhiều ngày, mưa lớn
Cây nghiêng
Nước sông, suối từ trong chuyển màu thành nước đục
Vết nứt tường nhà, sườn đồi, mái dốc
Mặt đất phồng lên, cây cối rung chuyển, âm thanh lạ trong lòng đất
• Dấu hiện nhận biết và cách phòng tránh
• Sạt lở đất
• Dấu hiệu nhận biết
• mưa nhiều ngày
• mưa lớn
• ...
• Dấu hiện nhận biết và cách phòng tránh sạt lở đất
• Dấu hiệu nhận biết
• Mưa nhiều ngày, mưa lớn
• ...

Figure 12: Image to Full Sentences Generation using EasyOCR and Gemini

5.4. BARTpho

BARTpho [56] is a sequence-to-sequence pre-trained language model specifically designed for Vietnamese. Inspired by the BART architecture for English, BARTpho is trained as a denoising autoencoder that learns to reconstruct corrupted input sequences, thereby capturing both syntactic and semantic structures of the Vietnamese language. It supports both encoding and generation tasks, making it well-suited for applications such as translation, summarization, and visual question answering. In this section, we introduce our motivation for choosing BARTpho, describe how we use it to encode textual information, and explain how it integrates into our answer generation decoder.

5.4.1. Rationale for BARTpho

Previous research, notably by Tran et al. [54], highlighted the presence of grammatical inconsistencies and translation errors in Vietnamese VQA datasets, even when curated by expert annotators. These inconsistencies suggest that a robust textual encoder must be resilient to noise, ambiguity, and structural variation. While models such as PhoBERT [71] focus primarily on word-level understanding in diverse contexts, our task requires a generative architecture capable of reconstructing coherent output sequences from noisy or fragmented input.

To address this, we adopt BARTpho, which is pre-trained using a two-stage denoising objective: first, applying a noising function to disrupt the input sentence, and second, training the model to reconstruct the original sentence from this corrupted version. This strategy enhances BARTpho's robustness and makes it especially suitable for handling real-world Vietnamese texts, such as OCR outputs from infographics, which are often noisy or imperfectly segmented. Furthermore, because BARTpho follows a full sequence-to-sequence architecture with both an encoder and a decoder, it aligns well with our task of generating free-form answers from paired visual and textual inputs.

BARTpho is released in two variants: BARTpho$_{\text{word}}$, which operates on pre-tokenized word-level input, and BARTpho$_{\text{syllable}}$, which works directly at the syllable level without requiring explicit word segmentation. In this study, we opt for BARTpho$_{\text{syllable}}$ due to its lighter computational footprint while still maintaining reliable accuracy and stability. Its syllable-level granularity is particularly useful for Vietnamese, a language where word boundaries can be ambiguous in OCR-processed text.

5.4.2. Textual Encoder

In both of our approaches, the encoder module of $\text{BARTpho}_{\text{syllable}}$ serves as the textual encoder responsible for transforming the question Q or the OCR information into a dense semantic representation. This operation is expressed as:

$$T = \text{BARTpho}_{\text{syllable}}(Q) \in \mathbb{R}^{n_q \times 1024} \tag{10}$$

where T is the encoded representation of the question and n_q is the number of syllables in the input

In the second model variant (Approach 2), which incorporates both visual features and full OCR-derived text, we face a major practical challenge: the OCR content is typically very long and often exceeds the token length limit of BARTpho. To address this, we divide the OCR sequence into multiple non-overlapping chunks, each consisting of up to 384 syllables. Let $C_1, C_2, ..., C_k$ denote these chunks, where the final chunk C_k may contain fewer than 384 syllables depending on the total input length. Each chunk is independently encoded using BARTpho, and the resulting feature vectors are concatenated to obtain a unified OCR embedding:

$$T_{\text{OCR}} = \text{Concat}\big(\text{BARTpho}_{\text{syllable}}(C_1), ..., \text{BARTpho}_{\text{syllable}}(C_k)\big) \in \mathbb{R}^{n_{\text{ocr}} \times 1024} \tag{11}$$

Here, $n_{\text{ocr}} \leq 384k$ represents the total number of syllables across all chunks after preprocessing. This chunking strategy allows us to retain all textual information from infographics without truncation, ensuring that the model receives a comprehensive representation of the document content.

To accommodate this extended encoding process while minimizing memory usage, we apply a Parameter-Efficient Fine-Tuning (PEFT) technique known as LoRA (Low-Rank Adaptation) [72]. LoRA allows us to fine-tune large pre-trained models like BARTpho without updating the full set of parameters. Instead, it introduces a small number of trainable parameters into specific parts of the model, particularly the attention layers. This approach significantly reduces GPU memory usage and training time while still enabling the model to adapt effectively to our VQA task. By incorporating LoRA, we are able to preserve the generalization capabilities of BARTpho while scaling our model to handle large input sequences within practical resource constraints.

5.4.3. Decoder

In both model variants, the decoder is responsible for generating the final answer based on the fused representation $F \in \mathbb{R}^{(n_v+n_t)\times 1024}$, which is the output of the Bi-directional Cross Attention module. This fused sequence serves as the encoder memory for the $\text{BARTpho}_{\text{syllable}}$ decoder, allowing the model to condition its predictions on the full multimodal context derived from the input question, the image, and, in Approach 2, the OCR content.

During training, we employ a standard teacher forcing approach, where the decoder is trained to predict the next token in the ground-truth answer sequence given all previous tokens. Specifically, given a gold answer sequence $Y = \{y_1, y_2, ..., y_m\}$, we optimize the negative log-likelihood loss across all positions:

$$\mathcal{L}_{\text{gen}} = -\sum_{t=1}^{m} \log P(y_t \mid y_{<t}, F) \tag{12}$$

This objective encourages the decoder to learn token-level dependencies while grounding its predictions in the fused cross-modal context. During training, the decoder input at each step t is the true token y_{t-1}, and the decoder attends over the encoder output F through multi-head cross-attention layers. The use of teacher forcing ensures stable convergence and helps mitigate exposure bias in the early stages of learning. To support efficient fine-tuning on our dataset without modifying the entire decoder, we optionally incorporate LoRA (Low-Rank Adaptation). LoRA injects small trainable low-rank matrices into key attention projections (such as query and value projections) while keeping the rest of the pretrained decoder weights frozen. This significantly reduces the number of trainable parameters and lowers memory consumption during training, making it especially suitable for scenarios with limited computational resources.

At inference time, the decoder operates autoregressively. Starting from the special end-of-sequence token, the decoder predicts one token at a time, each conditioned on all previously generated tokens and the encoder memory. Formally, at each time step t, the model computes:

$$y_t = \arg\max_{w \in V} P(w \mid y_{<t}, F) \tag{13}$$

This process continues until the model emits an end-of-sequence token or reaches a predefined maximum length. In our implementation, we adopt greedy decoding for simplicity and efficiency, though our framework also supports more advanced strategies such as beam search or sampling. Once the final sequence of token IDs is generated, we apply the BARTpho tokenizer to convert the sequence into natural Vietnamese text. This completes the end-to-end generation pipeline, enabling the model to produce fluent, context-aware answers grounded in both visual and textual input.

5.5. Bi-directional Cross Attention

The Bi-directional Cross-Attention Encoder is a multi-layer architecture designed to integrate visual and textual information into a unified representation. In the context of a typical Visual Question Answering (VQA) model, once we obtain the textual features from a given question and the visual features from an image, it becomes crucial to adopt an effective fusion strategy before feeding the combined representation into downstream modules for classification or generation tasks. Inspired by Nguyen-Tran et al. [52] (2022), we adopt the Bi-directional Cross-Attention architecture originally proposed by the authors as an effective mechanism for fusing multi-modal features.

Specifically, the input to this encoder consists of two sequences: a visual feature matrix $V \in \mathbb{R}^{n_v \times 1024}$, where V denotes the number of visual tokens, and a textual feature matrix $T \in \mathbb{R}^{n_t \times 1024}$, where T is the number of text tokens. Each sequence is accompanied by a binary mask $M_v \in \{0,1\}^{n_v}$ and $M_t \in \{0,1\}^{n_t}$ indicating the valid token positions. The Bi-directional Cross-Attention Encoder alternates attention between the two modalities such that visual features attend to textual ones and vice versa, allowing the model to capture fine-grained cross-modal interactions. The final output is a fused sequence of length $V + T$, which can be passed to downstream modules for prediction or generation. The overall structure of this module is described in Figure 13.

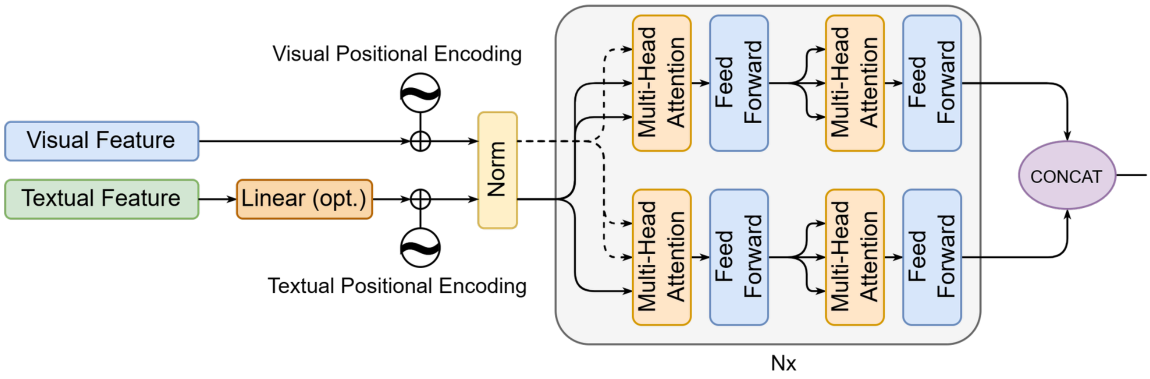


Figure 13: The overall structure of Bi-directional Cross Attention

5.5.1. Positional Embedding and Layer Normalization

Before the attention mechanism can be applied, the model first enriches the raw input features with explicit positional information. Since neither the visual nor textual representations inherently encode sequence order, we introduce learnable positional embeddings and apply layer normalization to ensure stable optimization and effective feature integration across modalities.

Let $V \in \mathbb{R}^{n_v \times 1024}$ be the input visual feature sequence and $T \in \mathbb{R}^{n_t \times 1024}$ be the input textual feature sequence. These feature vectors are obtained from upstream encoders, typically a combination of CNN-based and Transformer-based models for visual inputs and a pretrained language model for text. To inject positional information into each token, we use learned positional embedding matrices $P_v \in \mathbb{R}^{k \times 1024}$ for visual tokens and $P_t \in \mathbb{R}^{k \times 1024}$ for textual tokens, where k is a sufficiently large upper bound on sequence length. In practice, we set $k = 1028$ for all inputs except for the text modality in our second experimental setting, where we use $k = 6000$ to accommodate longer OCR-based token sequences.

We perform an elementwise addition between each input feature and its corresponding positional embedding, followed by layer normalization:

$$\tilde{V} = \text{LayerNorm}(V + P_v[0 : n_v]) \tag{14}$$

$$\tilde{T} = \text{LayerNorm}(T + P_t[0 : n_t]) \tag{15}$$

This ensures that each position-specific token - whether from the image or the text - receives a unique, trainable encoding of its position within the sequence. The layer normalization stabilizes the distribution of the features, which is particularly important given that the model will later combine information across modalities in a tightly coupled attention mechanism. These normalized inputs $\tilde{V}$ and $\tilde{T}$ are then passed to the subsequent bi-directional cross attention encoder layers, already enhanced with both semantic and positional context.

5.5.2. Bi-directional Cross Attention Module

The Bi-directional Cross Attention module is composed of multiple stacked layers, each designed to enable fine-grained interaction and iterative refinement between visual and textual modalities. Each layer comprises two key components: Cross-Modal Attention and Self-Attention Refinement. Both components are constructed using Multi-Head Attention, followed by a Feed Forward Network. Each integrated with residual connections and layer normalization

5.5.2.1. Multi-Head Attention

Each attention block in both the Cross-Modal Attention and the Self-Attention Refinement components uses a Multi-Head Attention mechanism to allow the model to jointly attend to information from different representation subspaces. The architecture of the Multi-Head Attention module is illustrated in Figure 14. It is largely inspired by the original Transformer formulation by Vaswani et al. [73], but we introduce additional dropout and layer normalization to enhance regularization. This component plays a central role in both our Cross-Modal Attention and Self-Attention Refinement, allowing the model to jointly attend to different representation subspaces at different positions.

To begin, the input sequence is projected into three separate matrices: Query Q, Key K and Value V. These projections are linearly transformed into h different subspaces corresponding to the number of attention heads. For each head i, the projections are computed as follows:

$$Q_i = QW_i^Q, \quad K_i = KW_i^K, \quad V_i = VW_i^V \tag{16}$$

where W_i^Q, W_i^K, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ are the learnable parameters, and $d_k = \frac{d_{\text{model}}}{h}$ is the dimensionality of each head. In our models, $d_{\text{model}} = 1024$, $h = 8$, consequently, $d_k = 128$.

Each head then performs Scaled Dot-Product Attention independently to capture contextual dependencies:

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{17}$$

The outputs from all heads are then concatenated along the feature dimension:

$$\text{Concat} = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h) \tag{18}$$

This concatenated representation is passed through a final linear layer to project it back to the original model dimension:

$$\text{Output} = \text{Concat} \cdot W^O \tag{19}$$

where $W^O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ is also a trainable parameter.

To improve generalization, we apply a Dropout layer immediately after this final projection. Finally, we add a residual connection from the original input and apply Layer Normalization:

$$\text{MultiHeadAttention}(x) = \text{LayerNorm}(x + \text{Dropout}(\text{Output})) \tag{20}$$

This complete block enables the model to attend to multiple aspects of the input simultaneously and serves as a core mechanism for learning complex interactions in both unimodal and multimodal settings.

5.5.2.2. Feed Forward Network (FFN)

Following each attention block, a position-wise feed-forward network is applied to each token independently, introducing non-linearity and enhancing representation capacity. The Feed Forward Network (FFN) is a position-wise fully connected transformation that further refines the output of the attention sublayers. The architecture of this component is depicted in Figure 15. While structurally similar to the original Transformer FFN, we make several enhancements to improve its expressiveness and robustness, including the use of GELU activation [74] and two dropout layers.

Given an input vector $x \in \mathbb{R}^{d_{\text{model}}}$, it is first passed through a linear transformation that expands its dimensionality to a larger intermediate space:

$$h_1 = xW_1 + b_1 \tag{21}$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_\pi}$ and d_π is typically set to a multiple of d_{model}, often 2x or 4x. In our models, $d_{\text{model}} = 1024$ and $d_\pi = 2048$.

A GELU activation is then applied to introduce non-linearity in a smooth and differentiable way:

$$h_2 = \text{GELU}(h_1) \tag{22}$$

This activated output is passed through a dropout layer for regularization:

$$h_3 = \text{Dropout}(h_2) \tag{23}$$

Next, the intermediate representation is projected back to the original model dimension:

$$h_4 = h_3 W_2 + b_2 \tag{24}$$

where $W_2 \in \mathbb{R}^{d_\pi \times d_{\text{model}}}$. Another dropout layer is applied to this projection:

$$h_5 = \text{Dropout}(h_4) \tag{25}$$

Finally, a residual connection is added from the original input, followed by a LayerNorm operation:

$$\text{FFN}(x) = \text{LayerNorm}(x + h_5) \tag{26}$$

This module allows each position to independently transform its representation in a non-linear way, strengthening the model's capacity to model complex hierarchical information.

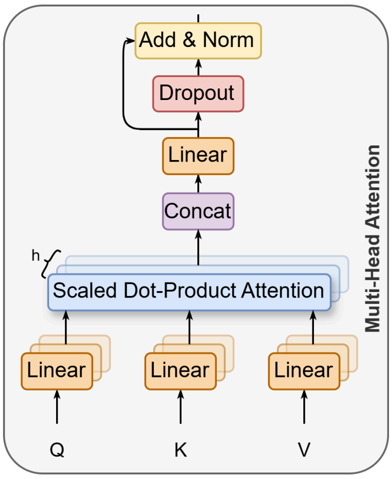


Figure 14: Multi-Head Attention

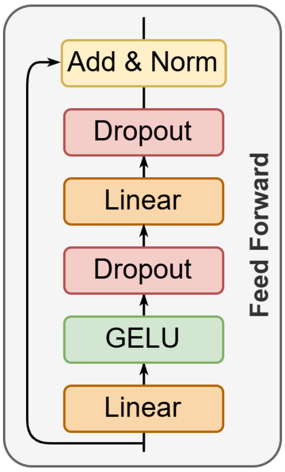


Figure 15: Feed Forward Network (FFN)

5.5.2.3. Cross-Modal Attention

This stage enables a bi-directional information exchange between the visual and textual modalities by employing two asymmetric attention mechanisms applied sequentially. The goal is to allow each modality to condition its representation on the features of the other, thereby achieving a more aligned and contextually enriched fusion.

Let $\tilde{V} \in \mathbb{R}^{n_v \times 1024}$ represent the visual embeddings, where n_v is the number of visual tokens, and $\tilde{T} \in \mathbb{R}^{n_t \times 1024}$ denote the textual embeddings, with n_t being the number of text tokens. Both sets of embeddings are first normalized and enriched with positional encodings, ensuring that the model is aware of spatial and sequential order.

The first step in the cross-modal exchange is the Vision-to-Text Attention, where visual tokens attend to the full sequence of textual tokens to gather semantic information that might aid visual understanding. This is achieved by computing a multi-head attention operation where the queries are the visual embeddings $\tilde{V}$, and both keys and values are the textual embeddings $\tilde{T}$. Formally, this step can be written as:

$$V' = \text{MultiHeadAttention}\left(Q = \tilde{V}, K = \tilde{T}, V = \tilde{T}, \text{mask} = M_t\right) \tag{27}$$

where M_t is a mask that preserves the valid structure of the text sequence, typically by masking out padding tokens. The result V' is then passed through a position-wise feed-forward network to produce the updated visual representation:

$$\hat{V} = \text{FFN}(V') \tag{28}$$

This updated representation $\hat{V}$ encodes visual information augmented with linguistic context, such as object names or spatial relations described in text.

Next, the updated visual features serve as the knowledge base for Text-to-Vision Attention, where textual tokens now attend to the enriched visual representations. The queries are the textual embeddings $\tilde{T}$, and both keys and values are $\tilde{V}$. This step is formulated as:

$$T' = \text{MultiHeadAttention}\big(Q = \tilde{T}, K = \tilde{V}, V = \tilde{V}, \text{mask} = M_v\big) \tag{29}$$

Here, M_v is a visual attention mask that enforces structural constraints such as image padding or region masking. The output T' is then processed through its own feed-forward layer to yield the final updated text representation:

$$\hat{T} = \text{FFN}(T') \tag{30}$$

This sequential two-phase attention design allows for reciprocal alignment between modalities at each layer. The visual tokens first gather high-level linguistic context, and then the text tokens refine their own embeddings using updated visual evidence.

5.5.2.4. Self-Attention Refinement

After the cross-modal attention has aligned and fused the visual and textual representations, each modality independently undergoes a self-attention refinement step. The purpose of this stage is to further contextualize each token within its own modality using intra-modal dependencies, ensuring that the fused information from the other modality is coherently integrated into the local semantic structure.

In the Visual Self-Attention Refinement, the updated visual representation $\hat{V}$ is used as input to a self-attention mechanism where queries, keys, and values are all set to $\hat{V}$, allowing each visual token to attend to all others within the same image context. This operation is defined as:

$$V^{(l)} = \text{MultiHeadAttention}\Big(Q = \hat{V}, K = \hat{V}, V = \hat{V}, \text{mask} = M_v\Big) \tag{31}$$

The output of this attention block is then passed through a feed-forward network:

$$V^{(l)} = \text{FFN}\big(V^{(l)}\big) \tag{32}$$

Here, the residual connection and layer normalization ensure that the model retains both the original and contextually enhanced signals. This refinement allows the visual features to self-organize and propagate the influence of text-aligned features across spatial locations in the image.

Similarly, in the Textual Self-Attention Refinement, the updated textual embeddings $\hat{T}$ are used as the input for another self-attention block. Each token attends to all other tokens in the sentence, allowing global linguistic dependencies (such as subject-verb-object relations) to be reestablished within the fused context. The attention and feed-forward steps are given by:

$$T^{(l)} = \text{MultiHeadAttention}\Big(Q = \hat{T}, K = \hat{T}, V = \hat{T}, \text{mask} = M_t\Big) \tag{33}$$

$$T^{(l)} = \text{FFN}\big(T^{(l)}\big) \tag{34}$$

As with the visual branch, each self-attention layer is followed by residual connections and layer normalization to maintain stability and preserve learned signals. Dropout is applied after both attention and feed-forward operations to prevent overfitting. This self-attention step ensures that each modality retains its internal coherence while incorporating complementary information obtained during cross-modal attention.

5.5.3. Final Fusion

After passing through three successive Bi-directional Cross Attention layers, the model produces two refined sequences: one for the visual modality and one for the textual modality. Specifically, we denote the output of the final Bi-directional Cross Attention block as $V^{(l)} \in \mathbb{R}^{n_v \times 1024}$ for the visual tokens and $T^{(l)} \in \mathbb{R}^{n_t \times 1024}$ for the textual tokens, where n_v and n_t represent the number of visual and textual tokens, respectively, and the embedding dimension remains fixed at 1024.

To integrate information from both modalities into a unified representation, these two sequences are concatenated along the sequence (token) dimension. This operation yields a joint multimodal embedding:

$$F = \left[V^{(l)} \parallel T^{(l)}\right] \in \mathbb{R}^{(n_v+n_t)\times 1024} \tag{35}$$

Here, $\parallel$ denotes the concatenation operation, and the resulting tensor F encapsulates comprehensive cross-modal interactions that have been progressively refined through earlier attention stages. By merging both modalities into a single stream, the fused embedding provides a robust semantic representation that encodes both visual content and linguistic context.

This joint representation F is well-suited for a variety of downstream tasks. Depending on the application, it may be further processed by a lightweight classifier (for example, a feedforward network followed by softmax for classification tasks), or alternatively, passed as input to a decoder (for example, in sequence generation settings such as captioning or visual question answering). The flexible structure of F ensures that it retains modality-specific granularity while enabling joint reasoning across vision and language.

6. Experiments

In this section, we present our experimental framework to evaluate the effectiveness of our proposed multimodal model on the ViInfographicsVQA dataset. We begin by detailing the evaluation metrics used to assess the quality of generated answers, including ROUGE [75], BLEU [76], and BERTScore [77], each capturing different aspects of text generation performance. Next, we describe the experimental settings, including model configurations and training parameters. Following this, we report the results obtained under various setups and compare them across different evaluation metrics. Finally, we provide a discussion on the observed trends, with a particular focus on how the learning rate influences the diversity and richness of the generated responses. Additionally, we examine the accuracy of the Gemini model in classifying question types as either Text or Non-text, highlighting its impact on the answer quality and its implications for modality-aware reasoning.

6.1. Evaluation Metrics

In natural language generation tasks such as summarization, translation, captioning, or general text generation, it is essential to objectively evaluate how closely a model's output aligns with human-written references. Evaluation metrics play a key role in quantifying the quality of generated text by assessing aspects such as structural coherence, semantic meaning, and word-level overlap. In this section, we present three widely adopted metrics: ROUGE, which measures recall-based overlap of words and phrases; BLEU, which emphasizes precision of matched n-grams; and BERTScore, which leverages contextual embeddings to assess semantic similarity between the generated and reference texts.

6.1.1. ROUGE

ROUGE (or Recall-Oriented Understudy for Gisting Evaluation) is a family of metrics for evaluating automatic summarization and text generation. It quantifies how much the reference text is recovered by the model's generated output. Since it is recall-oriented, it cares more about how much of the reference is captured, not necessarily how precise the generated text is.

ROUGE compares n-grams, word sequences, and word overlaps between the generated text and a reference. Key variants are:

Variant	**Description**	**Scenario**
ROUGE-1	Unigram (single word) overlap	Basic word matching
ROUGE-2	Bigram (two-word sequence)	Phrase-level fluency
ROUGE-L	Longest Common Subsequence (LCS)	Sentence structure

The appeal of ROUGE lies in its simplicity, interpretability, and established role in benchmarking summarization systems. By quantifying how many key terms, phrases, or structures from a reference appear in the output, it serves as a reliable proxy for content fidelity. This recall-based perspective is particularly useful in applications like summarization, where the goal is to retain as much essential information as possible. Although ROUGE does not directly account for semantic meaning or synonymy—meaning it may penalize paraphrasing or rewording—it remains a cornerstone metric for comparing system outputs, especially when aligned with extractive or content-preserving tasks.

6.1.2. BLEU

The BLEU (Bilingual Evaluation Understudy) score is a precision-based metric widely adopted for evaluating the quality of text generated by machine translation and natural language generation systems. Its fundamental objective is to assess the degree to which a system-generated sentence overlaps with one or more human-authored reference sentences, focusing on surface-level n-gram matching as a proxy for content fidelity and fluency. In this work, we utilize BLEU-4, the four-gram variant of BLEU, which is the most commonly used instantiation in the literature. BLEU-4 captures lexical overlap at both the word and phrase levels, offering a balance between local coherence (shorter n-grams) and longer-range dependency capture (up to 4-grams). The underlying assumption is that high-quality output will share multi-word sequences with reference translations, especially in domains where terminological precision and phrasing are important.

A critical consideration in BLEU's design is the inclusion of a **brevity penalty** (BP), which corrects for the tendency of precision metrics to favor shorter outputs. Without this component, a system could maximize its score by generating overly concise responses that omit necessary content. The brevity penalty exponentially penalizes outputs shorter than their closest reference, thus encouraging length-appropriate generations.

Formally, BLEU-4 is defined as:

$$\begin{gathered}\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases} \\ \text{BLEU-4} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{4} w_n \log p_n\right)\end{gathered} \tag{36}$$

where c is the length of the candidate, r is the reference length, p_n is the modified precision of n-grams of size n and w_n are weighting coefficients (typically uniform).

While BLEU-4 offers a fast, language-agnostic, and reproducible evaluation signal, it is inherently limited by its insensitivity to synonymy, paraphrase, and deeper semantic structures. Nonetheless, due to its widespread adoption and established benchmarking role, BLEU-4 continues to serve as a key metric in empirical evaluation pipelines.

6.1.3. BERTScore

BERTScore is a semantic evaluation metric that uses contextual embeddings from pre-trained language models like BERT [78] and RoBERTa [79] to assess the similarity between generated and reference texts. Unlike lexical metrics such as ROUGE or BLEU, which rely on exact token or n-gram matches, BERTScore compares tokens in embedding space, capturing deeper semantic relationships. It works by embedding each token in both the candidate and reference sequences, computing pairwise cosine similarities to identify the best semantic alignments. These scores are aggregated into precision, recall, and F1 values at the corpus level, enabling the metric to recognize paraphrasing and synonymy—for example, treating "photo" and "picture" as semantically similar despite token differences.

In practice, BERTScore demonstrates strong empirical correlation with human evaluations, particularly in tasks that require open-ended or abstractive generation, such as summarization or dialogue. It is especially effective in cases where semantic equivalence is more important than syntactic matching, making it well-suited for modern generative language systems. However, BERTScore introduces certain computational and methodological considerations. It requires significantly more resources than traditional metrics due to the need for forward passes through large transformer models. Moreover, its effectiveness depends on the choice of the underlying language model; differences in model architecture, training domain, and language support can meaningfully influence the resulting scores. Therefore, it is critical to report the specific pre-trained model and version used for reproducibility (for example, `bert-base-uncased`, `roberta-large`, or `xlm-roberta-large`).

6.2. Experimental Settings

In our experiments, the hyperparameters presented in Table 5 were utilized. The training process spanned 20 epochs with a batch size of 16. We used 8 heads in the Bi-directional Cross Attention Encoder and 3 encoder layers. The model was optimized using AdamW [80] with a learning rate of 1e-8. An Exponential learning rate scheduler with a decay factor $\gamma = 0.9$ was applied during training. The input image resolution was set to 224 × 224. All experiments were conducted on a computing environment equipped with two NVIDIA T4 GPUs (15GB each) and 29GB of RAM, provided by Kaggle.

Hyperparameters	**Value**
Epochs	20
Bi-directional Cross Attention Encoder Heads	8
Encoder Layers	3
Batch Size	16
Optimizer	AdamW
Learning Rate	1e-8
Learning Rate Scheduler Type	Exponential ($\gamma = 0.9$)

Table 5: Hyperparameters used in training our models

6.3. Experimental Results

To assess the performance of the two proposed approaches, we conducted evaluations on both validation and test sets. Specifically, the questions were categorized into two groups: "Text" and "Non-text" types, allowing us to analyze how each approach performs across different question types. We report a comprehensive set of evaluation metrics, including BLEU-4, ROUGE (1, 2, L), and BERTScore (Precision, Recall, and F1-Score), measured both before and after training. The obtained results are reported in Tables 6, 7, 8, and 9.

		Metrics						
	Model State	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore Precision	BERTScore Recall	BERTScore F1-Score
Approach 1	Before Training	0.0008	0.0111	0.0006	0.0107	0.7038	0.6616	0.6817
	After Training	0.0042	0.0903	0.0056	0.0903	0.7725	0.7360	0.7537
Approach 2	Before Training	0.0012	0.0693	0.0050	0.0617	0.7546	0.7228	0.7371
	After Training	0.0054	0.1117	0.0169	0.1068	0.7912	0.7516	0.7709

Table 6: Evaluation on Text Questions (Validation Set)

		Metrics						
	Model State	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore Precision	BERTScore Recall	BERTScore F1-Score
Approach 1	Before Training	0.0008	0.0113	0.0006	0.0109	0.6981	0.6567	0.6763
	After Training	0.0042	0.0906	0.0059	0.0906	0.7727	0.7361	0.7539
Approach 2	Before Training	0.0012	0.0676	0.0048	0.0600	0.7546	0.7226	0.7370
	After Training	0.0054	0.1112	0.0168	0.1065	0.7914	0.7517	0.7710

Table 7: Evaluation on Text Questions (Test Set)

		Metrics						
	Model State	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore Precision	BERTScore Recall	BERTScore F1-Score
Approach 1	Before Training	0.0009	0.0071	0.0005	0.0069	0.7090	0.6760	0.6918
	After Training	0.0041	0.0828	0.0058	0.0828	0.7752	0.7381	0.7561
Approach 2	Before Training	0.0013	0.1006	0.0060	0.0811	0.7175	0.7441	0.7300
	After Training	0.0059	0.1412	0.0126	0.1078	0.7812	0.7543	0.7675

Table 8: Evaluation on Non-Text Questions (Validation Set)

		Metrics						
	Model State	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore Precision	BERTScore Recall	BERTScore F1-Score
Approach 1	Before Training	0.0009	0.0062	0.0004	0.0060	0.7042	0.6716	0.6872
	After Training	0.0042	0.0836	0.0060	0.0831	0.7756	0.7387	0.7567
Approach 2	Before Training	0.0013	0.1002	0.0060	0.0808	0.7175	0.7439	0.7299
	After Training	0.0060	0.1409	0.0129	0.1079	0.7814	0.7546	0.7678

Table 9: Evaluation on Non-Text Questions (Test Set)

A consistent trend across all tables is that Approach 2, which represents a more advanced model capable of dynamically deciding whether to incorporate textual OCR information alongside visual features, consistently outperforms Approach 1, a simpler baseline that only considers visual cues and ignores embedded text. This distinction becomes especially clear when evaluating questions that require understanding text within images.

For text questions (Tables 6 and 7), Approach 2 shows substantial improvements over Approach 1 across all metrics after training. It yields higher scores in BLEU-4, ROUGE-1, ROUGE-2, and ROUGE-L, reflecting a stronger ability to produce lexically and structurally similar answers to the reference texts. In particular, the BERTScore F1-Score of Approach 2 reaches 0.7709 on the validation set and 0.7710 on the test set, surpassing Approach 1's respective scores of 0.7537 and 0.7539. These results indicate that leveraging OCR information significantly enhances semantic understanding, which is critical for text-intensive questions.

When analyzing non-text questions (Tables 8 and 9), we observe that Approach 2 remains competitive and often superior, even though textual information might not always be essential in these cases. On the test set, for example, Approach 2 achieves the highest ROUGE-1 (0.1409) and BERTScore F1-Score (0.7678) across all configurations. This suggests that the model's dynamic decision-making mechanism does not negatively impact performance on visually grounded questions, in fact, it may still offer subtle advantages in representing answer semantics.

6.4. Discussion

6.4.1. Learning Rate and Answer Variation

During our experiments, we observed that the learning rate plays a relatively significant role in determining the quality and diversity of answers generated by VQA models, particularly in generative settings. Specifically, we found that smaller learning rates tend to produce a greater variety of responses, leading to higher diversity across both the validation and test sets. This finding is especially relevant in open-ended VQA tasks where the ability to generate contextually appropriate and varied answers is crucial.

One possible explanation for this behavior is that larger learning rates may cause the model to converge too quickly to dominant or frequent patterns in the training data, potentially leading to overfitting on high-frequency answers. In contrast, a smaller learning rate encourages more gradual optimization, allowing the model to better explore the solution space and learn subtle variations in answer patterns. As a result, the model is less biased toward overrepresented answers and more capable of generating diverse outputs that align with different input contexts.

To support this claim, we conducted an experiment using our first model variant, which is the model in Figure 8, in which all hyperparameters were held constant as described in Table 5, except for the learning rate. We trained each model configuration for 5 epochs using different learning rates in descending order: 5e-3, 1e-4, 1e-6, and 1e-8. After training, we measured the number of distinct answers generated by each model on both the validation and test sets. The results are reported in Table 10.

Dataset Type	**Learning Rates**			
	5e-3	**1e-4**	**1e-6**	**1e-8**
Validation set	1	1	36	3266
Test set	1	1	52	5483

Table 10: Number of Unique Answers Generated with Different Learning Rates

From the results shown in Table 10, a clear trend emerges: as the learning rate decreases, the number of unique answers generated increases. This pattern reinforces our hypothesis that a smaller learning rate enhances the model's ability to produce more diverse and context-sensitive answers. Consequently, tuning the learning rate is not only essential for stability and convergence but also plays a vital role in improving the expressiveness and adaptability of generative VQA models.

6.4.2. Classifier's Accuracy

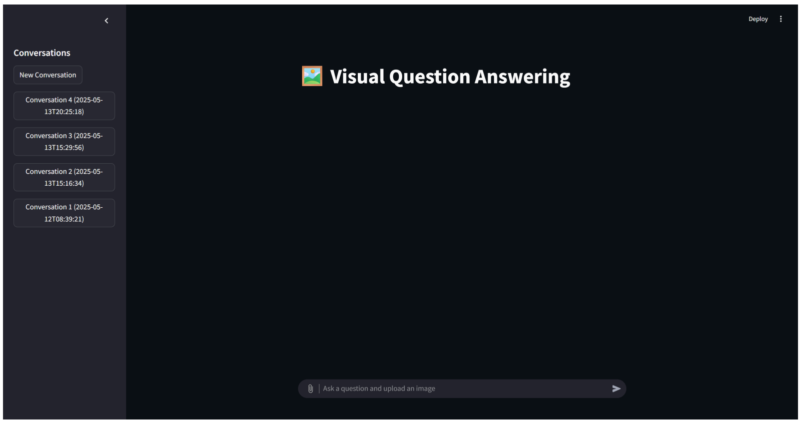
As introduced in the theoretical section describing our model variants, we proposed a component named the Classifier in our second approach. The primary purpose of this module is to determine whether a given question belongs to the Text or Non-text category. To ensure both efficiency and accuracy in this classification step, we employed an external model - Gemini - as previously discussed.

For the experiments, we selected Gemini-2.0-Flash due to its strong performance and fast inference capability. We ran this model on all questions from the train, validation, and test splits of the ViInfographicsVQA dataset. The resulting classification accuracy reached **93.21%**, which is a remarkably high score. This level of accuracy provides strong justification for incorporating Gemini into our second model variant, as it ensures reliable modality distinction with minimal overhead.

7. Our Website Application

7.1. Overview

The visual question answering application is a web-based system designed to interpret user-uploaded images and answer questions about their content, supporting both text-based and non-text-based queries. The system comprises a Streamlit-based frontend for user interaction, a FastAPI backend that processes inference requests using specialized machine learning models, a Python script that facilitates communication between the frontend and backend, and an SQLite database for storing conversation history. Users can ask questions about textual elements (for example, numbers, labels) or visual elements (for example, colors, shapes) in images, and the app intelligently selects the appropriate model to generate a response. The integration of these components ensures a seamless, interactive, and persistent user experience.



Deploy
Conversations
New Conversation
Conversation 4 (2025-05-13T20:25:18)
Conversation 3 (2025-05-13T15:29:56)
Conversation 2 (2025-05-13T15:16:34)
Conversation 1 (2025-05-12T08:39:21)
Visual Question Answering
Ask a question and upload an image

Figure 16: Application's User Interface

7.2. Application Architecture

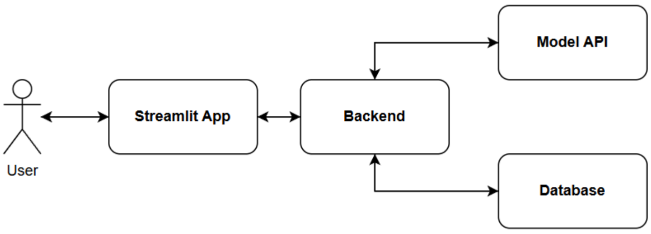


Figure 17: Application architecture

7.3. How the App Works

The application begins with the Streamlit frontend (app.py), which serves as the primary interface for user interaction. Upon accessing the app, users are greeted with a clean interface titled "Visual Question Answering," featuring a sidebar for managing conversations and a main chat area for input and output. In the sidebar, users can start a new conversation or select from a list of previous ones, each displayed with its creation timestamp. Starting a new conversation creates a new entry in the SQLite database, while selecting an existing one loads its message history. In the chat area, users input a question and upload an image (JPEG or PNG), both of which are required for submission. The app validates the input, converts the image to base64, and displays the user's question and image in the chat. This input is then packaged into a conversation object and saved to the database, ensuring all interactions are preserved for future reference.

Once the user submits their question and image, the frontend relies on the api_caller.py script to communicate with the backend. This script extracts the question and base64-encoded image from the conversation object and sends a POST request to the FastAPI server's /infer endpoint, using an API URL defined in environment variables. The simplicity of this script ensures efficient and reliable communication between the frontend and backend, acting as a bridge that passes user inputs and retrieves model-generated responses without introducing unnecessary complexity.

The FastAPI server, defined in fastapi-full.ipynb, is the core of the application's inference capabilities, hosting two specialized models: NonTextModel and TextModel. Upon receiving a request, the server decodes the base64 image and uses Google's Gemini model (gemini-2.5-flash) to classify the question as either "text" (related to textual elements) or "non-text" (related to visual elements). For text-based questions, the server employs EasyOCR to extract text from the image, which Gemini then refines into coherent sentences. The TextModel, built on BARTpho with LoRA fine-tuning and a bidirectional cross-attention mechanism, processes these sentences alongside the question to generate an answer. For non-text questions, the NonTextModel combines local visual features (from EfficientNet) and global visual features (from BLIP2), fused with the question via the same cross-attention mechanism, to produce a response. The server runs on Uvicorn with ngrok tunneling, making it publicly accessible, and returns the generated answer to the API caller.

The response from the backend is relayed back to the Streamlit frontend through the API caller, where it is displayed in the chat interface as an assistant message. This answer is also saved to the SQLite database (database.py), which maintains a structured record of all conversations and messages. The database schema includes a conversations table for tracking conversation IDs and creation timestamps, and a messages table for storing message details, including role (user or assistant), content, optional image data, and timestamps. Functions in database.py handle creating new conversations, retrieving conversation lists, fetching messages for a specific conversation, and adding new messages, ensuring that users can revisit past interactions at any time.

The application's workflow can be summarized as a streamlined data flow: the user inputs a question and image via the Streamlit frontend, which saves the interaction to the database and sends it to the API caller. The API caller forwards the request to the FastAPI backend, which processes the input using the appropriate model and returns an answer. The frontend then displays the answer and stores it in the database, updating the conversation history in real-time. This cycle ensures that users receive prompt, accurate responses while maintaining a persistent record of their interactions.

7.4. Key Aspects of Functionality

The application's ability to handle both text-based and non-text-based questions is a cornerstone of its functionality, achieved through the dual-model architecture hosted on the FastAPI server. The TextModel excels at interpreting questions that rely on textual content extracted from images, such as data points or labels, while the NonTextModel adeptly handles queries about visual attributes like colors or shapes. By leveraging Gemini for question classification and OCR refinement, the system intelligently routes inputs to the appropriate model, ensuring accurate and contextually relevant responses. The use of advanced components like EfficientNet, BLIP2, and BARTpho, combined with a custom bidirectional cross-attention mechanism, enables the models to effectively fuse visual and textual information, delivering robust performance across diverse question types.

Persistence is seamlessly integrated into the application through the SQLite database, which stores every conversation and message, including images, for later retrieval. This allows users to maintain a history of their interactions, switch between conversations, and revisit past questions and answers without loss of context. The database's lightweight design and the straightforward functions in database.py ensure efficient storage and retrieval, enhancing the user experience by making the app feel continuous and reliable.

The Streamlit frontend provides an intuitive and interactive user interface that simplifies the process of uploading images and asking questions. Its chat-based design, complete with real-time message display and image previews, creates a conversational feel, while the sidebar's conversation management features make it easy to organize and navigate multiple sessions. The requirement that users provide both a question and an image enforces clear input expectations, reducing errors and ensuring the models receive the necessary data to generate meaningful responses.

The application's scalability and accessibility are bolstered by the FastAPI backend's use of ngrok tunneling, which exposes the server to the public internet, allowing users to access the VQA system from anywhere. The FastAPI framework ensures high-performance request handling, while the modular design of the backend, with separate models for text and non-text processing, makes it adaptable to future enhancements or model updates. The API caller's role in bridging the frontend and backend is critical, as it abstracts the complexity of HTTP requests into a simple function, ensuring smooth integration across the system's components.

8. Conclusion and Future Work

In this study, we introduced ViInfographicsVQA, the first Vietnamese dataset tailored for Visual Question Answering (VQA) on infographic-style images. The dataset focuses on complex, counting-related questions that require precise reasoning over both visual and textual information. To address this challenge, we proposed two multimodal model architectures, one purely visual, and the other enhanced with OCR and a Vietnamese pretrained language model (BARTpho). Both models were equipped with a Bi-directional Cross Attention module to facilitate deep multimodal fusion. Through extensive experiments and metric-based evaluations using ROUGE, BLEU, and BERTScore, we demonstrated the effectiveness of our approaches and revealed insights into how different configurations affect answer quality and diversity. One of our notable findings is the relationship between learning rate and generation diversity. We observed that smaller learning rates tend to encourage the model to generate a wider variety of responses. This is especially valuable for open-ended generative VQA tasks, where a diverse yet contextually grounded answer set is more desirable than one skewed toward frequent patterns. This insight opens further discussion on how learning dynamics can shape not only model accuracy but also the richness of its output space.

Looking ahead, several directions remain open for future improvement. First, we plan to enhance the ViInfographicsVQA dataset by validating the correctness of all annotated answers. This refinement step will ensure higher data reliability and more robust model evaluation. Second, we intend to improve the model's understanding of counting logic, which is a key aspect of the dataset, by integrating task-specific modules or auxiliary supervision techniques. For instance, incorporating components designed explicitly for counting or leveraging explanation-based learning strategies such as Chain-of-Thought prompting may help the model reason more systematically about quantities rather than relying on superficial patterns. Lastly, we aim to further improve the quality of the generated answers, with a focus on fluency, factuality, and semantic alignment with the visual content. These enhancements will not only boost performance on ViInfographicsVQA but also contribute to broader progress in multimodal learning for low-resource and visually dense domains.

9. References

[1] S. Antol *et al.*, "VQA: Visual Question Answering," *arXiv preprint arXiv:1505.00468*, 2015.

[2] K. Q. Tran, A. T. Nguyen, A. T.-H. Le, and K. V. Nguyen, "ViVQA: Vietnamese Visual Question Answering," *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, 2021.

[3] N. H. Nguyen, D. T. Vo, K. V. Nguyen, and N. L.-T. Nguyen, "OpenViVQA: Task, Dataset, and Multimodal Fusion Models for Visual Question Answering in Vietnamese," *arXiv preprint arXiv:2305.04183*, 2023.

[4] Q. V. Nguyen, Q. D. Tran, H. Q. Pham, and T. K. B. Nguyen, "ViTextVQA: A Large-Scale Visual Question Answering Dataset for Evaluating Vietnamese Text Comprehension in Images," *arXiv preprint arXiv:2404.10652*, 2024.

[5] H. Q. Pham, T. K. B. Nguyen, Q. V. Nguyen, and Q. D. Tran, "ViOCRVQA: Novel Benchmark Dataset and Vision Reader for Visual Question Answering by Understanding Vietnamese Text in Images," *arXiv preprint arXiv:2404.18397*, 2024.

[6] K. V. Tran, H. P. Phan, K. V. Nguyen, and N. L.-T. Nguyen, "ViCLEVR: A Visual Reasoning Dataset and Hybrid Multimodal Fusion Model for Visual Question Answering in Vietnamese," *arXiv preprint arXiv:2310.18046*, 2023.

[7] N. S. Nguyen, V. S. Nguyen, and T. Le, "Advancing Vietnamese Visual Question Answering with Transformer and Convolutional Integration," *arXiv preprint arXiv:2407.21229*, 2024.

[8] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical Question-Image Co-Attention for Visual Question Answering," *arXiv preprint arXiv:1606.00061*, 2016.

[9] M. Mathew, V. Bagal, R. P. Tito, D. Karatzas, E. Valveny, and C. Jawahar, "InfographicVQA," *arXiv preprint arXiv:2104.12756*, 2021.

[10] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi, "Are You Smarter Than a Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] S. Yagcioglu, A. Erdem, E. Erdem, and N. Ikizler-Cinbis, "RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes," *arXiv preprint arXiv:1809.00812*, 2018.

[12] A. F. Biten *et al.*, "Scene Text Visual Question Answering," *arXiv preprint arXiv:1905.13648*, 2019.

[13] A. Singh *et al.*, "Towards VQA Models That Can Read," *arXiv preprint arXiv:1904.08920*, 2019.

[14] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty, "OCR-VQA: Visual Question Answering by Reading Text in Images," *arXiv preprint arXiv:1909.06758*, 2019.

[15] K. Kafle, B. Price, S. Cohen, and C. Kanan, "DVQA: Understanding Data Visualizations via Question Answering," *arXiv preprint arXiv:1801.08163*, 2018.

[16] S. E. Kahou, V. Michalski, A. Atkinson, A. Kadar, A. Trischler, and Y. Bengio, "FigureQA: An Annotated Figure Dataset for Visual Reasoning," *arXiv preprint arXiv:1710.07300*, 2017.

[17] R. Chaudhry, S. Shekhar, U. Gupta, P. Maneriker, P. Bansal, and A. Joshi, "LEAF-QA: Locate, Encode & Attend for Figure Question Answering," *arXiv preprint arXiv:1907.12861*, 2019.

[18] M. Mathew, D. Karatzas, and C. Jawahar, "DocVQA: A Dataset for VQA on Document Images," *arXiv preprint arXiv:2007.00398*, 2020.

[19] R. Tanaka, K. Nishida, and S. Yoshida, "VisualMRC: Machine Reading Comprehension on Document Images," *arXiv preprint arXiv:2101.11272*, 2021.

[20] 5CD-AI, "Viet-Doc-VQA: A Vietnamese Document Visual Question Answering Dataset." 2024.

[21] 5CD-AI, "Viet-OCR-VQA: A Vietnamese OCR Visual Question Answering Dataset." 2024.

[22] M. Acharya, K. Kafle, and C. Kanan, "TallyQA: Answering Complex Counting Questions," *arXiv preprint arXiv:1810.12440*, 2018.

[23] A. Trott, C. Xiong, and R. Socher, "Interpretable Counting for Visual Question Answering," *arXiv preprint arXiv:1712.08697*, 2017.

[24] R. Krishna *et al.*, "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations," *arXiv preprint arXiv:1602.07332*, 2016.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.

[30] P. Anderson *et al.*, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," *arXiv preprint arXiv:1707.07998*, 2018.

[31] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *arXiv preprint arXiv:1506.01497*, 2015.

[32] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller, and X. Chen, “In Defense of Grid Features for Visual Question Answering,” *arXiv preprint arXiv:2001.03615*, 2020.

[33] J. Lu, D. Batra, D. Parikh, and S. Lee, “ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks,” *arXiv preprint arXiv:1908.02265*, 2019.

[34] H. Tan and M. Bansal, “LXMERT: Learning Cross-Modality Encoder Representations from Transformers,” *arXiv preprint arXiv:1908.07490*, 2019.

[35] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, “VisualBERT: A Simple and Performant Baseline for Vision and Language,” *arXiv preprint arXiv:1908.03557*, 2019.

[36] Y.-C. Chen *et al.*, “UNITER: UNiversal Image-TExt Representation Learning,” *arXiv preprint arXiv:1909.11740*, 2020.

[37] R. Hu, A. Singh, T. Darrell, and M. Rohrbach, “Iterative Answer Prediction with Pointer-Augmented Multimodal Transformers for TextVQA,” *arXiv preprint arXiv:1911.06258*, 2020.

[38] Z. Yang *et al.*, “TAP: Text-Aware Pre-training for Text-VQA and Text-Caption,” *arXiv preprint arXiv:2012.04638*, 2020.

[39] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, “LayoutLM: Pre-training of Text and Layout for Document Image Understanding,” *arXiv preprint arXiv:1912.13318*, 2020.

[40] Ł. Garncarek *et al.*, “LAMBERT: Layout-Aware (Language) Modeling for Information Extraction,” *arXiv preprint arXiv:2002.08087*, 2021.

[41] Y. Xu *et al.*, “LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding,” *arXiv preprint arXiv:2012.14740*, 2021.

[42] R. Powalski, Ł. Borchmann, D. Jurkiewicz, T. Dwojak, M. Pietruszka, and G. Pałka, “Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer,” *arXiv preprint arXiv:2102.09550*, 2021.

[43] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, and R. Manmatha, “DocFormer: End-to-End Transformer for Document Understanding,” *arXiv preprint arXiv:2106.11539*, 2021.

[44] Y. Li *et al.*, “StrucTexT: Structured Text Understanding with Multi-Modal Transformers,” *arXiv preprint arXiv:2108.02923*, 2021.

[45] Z. Bylinskii *et al.*, “Understanding Infographics through Textual and Visual Tag Prediction,” *arXiv preprint arXiv:1709.09215*, 2017.

[46] S. Madan *et al.*, “Synthetically Trained Icon Proposals for Parsing and Summarizing Infographics,” *arXiv preprint arXiv:1807.10441*, 2018.

[47] N. Landman, “Towards Abstractive Captioning of Infographics,” Cambridge, MA, 2018.

[48] M. A. Borkin *et al.*, “What Makes a Visualization Memorable?,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2306–2315, 2013, doi: *10.1109/TVCG.2013.234*.

[49] Y. Zhang, J. Hare, and A. Prügel-Bennett, “Learning to Count Objects in Natural Images for Visual Question Answering,” in *International Conference on Learning Representations (ICLR)*, 2018.

[50] D.-K. Nguyen, V. Goswami, and X. Chen, “MoVie: Revisiting Modulated Convolutions for Visual Counting and Beyond,” *arXiv preprint arXiv:2004.11883*, 2020.

[51] M. F. Qharabagh, M. Ghofrani, and K. Fountoulakis, “LVLM-COUNT: Enhancing the Counting Ability of Large Vision-Language Models,” *arXiv preprint arXiv:2412.00686*, 2024.

[52] D.-M. Nguyen-Tran, T. Le, M. L. Nguyen, and H. T. Nguyen, “Bi-directional Cross-Attention Network on Vietnamese Visual Question Answering,” in *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, Manila, Philippines: Association for Computational Linguistics, 2022, pp. 834–841.

[53] N. H. Nguyen and K. V. Nguyen, “PAT: Parallel Attention Transformer for Visual Question Answering in Vietnamese,” *arXiv preprint arXiv:2307.08247*, 2023.

[54] K. V. Tran, K. V. Nguyen, and N. L. T. Nguyen, “BARTPhoBEiT: Pre-trained Sequence-to-Sequence and Image Transformers Models for Vietnamese Visual Question Answering,” *arXiv preprint arXiv:2307.15335*, 2023.

[55] W. Wang *et al.*, “Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks,” *arXiv preprint arXiv:2208.10442*, 2022.

[56] N. L. Tran, D. M. Le, and D. Q. Nguyen, “BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese,” *arXiv preprint arXiv:2109.09701*, 2021.

[57] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models,” *arXiv preprint arXiv:2301.12597*, 2023.

[58] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *arXiv preprint arXiv:1905.11946*, 2019.

[59] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, “VnCoreNLP: A Vietnamese Natural Language Processing Toolkit,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 56–60. doi: *10.18653/v1/N18-5012*.

[60] L. T. Nguyen and D. Q. Nguyen, “PhoNLP: A joint multi-task learning model for Vietnamese part-of-speech tagging, named entity recognition and dependency parsing,” *arXiv preprint arXiv:2101.01476*, 2021.

[61] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[62] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *arXiv preprint arXiv:1910.10683*, 2020.

[63] S. Zhang *et al.*, “OPT: Open Pre-trained Transformer Language Models,” *arXiv preprint arXiv:2205.01068*, 2022.

[64] H. W. Chung *et al.*, “Scaling Instruction-Finetuned Language Models,” *arXiv preprint arXiv:2210.11416*, 2022.

[65] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2018, pp. 4510–4520. doi: *10.1109/CVPR.2018.00474*.

[66] M. Tan *et al.*, “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2820–2828. doi: *10.48550/arXiv.1807.11626*.

[67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. doi: *10.1109/CVPR.2009.5206848*.

[68] JaidedAI, “EasyOCR: Ready-to-use OCR with 80+ Supported Languages.” 2020.

[69] E. Zacharias, M. Teuchler, and B. Bernier, “Image Processing Based Scene-Text Detection and Recognition with Tesseract,” *arXiv preprint arXiv:2004.08079*, 2020, [Online]. Available: *https://arxiv.org/abs/2004.08079*

[70] Y. Du *et al.*, “PP-OCR: A Practical Ultra Lightweight OCR System,” *arXiv preprint arXiv:2009.09941*, 2020, [Online]. Available: *https://arxiv.org/abs/2009.09941*

[71] D. Q. Nguyen and A. T. Nguyen, “PhoBERT: Pre-trained language models for Vietnamese,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1014–1024. doi: *10.18653/v1/2020.findings-emnlp.92*.

[72] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models,” *arXiv preprint arXiv:2106.09685*, 2021.

[73] A. Vaswani *et al.*, “Attention Is All You Need,” *arXiv preprint arXiv:1706.03762*, 2017.

[74] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” *arXiv preprint arXiv:1606.08415*, 2016, [Online]. Available: *https://arxiv.org/abs/1606.08415*

[75] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: *https://aclanthology.org/W04-1013/*

[76] 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*, Philadelphia, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: *https://aclanthology.org/P02-1040.pdf*

[77] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” *arXiv preprint arXiv:1904.09675*, 2019, [Online]. Available: *https://arxiv.org/abs/1904.09675*

[78] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint arXiv:1810.04805*, 2018, [Online]. Available: *https://arxiv.org/abs/1810.04805*

[79] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019, [Online]. Available: *https://arxiv.org/abs/1907.11692*

[80] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” *arXiv preprint arXiv:1711.05101*, 2017, [Online]. Available: *https://arxiv.org/abs/1711.05101*