Performance Analysis of Traditional VQA Models Under Limited Computational Resources

Jihao Gu
Department of Computer Science
University College London
UK, London
jihao.gu.23@ucl.ac.uk

***Abstract*—In real-world applications where computational resources are limited, effectively integrating visual and textual information for Visual Question Answering (VQA) presents significant challenges. This paper investigates the performance of traditional models under computational constraints, focusing on enhancing VQA performance, particularly for numerical and counting questions. We evaluate models based on Bidirectional GRU (BidGRU), GRU, Bidirectional LSTM (BidLSTM), and Convolutional Neural Networks (CNN), analyzing the impact of different vocabulary sizes, fine-tuning strategies, and embedding dimensions. Experimental results show that the BidGRU model with an embedding dimension of 300 and a vocabulary size of 3000 achieves the best overall performance without the computational overhead of larger models. Ablation studies emphasize the importance of attention mechanisms and counting information in handling complex reasoning tasks under resource limitations. Our research provides valuable insights for developing more efficient VQA models suitable for deployment in environments with limited computational capacity.**

***Index Terms*—Visual Question Answering (VQA), BidGRU, Attention Mechanisms, Counting, Resource-Constrained Environments**

I. Introduction

In real-world applications, computational resources are often limited, especially in fields like medicine and industrial automation, where deploying large-scale deep learning models is impractical. Under such constraints, traditional models still play a crucial role in tasks that require both efficiency and accuracy. Visual Question Answering (VQA) is one such task that demands the integration of visual and textual information to answer questions based on images, posing significant challenges when resources are constrained.

This study investigates the performance of traditional models in addressing VQA tasks under limited computational resources. We focus on understanding how various question feature extraction methods and model configurations impact efficiency and accuracy, particularly for numerical and counting questions. By exploring models based on Bidirectional GRU (BidGRU), GRU, Bidirectional LSTM (BidLSTM), and Convolutional Neural Networks (CNN), we aim to identify strategies that optimize performance without the overhead of large-scale models.

Our contributions include a comprehensive analysis of traditional models' adaptability in constrained environments and actionable insights for practitioners working in resource-limited scenarios. Through detailed experimental analysis, we demonstrate that certain configurations, such as the BidGRU model with specific embedding dimensions and vocabulary sizes, can achieve superior performance. These findings provide valuable guidance for developing more efficient VQA models suitable for deployment in environments with limited computational capacity.

II. Related Work

A. Visual Question Answering

Visual Question Answering (VQA) is an interdisciplinary task that combines computer vision and natural language processing. It requires models to answer questions based on visual content. Various approaches have been proposed to improve VQA performance through advanced attention mechanisms and neural architectures.

- **Spatial Memory Network**: Employs a two-hop attention mechanism. The first hop aligns question words with image regions, capturing detailed local evidence. The second hop refines this evidence by considering the entire question embedding, enhancing prediction accuracy [1].
- **BIDAF Model**: Utilizes a bi-directional attention mechanism to create query-aware context representations, capturing interactions between context and query [2].
- **CNN for Text Representation**: Replaces RNNs with CNNs for text representation in VQA, demonstrating superior capability in capturing textual features [3].
- **Structured Attentions**: Models visual attention as a multivariate distribution over a conditional random field (CRF) to better encode relationships between multiple image regions [4].
- **Inverse VQA (iVQA)**: Introduces the inverse VQA task, using question-ranking-based evaluation to diagnose model strengths and weaknesses [5].

B. Image Captioning

Image captioning generates descriptive textual information for images, requiring models to understand and describe visual content accurately.

- **Show, Attend and Tell**: Integrates CNNs with LSTMs and uses attention mechanisms to focus on relevant image regions for accurate captions [6].

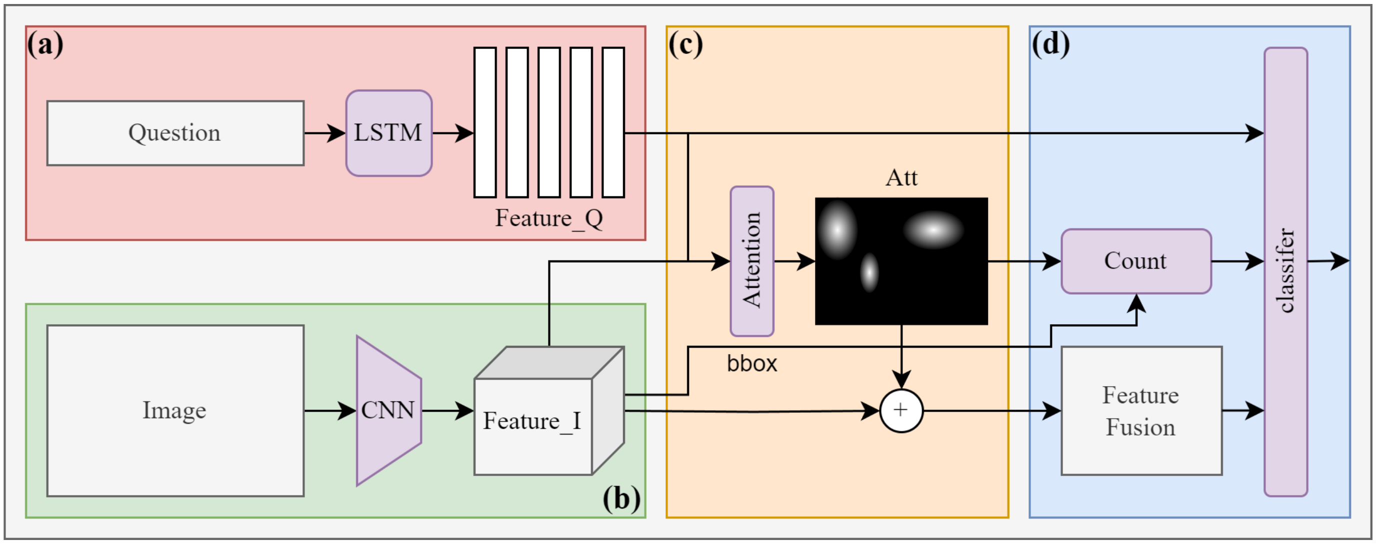

Fig. 1. The VQA model architecture consisting of (a) question feature extraction, (b) image feature extraction, (c) attention mechanism, and (d) feature fusion and classification modules.

- **Self-Critical Sequence Training (SCST)**: Uses the REINFORCE algorithm for reinforcement learning, optimizing the CIDEr metric to reduce exposure bias [7].
- **Meshed-Memory Transformer (M2)**: Employs multi-level encoding and memory-augmented attention for improved caption generation [8].
- **X-Linear Attention Networks (X-LAN)**: Captures second-order interactions using bilinear pooling, enhancing feature representation [9].

C. Multi-Modal

Multimodal research focuses on models that integrate information from multiple modalities, such as text, images, and audio, to perform complex tasks.

- **LayoutLMv2**: Integrates text, layout, and image information through a multi-modal Transformer architecture, enhancing document understanding [10].
- **Cross-Modal Context for Image Captioning**: Combines textual and visual contextual information using CLIP and Visual Genome datasets [11].
- **Transformer-Based Multi-Modal Proposal and Re-Rank**: Uses CLIP and XLM-RoBERTa for image-caption matching through a multi-modal approach [12].

III. Method

In this section, we detail the architecture of our Visual Question Answering (VQA) model, which is designed to effectively integrate visual and textual information for accurate answer prediction. The model consists of four main components: (a) question feature extraction module, (b) image feature extraction module, (c) attention mechanism module, and (d) feature fusion and classification module. 1 illustrates the overall structure of the model.

A. Overview

Our model takes an image V and a question Q as inputs and predicts the correct answer A. The process involves:

- **Question Feature Extraction**: Encoding the question using a BidGRU.
- **Image Feature Extraction**: Processing the image with pre-trained CNNs.
- **Attention Mechanism**: Highlighting relevant image regions based on the question.
- **Counting Module**: Estimating the number of relevant objects for counting questions.
- **Feature Fusion and Classification**: Combining visual, textual, and counting features to predict the answer.

B. Question Feature Extraction

The question feature extraction module aims to convert the variable-length question Q into a fixed-dimensional feature vector that encapsulates its semantic meaning.

1) Embedding Layer: We use an embedding layer to convert each word in the question into a continuous vector representation:

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_T], \quad \mathbf{e}_t \in \mathbb{R}^d \tag{1}$$

where T is the maximum question length, and d is the embedding dimension (set to 300 in our experiments). The embedding layer is initialized randomly and learned during training.

2) Bidirectional GRU Encoder: To capture contextual information, we employ a Bidirectional GRU:

$$\overrightarrow{\mathbf{h}}_t = \mathrm{GRU}(\mathbf{e}_t, \overrightarrow{\mathbf{h}}_{t-1}), \quad \overleftarrow{\mathbf{h}}_t = \mathrm{GRU}(\mathbf{e}_t, \overleftarrow{\mathbf{h}}_{t+1}) \tag{2}$$

The final question representation $\mathbf{q}$ is obtained by concatenating the last hidden states from both directions:

$$\mathbf{q} = [\overrightarrow{\mathbf{h}}_T; \overleftarrow{\mathbf{h}}_1] \in \mathbb{R}^{2h} \tag{3}$$

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT QUESTION FEATURE EXTRACTION METHODS

Model	number (s)	number (p)	count (s)	count (p)	Other	all (s)	all (p)
BidGRU (Dim=300)	**49.52%**	**23.25%**	**57.20%**	**26.80%**	**57.33%**	**65.74%**	**37.83%**
GRU (From Paper)	49.59%	23.37%	57.29%	26.96%	57.18%	65.42%	37.25%
BidGRU (2048-1024)	49.39%	23.09%	57.13%	26.69%	56.88%	65.37%	37.27%
BidLSTM (Dim=300)	47.39%	20.40%	54.73%	23.56%	56.78%	64.92%	36.67%
GRU (300 MFB k=5)	46.27%	19.30%	53.57%	22.29%	55.98%	64.13%	35.96%
CNN (kernel=3+4+5)	41.08%	13.55%	47.32%	15.62%	54.41%	61.86%	32.49%

where h is the hidden size of the GRU.

C. Image Feature Extraction

For image feature extraction, we use pre-trained convolutional neural networks (CNNs) to obtain spatial feature maps from the input image:

$$\mathbf{V} = \text{CNN}(V) \in \mathbb{R}^{C\times H\times W} \tag{4}$$

where C is the number of feature channels, and $H \times W$ is the spatial dimension.

1) Feature Normalization: The visual features are normalized to unit length to ensure numerical stability:

$$\mathbf{V} = \frac{\mathbf{V}}{\|\mathbf{V}\|_2 + \epsilon} \tag{5}$$

where ϵ is a small constant to avoid division by zero.

D. Attention Mechanism

The attention mechanism aligns the question representation with relevant image regions, allowing the model to focus on pertinent visual information.

1) Attention Computation: We compute attention maps using the following steps:

- **Linear Transformations**: Project the visual and question features into a common space:

$$\mathbf{V}' = \text{Conv2d}(\mathbf{V}), \quad \mathbf{q}' = \text{Linear}(\mathbf{q}) \tag{6}$$

- **Feature Tiling**: Tile the question features over the spatial dimensions of the visual features:

$$\mathbf{Q} = \text{Tile}(\mathbf{q}', H, W) \tag{7}$$

- **Fusion**: Combine the visual and question features using a fusion function f:

$$\mathbf{F} = f(\mathbf{V}', \mathbf{Q}) \tag{8}$$

We employ a custom fusion function:

$$f(\mathbf{x}, \mathbf{y}) = -(\mathbf{x} - \mathbf{y})^2 + \text{ReLU}(\mathbf{x} + \mathbf{y}) \tag{9}$$

- **Attention Map Generation**: Apply a convolution to produce the attention maps:

$$\mathbf{A} = \text{Conv2d}(\mathbf{F}) \in \mathbb{R}^{G\times H\times W} \tag{10}$$

where G is the number of glimpses (set to 2).

2) Attention Application: We apply the attention maps to the visual features to obtain attended features:

$$\mathbf{V}_{\text{att}} = \sum_{i=1}^{G} \text{Softmax}(\mathbf{A}_i) \odot \mathbf{V} \tag{11}$$

where $\odot$ denotes element-wise multiplication.

E. Counting Module

To enhance the model's ability to handle counting questions, we incorporate a counting module that estimates the number of objects in the image relevant to the question.

1) Counting Features: We use the first attention map $\mathbf{A}_1$ and bounding box information $\mathbf{b}$ to compute counting features:

$$\mathbf{c} = \text{Counter}(\mathbf{b}, \mathbf{A}_1) \tag{12}$$

The counting module processes the attention weights and spatial information to produce a count feature vector.

F. Feature Fusion and Classification

The final step combines the attended visual features, question features, and counting information to predict the answer.

1) Fusion Mechanism: We fuse the features using a combination of linear transformations and the fusion function:

$$\mathbf{x} = f(\text{Linear}(\mathbf{V}_{\text{att}}), \text{Linear}(\mathbf{q})) \tag{13}$$

We also incorporate the counting features:

$$\mathbf{x} = \mathbf{x} + \text{BatchNorm}(\text{ReLU}(\text{Linear}(\mathbf{c}))) \tag{14}$$

2) Answer Prediction: The fused features are passed through a final linear layer to produce the answer probabilities:

$$\mathbf{A} = \text{Softmax}(\text{Linear}(\mathbf{x})) \tag{15}$$

The model is trained using a cross-entropy loss function.

G. Summary

Our proposed model effectively integrates visual and textual information through a carefully designed architecture that leverages Bidirectional GRUs for question encoding, attention mechanisms for focusing on relevant image regions, and a counting module for handling counting questions. The fusion of features enables the model to perform complex reasoning required for accurate Visual Question Answering.

TABLE II
PERFORMANCE COMPARISON WITH DIFFERENT TOKEN SIZES

Model	Token Size	number (s)	number (p)	count (s)	count (p)	Other	all (s)	all (p)
BidGRU (Dim=300)	2000	49.23%	22.70%	56.87%	26.18%	56.96%	65.46%	37.54%
BidGRU (Dim=300)	**3000**	**49.52%**	**23.25%**	**57.20%**	**26.80%**	**57.33%**	**65.74%**	**37.83%**
BidGRU (Dim=300)	4000	49.16%	22.95%	56.83%	26.51%	57.33%	65.65%	37.61%

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODEL FINE-TUNING STRATEGIES

Model	Fine-Tuning Strategy	number (s)	number (p)	count (s)	count (p)	Other	all (s)	all (p)
BidGRU (Dim=300)	Baseline	**49.52%**	**23.25%**	**57.20%**	**26.80%**	**57.33%**	**65.74%**	**37.83%**
+ Dropout 0.5	Dropout	48.83%	22.36%	56.47%	25.85%	57.18%	65.51%	37.41%
Hidden (2048-1024)	Adjusted Dimensions	49.39%	23.09%	57.13%	26.69%	56.88%	65.37%	37.27%
+ Soft Labels	Soft Labels	49.24%	22.92%	56.92%	26.36%	57.05%	65.25%	37.06%

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present a comprehensive analysis and discussion of the experimental results on the Visual Question Answering (VQA) task. We investigate the effects of different model architectures, vocabulary sizes (token sizes), question feature extraction methods, and model fine-tuning strategies on performance. Besides the overall accuracy (*All*), we pay special attention to fine-grained metrics such as *number (single)*, *number (pair)*, *count (single)*, *count (pair)*, *all (single)*, and *all (pair)*. These metrics help us understand the models' capabilities in handling different types and complexities of questions.

A. Comparison of Question Feature Extraction Methods

We first compare the impact of different question feature extraction methods on model performance, including models based on Bidirectional GRU (BidGRU), GRU, Bidirectional LSTM (BidLSTM), and CNN. The experimental results are shown in Table I.

Analysis: The **BidGRU Embedding Dim 300** model achieved the best performance across all metrics, especially in fine-grained indicators like `number (single)`, `number (pair)`, `count (single)`, and `count (pair)`. This indicates that BidGRU effectively captures the contextual information of questions and has strong understanding and reasoning capabilities for numerical and counting questions. The **GRU (Reproduced from Paper)** model performed slightly lower than the BidGRU model but had a slight advantage in the `number (single)` metric. This might be because the unidirectional GRU has certain benefits when handling simple numerical questions. The **BidLSTM Embedding Dim 300** and **GRU 300 MFB k=5** models showed lower performance across all metrics compared to the BidGRU model, especially in paired questions (`number (pair)`, `count (pair)`), indicating that these models are less capable in handling complex comparison and reasoning questions. The **CNN kernel=3+4+5** model had the lowest overall performance, with significantly lower accuracy in fine-grained metrics, particularly in paired questions. This suggests that CNNs have limitations in capturing sequential information and dealing with complex language structures in VQA tasks.

Conclusion: The BidGRU model, due to its bidirectional structure, can better understand the semantics and contextual information of questions, performing better in handling numerical and counting questions, particularly in complex comparison and reasoning scenarios involving paired questions.

B. Impact of Token Size

We investigated the effect of different vocabulary sizes (token sizes) on model performance. The experimental results are presented in Table II.

Analysis: With a **token size of 3000**, the model achieved the best performance across all metrics, suggesting that an appropriate vocabulary size helps the model fully learn textual features. A **token size of 2000** led to decreased performance, possibly due to the vocabulary being too small, resulting in some key words being excluded and limiting the model's expressive capacity. Increasing the **token size to 4000** did not further improve performance; some metrics slightly decreased, possibly because an overly large vocabulary increases model complexity and training difficulty.

Conclusion: Selecting an appropriate vocabulary size balances information completeness and model complexity; excessively large or small vocabularies can negatively impact model performance.

C. Impact of Fine-Tuning Strategies

We explored the impact of different model fine-tuning strategies on performance, including adding Dropout, adjusting hidden layer dimensions, and using soft labels. The results are shown in Table III.

Analysis: **Adding Dropout** led to slight decreases in performance across all metrics, especially in paired questions (`number (pair)` and `count (pair)`), indicating that a high Dropout rate may suppress the model's ability to learn complex problems. **Adjusting hidden layer dimensions** did not significantly improve performance; some metrics slightly decreased, possibly because the increased model capacity was not effectively utilized, adding to training difficulty. **Using soft labels** resulted in a slight performance decrease, suggesting

TABLE IV
PERFORMANCE COMPARISON OF MODELS WITH EMBEDDING DIMENSION 512

Model	Fine-Tuning Strategy	number (s)	number (p)	count (s)	count (p)	Other	all (s)	all (p)
BidGRU (Dim=512)	+ Dropout 0.5	**49.16%**	**22.87%**	**56.90%**	**26.42%**	**57.21%**	**65.63%**	**37.65%**
Baseline	No Fine-Tuning	48.20%	21.75%	55.71%	25.08%	56.72%	65.23%	37.15%
+ 2 Attention Heads	Multi-Head Attention	40.60%	13.04%	46.77%	15.07%	53.51%	61.52%	32.12%
+ Batch Normalization	Batch Normalization	37.81%	10.18%	43.47%	11.75%	49.34%	57.82%	26.57%

TABLE V
RESULTS OF ABLATION STUDY

Model	Ablation Strategy	number (s)	number (p)	count (s)	count (p)	Other	all (s)	all (p)
Baseline	No Ablation	**49.52%**	**23.25%**	**57.20%**	**26.80%**	**57.33%**	**65.74%**	**37.83%**
Fusion w/o Count	Remove Count Info	44.86%	17.25%	51.74%	19.98%	57.07%	64.93%	36.68%
Fusion w/o Text	Remove Text Info	44.00%	16.77%	50.77%	19.40%	54.23%	62.21%	33.72%
Fusion w/o Attention	Remove Attention	38.81%	11.17%	44.64%	12.90%	47.05%	57.48%	27.00%
Fusion w/o Attn+Count	Remove Attn and Count	39.19%	11.24%	45.13%	12.97%	47.62%	57.89%	27.27%

that precise label information is more beneficial for model learning in this task.

Conclusion: Model fine-tuning strategies should be carefully selected; excessive regularization or blind parameter adjustments may negatively impact model performance.

D. Embedding Dimension Analysis

We further compared models with an embedding dimension of 512 under different fine-tuning strategies. The results are shown in Table IV.

Analysis: The **embedding dimension 512 model with Dropout** achieved performance close to the baseline model with embedding dimension 300 but did not surpass it, indicating that simply increasing the embedding dimension does not lead to significant improvements. The **model with multi-head attention** experienced a notable decrease in performance, especially in paired questions, possibly due to increased model complexity leading to unstable training. The **model with batch normalization (incomplete training)** had the worst performance, possibly due to improper integration of batch normalization or incomplete training.

Conclusion: Model complexity must align with data scale and task difficulty; blindly increasing complexity may be counterproductive.

E. Ablation Study

To evaluate the contribution of each component to the model's performance, we conducted an ablation study. The results are presented in Table V.

Analysis: Removing count or attention mechanisms significantly degrades performance, especially on paired questions, confirming their critical role in complex reasoning tasks.

Conclusion: All components are indispensable for enhancing model performance, especially in handling complex paired questions where these components synergize effectively. The attention mechanism, counting module, and textual features collectively contribute to the model's reasoning capabilities, and their removal significantly impairs performance.

F. Discussion of Fine-Grained Metrics

Analyzing the fine-grained metrics further elucidates the models' performance on different question types and complexities.

- **Single vs. Paired Questions**: Across all models, accuracy in *number (single)* and *count (single)* is higher than in their paired counterparts. This indicates that paired questions require higher reasoning and comparison abilities from the models.
- **Numerical vs. Counting Questions**: The BidGRU model performs better on ‘count‘ questions compared to ‘number‘ questions, possibly because counting involves direct recognition of object quantities in images, whereas numerical questions may require more complex numerical reasoning.
- **Performance Bottlenecks**: Accuracy on paired questions is generally lower, especially after key components are removed, indicating that models still have room for improvement in handling complex reasoning and comparison tasks.

V. CONCLUSION

This study investigated the performance of traditional models in Visual Question Answering (VQA) tasks under limited computational resources, focusing on optimizing efficiency and accuracy. Experimental results demonstrated that the BidGRU model, configured with an embedding dimension of 300 and a vocabulary size of 3000, achieved superior performance, particularly in handling numerical and counting questions. Additionally, attention mechanisms and counting features were identified as critical components for improving the model's capability to address complex reasoning tasks. These findings provide valuable insights for developing efficient VQA models suitable for deployment in resource-constrained environments, such as medical diagnostics and industrial automation.

REFERENCES

[1] Chen, X., et al. (2018). ”Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering.”

[2] Seo, M., et al. (2017). "Bi-Directional Attention Flow for Machine Comprehension."
[3] Noh, H., et al. (2016). "Learning Convolutional Text Representations for Visual Question Answering."
[4] Zhang, L., et al. (2018). "Structured Attentions for Visual Question Answering."
[5] Gordon, D., et al. (2018). "iVQA: Inverse Visual Question Answering."
[6] Xu, K., et al. (2015). "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention."
[7] Rennie, S., et al. (2017). "Self-Critical Sequence Training for Image Captioning."
[8] Cornia, M., et al. (2020). "Meshed-Memory Transformer for Image Captioning."
[9] Pan, Y., et al. (2020). "X-Linear Attention Networks for Image Captioning."
[10] Xu, Y., et al. (2020). "LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding."
[11] Mokady, R., et al. (2021). "Beyond a Pre-Trained Object Detector: Cross-Modal Textual and Visual Context for Image Captioning."
[12] Li, J., et al. (2021). "Transformer-Based Multi-Modal Proposal and Re-Rank for Wikipedia Image-Caption Matching."