Visual Question Answering Using Gated Bidirectional Cross-Attention

Poulami Das Ghosh, Min Chen
Computing and Software Systems, School of STEM
University of Washington Bothell
Bothell, USA
{pdghosh, minchen2}@uw.edu

***Abstract*—Visual Question Answering (VQA) is a multimodal AI task that requires joint reasoning over image and text inputs to answer open-ended questions. This study presents a gated bidirectional Cross-Attention mechanism that integrates BERT-based text embeddings with spatial and global image representations extracted using ResNet-50. The bidirectional Cross-Attention enables dynamic feature interaction between textual and visual modalities, and the gating mechanism filters irrelevant information to reduce computational overhead. Using this approach, we achieved an accuracy of 58% on the VQA v2 validation set, a 10% improvement over the hierarchical co-attention-based baseline work. Furthermore, the system is deployed as a user-friendly and intuitive web application that supports both voice and text input/output for enhanced usability.**

Keywords—cross-attention, gating mechanism, multimodal deep learning, visual question answering

I. Introduction

Visual Question Answering (VQA) is a multimodal artificial intelligence (AI) task that specializes in generating natural-language answers to questions about an image [1]. Questions posed to a VQA system can vary in complexity—from binary queries such as "Is there a cat in the image?" to complex reasoning tasks involving spatial or temporal relationships between objects. Advancements in VQA can provide tangible benefits in real-world applications including accessibility technologies for visually impaired users. In addition, VQA technologies have potential applications in domains such as autonomous driving, assistive robotics, and intelligent surveillance.

VQA requires the integration of computer vision and natural language understanding to perform reasoning over diverse visual and textual inputs. It is a challenging topic because it is difficult to align heterogeneous visual and linguistic information given to the inherent ambiguity and open-ended nature of natural language and the variability of image content. In addition, large-scale generative VQA models often require substantial computational resources, making them impractical for cost-sensitive or real-time applications.

This study proposes a lightweight VQA model that balances performance, efficiency, and user accessibility. It uses a gated bidirectional cross-attention mechanism to integrate linguistic and visual features efficiently. Textual features are derived from BERT embeddings, while visual representations are obtained from both global and spatial feature extractors based on ResNet-50. The bidirectional cross-attention module enables dynamic two-way interaction between visual and textual modalities, and the gating function filters redundant information, improving both interpretability and computational efficiency.

II. Related Work

A. Benchmark Datasets

The development of large-scale, diverse datasets has been instrumental in advancing VQA research. Early datasets such as DAQUAR [2] used indoor scenes with manually annotated question–answer (QA) pairs. However, its small size (1,449 images and ~12K QA pairs) limited generalization. The COCO-QA dataset [3] expanded this by introducing 123K images and automatically generated questions, though its reliance on single-word answers and NLP-based question generation often led to grammatical inconsistencies [4]. Subsequent datasets improved linguistic and visual diversity. VQA v1 [4] utilized more than 200K MS-COCO images and 50K abstract cartoons paired with over 760K questions and approximately 10 million answer annotationsbut suffered from answer bias and limited question types. VQA v2 [5] introduced complementary image pairs for the same question, balancing "yes/no," "number," and "other" categories. With its scale, balance, and public evaluation support via the VQA test-standard server [6], VQA v2 has become the benchmark for most current VQA research and is adopted in our study.

B. Model Architectures and Techniques

VQA models are broadly categorized as generative or discriminative. Generative models treat VQA as a sequence generation task to enable free-form responses [7]. The PALI-X model [8] exemplifies this trend, achieving high accuracy on VQA v2 with 55B parameters. However, such generative models demand extremely high computational cost, limiting its practicality for resource-constrained environments.

Discriminative models, in contrast, frame VQA as a classification task with a fixed answer vocabulary. Early approaches [9,10] employed multimodal fusion of CNN-based visual features and RNN-based text encoders. Anderson et al. [1] achieved notable performance with a bottom-up and top-down attention framework that combined Faster R-CNN for region-level visual features and LSTMs for question encoding. However, this two-stage extraction was computationally intensive.

979-8-3315-7111-5/26/$31.00 ©2026 IEEE

Recent works have introduced cross-modal attention to enhance feature interactions. For example, a Bidirectional Cross-Attention Network (BiCAN) was proposed in [10] to allow image and text features to attend to each other while cross-modal gating was integrated in [9] to filter irrelevant relationships. Studies show that attention-based VQA models such as hierarchical co-attention [11] improved joint reasoning by simultaneously attending to both text and visual features at multiple granularities. Therefore, the hierarchical co-attention architecture proposed in [12] is used as the baseline for this study. Bidirectional cross-attention and gating are used to further improve reasoning and reduce redundancy in multimodal fusion.

III. Methodology

A. Data Preprocessing

Effective preprocessing is essential for preparing inputs for machine learning models, particularly in multimodal tasks such as VQA. In this work, we applied a set of preprocessing operations to both images and questions to ensure consistency, enhance generalization, and improve model performance.

a) Image Preprocessing: Image preprocessing transforms raw inputs into a standardized format suitable for ResNet-50 feature extraction. Unlike the baseline in [12] that relied solely on resizing, our pipeline incorporates additional augmentation and normalization steps to improve robustness as follows.

1. RGB Conversion and Resizing: All images were converted to 3-channel RGB and resized to 224×224 pixels, the recommended ResNet-50 input size [13], for dimensional consistency and efficient batch processing.
2. Data Augmentation: To improve generalization, the training pipeline applies *Random Horizontal Flip* (p = 0.5) and *Color Jitter*, which perturb brightness, contrast, saturation, and hue. These augmentations expose the model to viewpoint and illumination variations, reducing positional and environmental bias.
3. Normalization: Pixel values were standardized by channel-wise mean subtraction and variance scaling, which stabilizes training and improves convergence.
4. Tensor Conversion: Processed images were converted to tensors compatible with the ResNet-50 feature extractor.

b) Question Preprocessing: Question preprocessing converts raw natural-language queries into model-readable embeddings. Instead of manual text normalization, we employ the BERT tokenizer [14], which performs tokenization, lower-casing, and subword segmentation, producing contextualized token indices aligned with the BERT encoder. This ensures consistent and semantically rich textual representations for the downstream cross-attention modules.

B. Feature Extraction

VQA requires integrating visual and textual information; therefore, robust feature extraction from both modalities is required. We use a dual-feature ResNet-50 image extractor to obtain spatial and global visual representations, and a BERT-based text encoder to generate contextualized question embeddings.

*a) Image Feature Extractor:*To capture complementary visual information, we use a ResNet-50 backbone pretrained on ImageNet to extract spatial and global image features. Spatial features retain localized patterns critical for fine-grained reasoning [15], while global features encode overall scene context [13]. Let the input image tensor I be $I \in R^{B \times 3 \times 224 \times 224}$. Here, B is the input batch size, and the image is converted to 'RGB' 3-channels and resized to 224×224. We remove ResNet-50's Adaptive Average Pooling and fully connected layers to preserve spatial structure. As I passes through the modified backbone, a feature map X is created as $X \in R^{\{B \times 2048 \times 7 \times 7\}}$, where B represents the batch size, 2,048 is the number of channels in ResNet-50, and 7 x 7 is the declared spatial dimension size.

1. Spatial Feature Extraction: Spatial feature extraction begins with a 1×1 convolution to reduce the channel dimension from 2048 to 768 as denoted in (1).

$$P = W_p \cdot X + b_p \tag{1}$$

Here, P represents projection output, W_p and b_p represent weight of projection and bias respectively [16]. This helps to generate compact dimensions and reduce computation overhead [15]. Additionally, this dimensionality reduction aligns the feature size with BERT's hidden dimensions for subsequent processing.

Next, ReLU activation function and Batch Normalization function are applied to introduce non-linearity to the projection and to stabilize and normalize each channel [17]. A dropout layer (6) is then introduced to prevent overfitting by randomly deactivating neurons in the network [18]. The dropout is for training phase only. Lastly, the resulting tensor $D \in R^{768 \times 7 \times 7}$ is flattened and combined with positional embeddings to yield the spatial feature sequence Z as in (2)

$$Z = Flatten(D) + E_{pos} \in R^{B \times 768 \times 49} \tag{2}$$

2. Global Feature Extraction: To obtain the fully refined global image representations, the feature map X is first processed by an adaptive average pooling layer that averages the value of each feature channel across all spatial positions as shown in (3) [13].

$$G_c = \frac{1}{7 \times 7} \sum_{i=1}^{7} \sum_{j=1}^{7} F_{c,i,j} \tag{3}$$

Here, $F \in R^{B \times 2048 \times 1 \times 1}$, 7×7 represent the spatial grid dimension, and c indicates 1 to 2,048 channels. The resulting tensor G_c is then flatten and passed through a fully connected layer that reduce its dimension to 768 to aligns with BERT's hidden dimension. Non-linearity is introduced to the network to enable the model to learn complex semantics using ReLU activation [19]. A regularization technique called dropout is

also introduced to randomly deactivate the neuron of the network during training and prevent overfitting [18] and resulting in a tensor D. Finally, layer Normalization is applied to stabilize and standardize the feature distribution [16], which produces the final global embedding F as defined in (4).

$$F = \frac{D-\mu}{\sqrt{\sigma^2+\epsilon}}\gamma + \beta \ \in R^{B\times 768} \quad (4)$$

Here, μ is the arithmetic mean and σ is the standard deviation for all of its features in each sample, γ, β are the learnable parameters and, ε is a small constant added for numeric stability to avoid division by zero.

b) Question Feature Extractor: We generate text features using BERT (bert-base-uncased) to obtain contextual token embeddings suitable for multimodal fusion. Let the input question q with l tokens be: $q = [w_1, w_2, \dots, w_l]$, BERT processes the tokenized sequence to generates contextualized embeddings $H = [h_1, h_2, \dots, h_l]$ in (5).

$$H = BERT(q; \theta_{BERT}) \quad (5)$$

Here, θ_{BERT} represents the parameters of the pre-trained BERT model and each $h_i \in R^{d_{BERT}}$ denotes the contextual feature vector corresponding to the i^{th} token. These dynamic embeddings encode semantic and contextual relationships necessary for downstream gated bidirectional cross-attention.

C. Model Design

We propose a VQA architecture that extends the Bidirectional Cross-Attention framework [10] by integrating the adaptive gating mechanism introduced in [9]. Unlike many lightweight VQA models that rely on simple concatenation of image and text embeddings [3, 20], our design enables bidirectional attention between modalities and selectively filters irrelevant information, improving multimodal alignment for open-ended question answering.

a) Model Architecture: The model consists of three primary modules: Attention Layer, Fusion Layer, and Classifier Layer. Image features (spatial and global) and BERT-derived question embeddings serve as inputs.

1) *Attention Layer:* Spatial features $Z \in R^{B\times 768\times 49}$, and global features $F \in R^{B\times 768}$ are first integrated into a compact combined image feature vector C. It, along with the question text embeddings H, is then passed to two parallel multi-headed cross-attention layers: visual-to-text (18) and text-to-visual (19). The visual-to-text cross-attention layer enables the model to assess the weight of each language token for a given image region and identify the most relevant word for that patch. On the other hand, the text-to-visual cross-attention layer enables the model to use question representations to identify the most relevant image patch. Through these bidirectional operations, the model remains contextually grounded while also developing a deep understanding of the image and the question [10], resulting in both the attended vectors Q_I, I_q and attention weights $G_{q\to i}, G_{i\to q}$.

To capture hierarchical contextual cues, the attended text features are refined using multi-scale 1-D convolutions (6). Here LP stands for linear projection operation.

$$Q_{ms} = Q_I + LP\left(Conc\left(Conv_{k_1}(Q_I), Conv_{k_2}(Q_I), Conv_{k_3}(Q_I)\right)\right) \quad (6)$$

Next, an adaptive gating layer is introduced to filter out irrelevant information by regulating the flow between the attended and original feature representations. As the attended image and text features pass through this layer, gating weights for both modalities are computed, as defined in (7) and (8). The final filtered representations for the image V' (9) and question Q' (10) are then obtained by adaptively combining the attended and original features through element-wise gating [9]. This mechanism enables the model to generate compact and filtered representations for both modalities. The resulting gated image and text features are subsequently forwarded to the next fusion layer for joint reasoning.

$$g_I = \sigma(Concat(I, I_q).G_{i\to q} + b_{iq}) \quad (7)$$

$$g_q = \sigma(Concat(Q, Q_{ms}).G_{q\to i} + b_{qi}) \quad (8)$$

$$V' = g_I * I_q + (1 - g_I) * I \quad (9)$$

$$Q' = g_q * Q_{ms} + (1 - g_q) * Q \quad (10)$$

Here, $\sigma(\cdot)$ denotes the sigmoid activation function, I and Q are the original visual and text feature representations, I_q and Q_{ms} are their respective cross-attended (and multi-scale enhanced) features, $G_{i\to q}, G_{q\to i}$ and b_{iq}, b_{qi} are learnable parameters (attention weight and bias) and * denotes elementwise multiplication. These refined multimodal features are passed to the fusion module.

2) *Final Fusion Layer:* As $V' \in R^{768}$ and $Q' \in R^{768}$ reaches the final fusion layer the attended text Q' and image representations V' are concatenated to form a joint feature vector $Z = V'||Q' \in R^{1536}$ that incorporates information from both modalities. These multimodal feature representations are then simultaneously passed through two parallel components: the main fusion path and the shortcut highway path [21] to produce two compact intermediate joint embeddings, f and r. The main fusion path consists of a two-layer feed-forward network with nonlinear ReLU activation and dropout regularization. The shortcut highway path provides a direct route for gradient flow during backpropagation to prevent gradients vanishing. The final fused embedding is then defined as $F = f + r \in R^{768}$.
3) *Classifier Layer:* The joint embedding F is fed into a three-layer MLP with ReLU activations and dropout followed by a final linear layer:

$$h' = Dropout(ReLU(W_1.F + b_1)) \quad (11)$$

$$h'' = Dropout(ReLU(W_2.h' + b_2)) \quad (12)$$

$$Z = W_3.h'' + b_3 \in R^{1000} \quad (13)$$

where W_1, W_2, W_3 and b_1, b_2, b_3 represent the learnable weights and biases for the corresponding layers, respectively. The linear layer maps the learned representations to the answer space of 1,000 classes. The final logits represent the unnormalized probabilities for each class and are used by the VQA model to generate a normalized class probability distribution over the answer space. The class with the highest probability is selected as the predicted answer, since it reflects the model's highest confidence in that response.

b) Model Training: We treat VQA as a multi-label classification task due to multiple ground-truth answers per question in VQA-v2. Sigmoid Normalization followed by the BCE loss function is a well-known method for multi-label classification tasks such as VQA [22]. Therefore, we have applied the same loss function (14) to our model to enable it to accommodate partially correct answers [22].

$$L = -\left[y * \log\left(\sigma(z)\right) + (1 - y) * \log(1 - \left(\sigma(z)\right)\right] \quad (14)$$

Here, L stands for loss function, y indicates the true binary level (0 or 1), z the raw logit output, and σ the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. The model then performs backpropagation to compute gradients. Additionally, gradient clipping is applied during training to prevent exploding gradients.

To minimize the loss function, the AdamW optimizer is employed with the following linear warmup and component-specific learning rates as recommended by [23]:

- BERT: 2×10^{-5}(minimal fine-tuning),
- ResNet-50: 1×10^{-4},
- Attention + Fusion + Classifier: 5×10^{-4}.

To make the training process computationally efficient, we have introduced an early stopping logic that terminates training if validation accuracy does not improve beyond a defined threshold after a set number of epochs. The validation phase follows a similar process as training, except no gradient updates are performed.

IV. Model Evaluations and System Deployment

To develop and evaluate the proposed model, we downloaded the recommended training and test sets from *visualqa.org* [6] . These subsets comprise approximately half of the complete VQA v2 dataset and consist solely of balanced, real-world images [5]. The selected data were randomly split into two-thirds for training and one-third for validation. Consequently, the training set contains 443,757 question–answer pairs and 82,783 images, while the validation set includes 214,354 question–answer pairs and 40,504 images.

A. Model Performance

The model performance is evaluated using VQA accuracy metric (14) that is designed exclusively for the VQA task. This involves gathering multiple ground-truths for each question. Since it allows multiple answers, the model gets partial credit for matching an incorrectly annotated answer. This approach is especially suitable for most benchmark VQA datasets like VQA v1 [4] and VQA v2 [5], where 10 different human annotators answer the same question. This technique is also used by the VQA v2 test-standard server and visualqa.org [6].

$$Accuracy = \min\left(\frac{\#Annotation\ with\ the\ answer}{3}, 1\right) \quad (14)$$

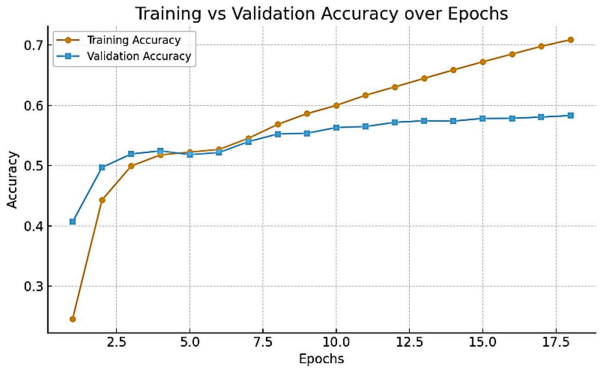

Fig. 1. Training vs. validation accuracy

Fig. 1 summarizes the model's training and validation accuracy over 18 epochs. As we can see, training accuracy increased from 24% to 71%, creating a widening gap from validation accuracy and indicating emerging overfitting. Stronger regularization—such as dropout tuning, data augmentation, or weight decay—may further improve generalization. On the other hand, validation VQA accuracy improved from 40% in the first epoch to 58% at convergence, outperforming the baseline model [12] by approximately 10%. Most accuracy gains occurred within the first five epochs, suggesting an effective learning rate and architecture. Finally, the training concluded at epoch 18 with the early stopping criterion being triggered, as validation accuracy plateaued at around 58%, signaling convergence. Overall, the model demonstrated consistent, stable performance gains, validating its ability to learn complex relationships in the data.

B. Computational Efficiency

A key goal of this project was maintaining a low-cost VQA model. To contextualize efficiency, prior VQA systems often require days of multi-GPU training; for example, Anderson et al. [1] trained on eight NVIDIA M40 GPUs for five days, while Singh et al. [24] required approximately 300 GPU-hours.

In contrast, our model trained efficiently on a single NVIDIA A100-SXM4 GPU through Google Colab Pro, achieving stable convergence in 18 epochs with an average throughput of 2.80 iterations per second. This demonstrates that competitive VQA performance can be achieved with modest computational resources.

C. System Deployment

The VQA model is developed and deployed as a fully cloud-hosted system on Microsoft Azure, leveraging Azure Blob Storage (ABS) for model hosting and frontend delivery, and Azure Container Apps (ACA) [25] for serving the backend inference service. This architecture enables high availability, scalability, and secure communication via HTTPS.

The user interface is illustrated in Fig. 2, where users can upload an image and submit a question through their device's camera, microphone, or manual input, and view the most probable human-readable answers by the model. Various test cases were conducted. Overall, the model demonstrated strong performance in object recognition, color identification, and temporal reasoning, while showing limitations in text recognition (e.g., airline names) and counting tasks—a known challenge for attention-based VQA models.

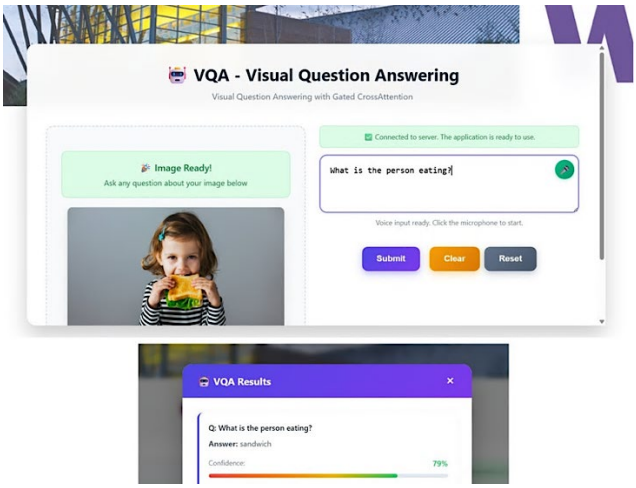

Fig. 2. System interface with one example test case

V. Conclusion

This work presents a cost-efficient Visual Question Answering (VQA) model based on a gated bidirectional cross-attention architecture. The proposed design enables effective cross-modal reasoning by allowing visual and textual representations to attend to each other while employing an adaptive gating mechanism to suppress irrelevant information. This results in more discriminative multimodal features prior to classification. Trained on the VQA v2 dataset, the model achieved 58% accuracy, representing a 10% improvement over the baseline while maintaining computational efficiency. It demonstrates that substantial improvements over a traditional co-attention baseline can be achieved using a lightweight and computationally efficient cross-attention design. The proposed gated bidirectional cross-attention mechanism offers a promising direction for advancing discriminative VQA models and may extend to broader multimodal tasks such as visual entailment, image captioning, and image-text retrieval.

References

[1] P. Anderson et al., "Bottom-up and top-down attention for image captioning and visual question answering," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 6077–6086.

[2] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," Adv. Neural Inf. Process. Syst., vol. 27, pp. 1682–1690, 2014.

[3] M. Ren, R. Kiros, and R. Zemel, "Exploring Models and Data for Image Question Answering," arXiv preprint arXiv:1505.02074, 2015. [Online].

[4] H. Sharma and A. S. Jalal, "A survey of methods, datasets and evaluation metrics for visual question answering," Image Vis. Comput., vol. 116, p. 104327, Dec. 2021, doi: 10.1016/j.imavis.2021.104327.

[5] Y. Goyal, et. al, "Making the V in VQA Matter: Elevating the role of image understanding in visual question answering," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 6325–6334.

[6] "VQA: Visual question answering," [Online]. Available: https://visualqa.org/evaluation.html. [Accessed: Oct. 1, 2025].

[7] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in Proc. 39th Int. Conf. Mach. Learn. (ICML), Baltimore, MD, USA, Jul. 2022, pp. 12888–12900. [Online].

[8] X. Chen et al., "PaLI-X: On scaling up a multilingual vision and language model," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 2024.

[9] W. Li, et. al, "Visual question answering with attention transfer and a cross-modal gating mechanism," Pattern Recognit., vol. 121, 2021.

[10] D.-M. Nguyen-Tran, T. Le, M. L. Nguyen, and H. T. Nguyen, "Bi-directional cross-attention network on Vietnamese visual question answering," in Proc. 36th Pacific Asia Conf. Lang., Inf. Comput. (PACLIC), Manila, Philippines, Oct. 2022, pp. 834–841.

[11] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," arXiv preprint arXiv:1606.00061, Jan. 2017. [Online].

[12] A. Nada, Visual Question Answering (Capstone project report). Univ. Washington Bothell, 2023.

[13] K. He, et. al, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[14] P. Kashyap, "Guide to tokenization and padding with BERT," Medium, Sep. 2023. [Online].

[15] A. Baloian, N. Murrugarra-Llerena, and J. M. Saavedra, "Scalable visual attribute extraction through hidden layers of a residual ConvNet," arXiv:2104.00161, 2021.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv:1607.06450, 2016. [Online].

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. 32nd Int. Conf. Mach. Learn. (ICML), Lille, France, Jul. 2015, pp. 448–456.

[18] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., 15(56), pp. 1929–1958, 2014.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in Proc. 27th Int. Conf. Mach. Learn. (ICML), 2010, pp. 807–814.

[20] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," arXiv:1512.02167, 2015.

[21] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," arXiv:1505.00387, May 2015. [Online].

[22] D. Teney, P. Anderson, X. He, and A. van den Hengel, "Tips and tricks for visual question answering," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 4223–4232.

[23] L. N. Smith, "Cyclical learning rates for training neural networks," in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Santa Rosa, CA, USA, Mar. 2017, pp. 464–472.

[24] A. Singh et al., "Towards VQA models that can read," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 8309–8318.

[25] Microsoft, "Azure Container Apps documentation," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/azure/container-apps/. [Accessed: Oct. 1, 2025].