

Empirical Benchmarking of Vision-Language Transformer Combinations for Visual Question Answering Tasks

Satyam Singh, Aditya Sharma, Smita Tiwari

Department of Computer Science & Engineering, Sharda University

Greater Noida, Uttar Pradesh, India

satyam1508singh@gmail.com, adityasharma344234@gmail.com, smita.tiwari@sharda.ac.in

Abstract

The task of Visual Question Answering (VQA) also requires intelligent methods that bring together visual and textual information in order to compute accurate answers. An exhaustive empirical evaluation of six specialized vision language transformer models for VQA are conducted on the DAQUAR dataset. The experiments use a Vision Transformer (ViT), a Data-efficient Image Transformer (DeiT), and a Swin Transformer as image encoders, and BERT and RoBERTa as language models. Following the same experimental protocol, we track per-epoch validation accuracy, and consider a potential impact on computational efficiency. The results produced the following peak accuracies: ViT + BERT (65.50%), ViT + RoBERTa (68.47%), DeiT + BERT (61.15%), DeiT + RoBERTa (66.47%), Swin + BERT (73.42%) and Swin + RoBERTa (64.50%). The Swin+BERT combination produced the best accuracy among the combinations tested, achieving a peak accuracy of 73.42%. This illustrates the possible power of hierarchical vision transformers combined with powerful language comprehension on VQA tasks. We hope that this benchmark provides useful guidance to researchers when seeking effective vision-language models based on performance and resource limitations.

Keywords: *VQA, Multimodal, Vision-Language Models, NLP, CV, Transformer.*

1. Introduction

Visual question answering (VQA) is a challenging activity that sits at the crossroads of computer vision and natural language processing. It challenges artificial systems for intelligence to comprehend visual information and produce answers for natural language questions, while exhibiting deep comprehension and sophisticated reasoning across different modalities [1]. Instead of performing more straightforward tasks such as image classification or object detection, VQA requires the systems to perform interpretive comprehension of both, the visual context, and the questions expressed in a natural language; in this sense, VQA provides a challenging test environment for multimodal intelligence. VQA systems are able to process real-world images and respond to human language questions in the form of answers this has a potentially wide application across numerous domains such as education, healthcare, assistive technologies for visually impaired individuals, and human-robot interaction [2, 3].

VQA history has roots in early work on image understanding and language modelling. Early VQA models incorporated deep learning with Convolutional Neural Networks primarily to extract visual features, with Recurrent Neural Networks and Long Short-Term Memory networks to process natural language questions [4]. Mainstream architectures typically fused the extracted features through either concatenation or element-wise sum before input into a classification layer to predict the output.

Pioneering data sets, such as DAQUAR and VQA v1, were hugely influential in setting the task up and providing early benchmarks to the research community [5, 6]. A major leap forward in VQA followed the introduction of attention mechanisms, which allowed models to effectively attend to salient areas in an image and individual words in a question, thus achieving improved alignment and interaction between the two modalities [7]. By using more fine-grained reasoning instead of relying on global image features, the model can weigh visual evidence for identifying the answer to the question. However, these earlier attention-based models still used deep convolutional backbones for processing images and so were computationally taxing, as was the growing issue of interpretability with models

advancing to have complexities, leading to the development of utilities to effectively x-ray them and interrogate their internal reasoning and possible biases with the research community [8].

The introduction of transformer architectures, first developed for machine translation [9], revolutionized the AI community. Its ground-breaking invention, the self-attention mechanism showed to be remarkably adept at managing long-range dependencies across contexts for sequential data. After the success of this architecture transferred to image-based datasets with models like ViT, thus proving that sequences did not need to be limited to language but could also represent visual data as sequences of patches, similar to words in a sentence [10]. The result of this convergence were multi-modal transformers that could process and learn joint representations across visual and textual data, thus providing an integral approach to solving VQA [11, 12]. The demand to VQA systems that operate on resource-limited devices has also prompted ongoing studies of lightweight transformer architecture that Favors computational efficiency over overall accuracy [13].

The incredible development and common usage of numerous vision and language transformer architectures, makes it essential to conduct a thorough empirical evaluation of combinations on widely used VQA benchmarks in order to advance future research and build new applications. Vision transformers ViT, DeiT, and Swin, and language models like BERT and RoBERTa, possess distinct architectural advantages and properties of computation. This paper strives to mitigate this gap by conducting a comprehensive experiment of six of the most common vision-language transformer architectures using the DAQUAR dataset. This work aims to find the most effective combinations by accuracy and to discuss the individual architecture's computational properties in detail, providing a practical benchmark that the VQA community can utilize.

2. Research Methodology

The area of Visual Question Answering has changed dramatically from the early RNN and CNN-based models to transformer-based models in the later versions [4]. Of foremost impact was the addition of attention mechanisms, which provided models the ability to focus on precise regions in the image and the question tokens, allowing for better multimodal alignment and reasoning [7]. This model was the precursor for tools such as VQA, which was designed to provide some interpretability of VQA models by visualizing their internal attention and identifying possible biases and issues through attention visualization processes, which provided broad visualization in an interpretive stance [8]. The addition of the Transformer architecture radically changed the perception that the domains of computer vision and natural language processing could not be merged and later provided a substantial impact to VQA [9]. BERT indicated high contextual understanding in NLP, and the ViT exemplified the model extension to images in using images as patch sequences [10, 14]. The transformer model generalized to multimodal transformers that can learn jointly from text and visual, which in many cases are a combined and therefore more efficient VQA solution [11, 12].

The advancements made with vision transformers consist of many things from data-efficient image transformers, improvements to training efficiency through transformers, and the introduction of a hierarchical approach to scalable visual feature extraction with respect to Swin transformers [15, 16]. In the multimodal space, the introduction of cross-modal attention using those such as ViLBERT and LXMERT allowed for deep cross-modal interactions between the streams of information being generated from visual and linguistic modalities [17, 18]. The benefit of efficiency has also, the lightweight transformer architecture has made VQA efficient with a high degree of accuracy and very low computational complexity [13]. Beyond VQA, these multimodal methods using transformers have predicted with accuracy in a range of applications, like improving customer service chatbots by taking the user query and embedding contextual information to generate a response or designing medical diagnostic pipelines by combining multiple signal modalities in to one visual of extracting information [19–21].

2.1 Dataset

DAQUAR (Dataset for Question Answering on Real-world images) dataset is the first benchmark for VQA research relating to real indoor scenes. Included in DAQUAR are thousands of human annotated question-answer pairs for a collection of images depicting complex indoor scenes ranging from kitchens to living rooms and home offices. The images used in DAQUAR are taken mostly from the NYU-Depth V2 dataset and are ambiguous with cluttered and complex scenes with multiple objects. In these types of images, object detection, spatial reasoning and fine-grained interpretation become extremely important. The questions included in DAQUAR test several facets of the models' abilities to understand visual and text information - looking for the presence of an object, its attributes such as colour, spatial relationships, and counting. DAQUAR has a defined vocabulary of answer categories, and thus the task can be framed as a multi-class classification problem. Having defined training and test splits, DAQUAR is an important benchmark that provides a challenge to developing models for fine-grained visual understanding and complex reasoning involving real-world indoor scenes. The DAQUAR dataset that was used in this study is available at <https://www.kaggle.com/datasets/tezansahu/processeddaquar-dataset>.

Table 1: DAQUAR Dataset File Structure and Descriptions

File Name	Description
images	Folder containing all of the indoor scene images that are used for VQA. Each image may have different ids for instance.
train.json	A JSON file that contains question-answer pairs for the training split. A question-answer pair associates a question, image id, and answer(s).
test.json	A JSON file for validation or testing that contains question-answer pairs that have a similar structure.
answer.txt	A text file listing all valid answers accepted during preprocessing and training.
vocab.txt	A vocabulary listing of questions or answers given in various forms, sometimes for language casing.
marks.json	Sometimes there are more detailed marks per image/question, such as object segmentation masks.

3.2 Preprocessing

In order to make sure that the DAQUAR dataset was consistent, had integrity and was compatible with the model pipeline, the DAQUAR data set was subjected to a formal preprocessing and filtering process. The following steps were taken:

- **Dataset Acquisition**
 - The DAQUAR dataset was acquired and extracted from the official ZIP file.
 - The dataset includes both images and corresponding natural language question-answer pairs.
- **Answer Space Filtering**
 - Deleted any question-answer pairs with answers that were not included in the defined vocabulary in answer.txt.
 - Validated that all target labels for classification were correct and within scope.
 - Tried to keep the answer space clean and well defined to minimize potential label mismatches in training.
- **Image File Validation**
 - Validated that all image files were physically present in the image's directory of the dataset.
 - Identified and removed any row/script without visual content to avoid runtime errors when trying to load an image.
 - Knew that all training instances must have both text and visuals.

- **Final Data Statistics**

- The size of the dataset initially was: 6,795 training samples and 5,673 validation samples.
- After filtering: 6,149 training samples and 5,076 validation samples.
- Used the standard training and evaluation splits provided with the processed DAQUAR dataset.

The filtering and preprocessing steps produced a cleansed and consistent dataset that was ready for model training and evaluation. The model uses the standard training and evaluation splits provided from the processed DAQUAR dataset. An example from the dataset has been illustrated in Figure 1.

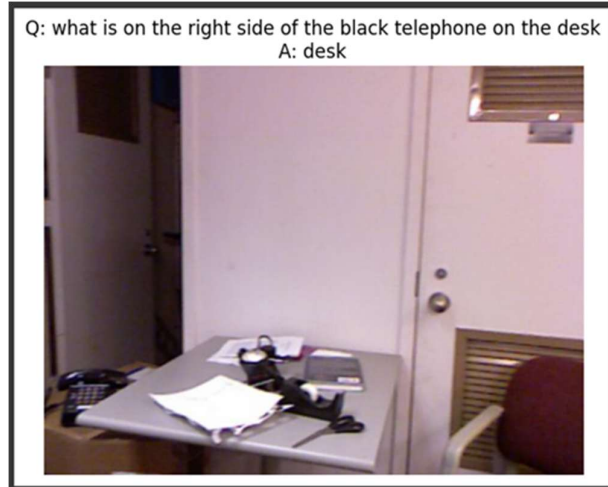


Figure 1: Sample image from the DAQUAR dataset [1].

3.3 Image Encoder

The function of Image Encoder is to extract visual features from the input image that will be merged with language features for multimodal understanding. The encoder can use transformer architecture based visual backbones such as ViT, DeiT, or Swin Transformer to generate rich image representations.

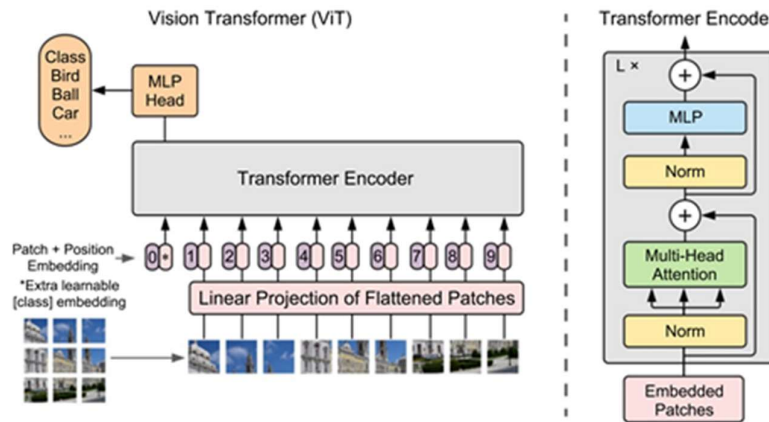


Figure 2: Vision Transformer architecture adapted from [10].

3.3.1 Vision Transformer: The Vision Transformer has made a tremendous impact in the area of computer vision in part due to experimental results demonstrating a transformer architecture, with no convolutions, can be used to achieve state-of-the-art results in image classification tasks.

- **Patch Processing:** The input image is separated into fixed size patches.
- **Linear Encoding:** Each patch is mapped to a vector representation.
- **Sequence Formation:** Combining the patch embeddings into a sequence of tokens, there is a learned class token added to the start.

- **Self-Attention:** A multi-head self-attention mechanism represent the relationships between the patches.
- **Feature Extraction:** The final hidden state that corresponds to the class token is considered to be the representation of the image as a whole.

Essentially, ViT shares all the architectural characteristics of transformers as multi-head self-attention is utilized to represent relevance between instances of different patches.

3.3.2 Data-Efficient Image Transformer: The Data-Effective Image Transformer (DeiT) builds on the Vision Transformer architecture and focuses on training efficiency, especially when training with limited data. DeiT defines a new distillation token, allowing the transformer model to learn from a teacher network (most commonly a convolutional neural network) and to transfer knowledge in that way, leading to better performance. This allows DeiT to achieve performance competitive with the large-scale pre-training of models from other architectures without leveraging large external datasets. Similar to ViT, DeiT divides images into a sequence of image patches of fixed sizes, applies a linear embedding to each patch, and feeds them into a transformer encoder for feature extraction. The distillation token is processed in parallel with the class token, and interacts with the patch embeddings via self-attention layers. In this way, the network benefits from direct supervision via the class and patch tokens, along with supervision from the teacher via the distillation token.

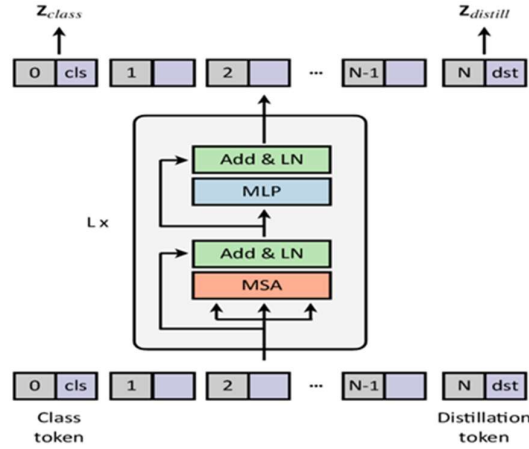


Figure 3: Architecture of the Data-Efficient Image Transformer (DeiT), adapted from [15].

3.3.3 Swin Transformer: The Transformer efficiently reduces the quadratic computational complexity of global self-attention in its hierarchical structure by utilizing a shifted array of windowing. The first component of the Swin Transformer's approach to self-attention, is only computing self-attention locally rather than globally, thus the self-attention is only computed locally to nonoverlapping windows in the feature map at the first layer. This local self-attention reduces resource usage. Then in the next layers of the architecture, the self-attention is used in a cross-window manner. Windows were shifted to create connections within self-attention, created neighbouring regions to be able to attend to one another. This allows the Swin Transformer to efficiently capture the local context and conduct global self-attention and, thus, has proven to be one of the most efficient architectures to date for a variety of vision tasks, with a solid multiscale feature representation, and generalise and scale strongly.

3.4 Text Encoder

The Text Encoder is responsible for processing the natural language question and extracting linguistic features. It typically uses transformer-based architectures such as BERT or RoBERTa to understand contextual word embeddings. The encoder leverages multiheaded self-attention, using the definition below:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where Q , K , and V are the query, key, and value matrices, respectively, and dk is the dimension of the keys.

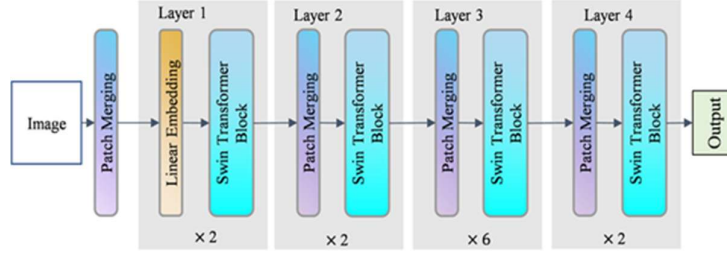


Figure 4: Swin Transformer architecture adapted from [16].

where d_k denotes the key dimension. For ViT, DeiT, and Swin Transformers, the image is first divided into fixed-size patches, each patch is flattened and projected into an embedding vector. For BERT and RoBERTa, the input question tokens are embedded through multi-layer bidirectional transformer blocks. In this work, the visual feature vector F_V and language feature vector F_T are extracted independently, and are then fused by concatenation:

$$F_{fusion} = [F_V; F_T]$$

The fused vector is passed through a fully connected projection layer and a nonlinear activation to map the representation to a compact joint embedding:

$$F_{joint} = ReLU(W \cdot F_{fusion} + b)$$

Finally, the classification head produces logits over the answer vocabulary, followed by SoftMax to obtain the final predicted answer.

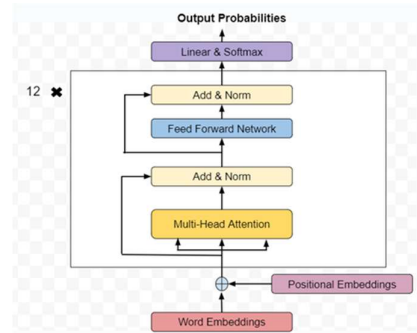


Figure 5: BERT architecture adapted from [14].

3.4.1 Bidirectional Encoder Representations from Transformers: BERT is a base language model that took the NLP world by storm with its bidirectional training. All the other models processed the text either left-to-right or right-to-left, while BERT does both. It uses masked language modeling and next-sentence prediction tasks to learn on large text corpora, allowing BERT will be able to create rich contextualized word embeddings in the first step.

3.4.2 RoBERTa: It is an optimized version of BERT that improves its predecessor's performance with a number of large changes. These changes included pre-training on a much larger dataset, implementing dynamic masking patterns with the tokens that are masked varying by epoch, drop the NSP objective, and pre-training with larger mini-batches and longer sequences. As a result, much more stable and higher performing language representations were produced than were observed in baseline BERT.

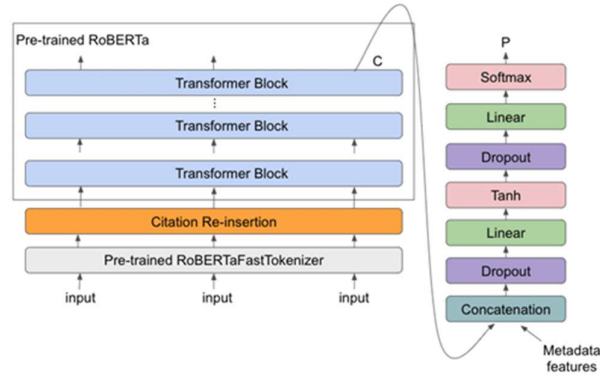


Figure 6: Conceptual architecture of RoBERTa adapted from [21].

4 Model Architecture and Experimental Setup

The visual and textual features attained are combined by concatenation. For the ViT and DeiT models, both provide feature vectors of size 768; therefore, the concatenated feature vector has size $768 + 768 = 1536$. For the Swin transformer that produces features of size 1024, the concatenated feature vector has size $1024 + 768 = 1792$. The concatenated feature vector is passed to a linear layer that projects the concatenated vector down to civil representation 4 of 512 dimensions, followed by application of a ReLU activation function

4.1 Model Architecture

The classification head is a linear layer that maps the fused multimodal embeddings to target answer classes, the classification head then applies softmax activation to create a probability distribution. The specific extraction of image features was conducted using the Vision Transformer (ViT), Data-efficient Image Transformer (DeiT), and Swin Transformer architectures. Language features were extracted from BERT and RoBERTa models. The features were then fused together in the classification head via concatenation and passed through the classification head.

Throughout the training procedure, the training for all model parameters, including both feature extractors and classification head were fine-tuned. The architecture was developed with PyTorch 2.x and HuggingFace Transformers 4.x, ensuring that the architecture is current with contemporary transformer models for vision and language. The overall architecture for each model combination was two-stream, whereby visual and textual feature representations flow through the model independently before being fused. An example of the entirety of the architecture is shown in Figure 7.

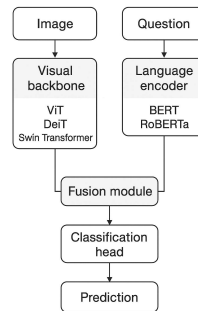


Figure 7: Simple architecture of the VQA system.

4.2 Experimental Setup

The experimental procedures were performed with the AdamW optimizer (PyTorch 2.x) at a learning rate of 2×10^{-5} . Each model was trained for 80 - 100 epochs with nn.CrossEntropyLoss as the loss function. The data was managed with PyTorch's DataLoader (torch.utils.data (PyTorch 2.x) with a batch size of 8 and num_workers = 2 to allow for parallel data loading. The performance of the model

was evaluated after each epoch on a dedicated validation set, with classification accuracy as the key performance metric. Computational resources, including per-epoch training time and estimated GPU memory usage, were monitored to assess overall efficiency. Final model parameters were saved upon completion of training. The experimental environment included Python version 3.11 (or 3.x), PyTorch version 2.x, torchvision version 0.15 (or 0.16), and HuggingFace Transformers version 4.x for loading models and tokenizers.

Algorithm 1 VQA Inference Pipeline

Input: Image I , Question Q

Step 1: Image Preprocessing

Resize I to 224×224 pixels

Convert to tensor and normalize pixel values

Step 2: Question Tokenization

Tokenize Q using BERT or RoBERTa tokenizer

Pad and truncate to fixed max_length

Create attention mask for tokenized sequence

Step 3: Feature Extraction

Extract visual features FI using image encoder (ViT, DeiT, Swin)

Extract textual features FQ using text encoder (BERT, RoBERTa)

Step 4: Feature Fusion

Concatenate FI and FQ to form $F_{\text{multimodal}}$

Pass through linear layer + ReLU:

$$F_{\text{joint}} = \text{ReLU}(W F_{\text{multimodal}} + b)$$

Step 5: Answer Prediction

Compute logits: $\text{logits} = \text{FC}(F_{\text{joint}})$

Apply softmax over logits to get probabilities

Select answer: $A = \arg \max(\text{softmax}(\text{logits}))$

return Predicted answer A

4.3 Evaluation Metrics

The performance of the models was evaluated on the following criteria:

- **Accuracy:** The principal evaluation metric the ratio of the number of correct predictions to the total number of questions in the validation set. Accuracy is an absolute representation of the classification performance of a model and is the most frequently used benchmarking metric in visual question answering systems. A higher accuracy indicates the model consistently selected the correct answer from the candidates. For this research, accuracy was computed over the entire validation set and no sampling was necessary and therefore, is a fair comparison of all architectures to each other. The advantage of accuracy measured was that it allowed us to compare our recorded results with reported findings on the dataset.

$$\text{Accuracy} = \frac{\text{number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- **Training Loss:** The average Cross-Entropy Loss computed over the training data in the last epoch. The loss function calculates the distance between the predicted probability distribution and the empirical label distribution. In general, a lower training loss suggests that the model has correctly learned to map inputs to outputs, or low loss could also mean that the model is overfitting the training data. Monitoring training loss across training epochs is useful in answering two questions: is the model still improving and is the model still stable with respect

to training loss, meaning it is not displaying possibly training instabilities such as vanishing/exploding gradients.

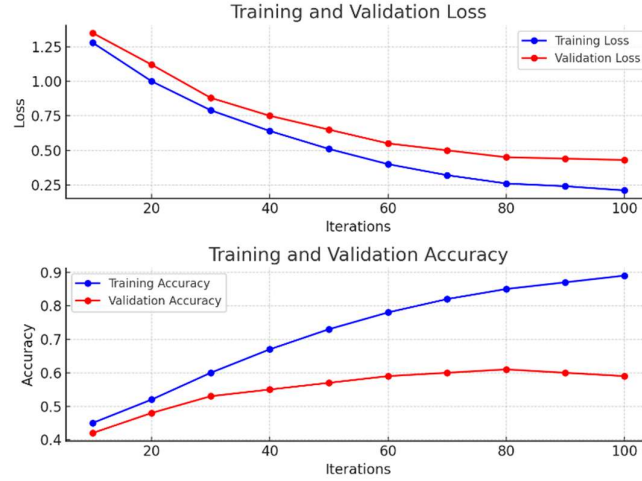


Figure 8: Training and validation loss and accuracy over iterations

- **Computation Time:** The total time (in seconds) that it takes for each model to go through the full training process, from first epoch to converged. Computation time is especially important when comparing models with similar accuracy, but different computational costs, as lower training time might be important in large-scale experiments or application in real time. For the sake of completeness, we also tracked training time per epoch and GPU memory use; these values provide additional information to help understand the efficiency and scalability of each architecture when applied to data sets of different size and complexity.

3. Results and Discussion

A systematic evaluation was performed across six vision–language transformer combinations on the DAQUAR dataset. Table 2 summarises the final validation accuracies, final loss values and total training time (hours) observed for each configuration. The Swin+BERT combination achieved the highest peak validation accuracy of 73.42% (final loss 1.4755, training time 5.12 hrs), followed by ViT+RoBERTa (68.47%) and DeiT+RoBERTa (66.47%). The full set of results is shown in Table 2.

Table 2 (summary, place near this paragraph):

Model	Loss	Accuracy	Training Time
Vit_bert	1.53	66.50%	4.45
Vit_roberta	1.49	68.47%	4.59
Swin_bert	1.47	73.42%	5.12
Swin_roberta	1.44	64.50%	6.11
Deit_bert	1.54	61.53%	4.57
Deit_roberta	1.48	66.47%	4.47

The superior performance of Swin+BERT appears to stem from the Swin transformer’s hierarchical, shifted-window attention that better captures local and multi-scale visual structure, combined with BERT’s rich contextual language representations. This synergy enabled more effective visual–verbal alignment on the DAQUAR indoor-scene questions, explaining the relatively higher accuracy for that configuration. Conversely, while Swin+RoBERTa produced a slightly lower loss, its accuracy was reduced (64.50%), suggesting that differences in pre-training regimes and optimisation dynamics between BERT and RoBERTa can influence final classification accuracy in non-trivial ways.

Comparing the ViT and DeiT backbones, ViT+RoBERTa outperformed ViT+BERT and all DeiT configurations, indicating that RoBERTa's larger-scale pretraining and dynamic masking may pair well with patch-based ViT embeddings for this dataset. When the DeiT backbone (intended for data efficiency) was merged with BERT, it attained a lower peak accuracy (61.53%). However, the performance was elevated when DeiT was combined with RoBERTa (66.47%), thus, the selection of a language model can alleviate to some extent the use of a lighter visual encoder. These comparisons highlight the practical trade-offs between model complexity, pretraining strategy, and dataset characteristics.

Computational efficiency is an important consideration. Although Swin+BERT gave the highest accuracy, it also required moderate training time (5.12 hrs) and larger GPU resources compared to ViT/DeiT configurations. Swin+RoBERTa in our runs took the longest (6.11 hrs) yet produced lower accuracy underscoring an effectiveness vs. cost trade-off that practitioners must consider when selecting a model for deployment or further scaling. For applications where resources are constrained, ViT+RoBERTa or DeiT+RoBERTa may offer a favourable balance of accuracy and efficiency.

Limitations and noisy factors: The experiments used the processed DAQUAR splits after filtering; final dataset counts were reduced through preprocessing (from ~6,795 to ~6,149 training samples and ~5,673 to ~5,076 validation samples), which may affect generalisability to larger, more diverse VQA datasets. Training durations reported reflect the experimental protocol used here (80–100 epochs with batch size 8 and AdamW at $\text{lr}=2 \times 10^{-5}$) extending epochs or tuning hyperparameters further may change relative rankings. Additionally, answer vocabulary filtering (kept only answers in answer.txt) constrains the answer space and can inflate apparent classification accuracy compared to open-answer settings.

Interpretation and practical implications: The results imply that hierarchical visual encoders that maintain locality (Swin) can provide tangible benefits for indoor-scene VQA tasks where fine-grained local structure matters. Meanwhile, the language encoder choice modulates how well the fused multimodal representation can be interpreted by the classifier RoBERTa tends to improve performance with ViT/DeiT, while BERT pairs especially well with Swin in this benchmark. Practitioners should therefore evaluate both visual and language backbones jointly rather than optimising them in isolation.

Suggestions for sub-sequent experiments

- (1) Perform prolonged hyperparameter variations (learning rates, weight decay, batch size) and longer training to see if the ranking of the results that were observed persists after full convergence.
- (2) Benchmark bigger and different VQA datasets like VQA v2, TextVQA to test generalisability.
- (3) Investigate different cross-modal fusion methods besides concatenation such as cross-attention, gated fusion to find out whether the Swin+BERT advantage can be further enhanced or a lighter backbone can be used to get the same effect.
- (4) Provide the accuracy for each question type (counting, colour, object presence, spatial) in order to deepen the understanding of the strengths and weaknesses of each pairing.

References

- [1] Pandey, Anupam, Deepjyoti Bodo, Arpan Phukan, and Asif Ekbal. "The quest for visual understanding: A journey through the evolution of visual question answering." *arXiv preprint arXiv:2501.07109* (2025).
- [2] Khandaker, Sajidul Islam, Tahmina Talukdar, Prima Sarker, Md Humaion Kabir Mehedi, Ehsanur Rahman Rhythm, and Annajiat Alim Rasel. "Vision meets language: Multimodal transformers elevating predictive power in visual question answering." In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pp. 1-6. IEEE, 2023.
- [3] C. Dewi et al., "Adjusted Reasoning Module for Deep Visual Question Answering Using Vision Transformer," *Comput. Mater. Contin.*, vol. 82, no. 1, pp. 1–22, 2024.
- [4] A. Pujari et al., "Visual Question and Answering (VQA): ViT/SwinT and BERT/ROBERTA," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 13, no. 7, pp. 47–51, 2024.

- [5] Gusain, N., & Sharma, H. (2025). Communication-Efficient Federated Learning in Industrial IoT—A Framework for Real-Time Threat Detection and Secure Device Coordination. *International Journal on Computational Modelling Applications*, 2(2), 18-29.
- [6] Dekshit, S., J. Raghav, G. Shrivastava, and K. Sharma. "Graphic system based on flood fill algorithm with images." In *International Conference on Recent Development in Control, Communication & Computer Technology*, pp. 24-27. 2012.
- [7] Sharma, Himanshu, Prabhat Kumar, Kavita Sharma, and Nitish Raj. "Hybrid Deep Learning based Attack Detection Across Different Layers in IoT Environments." *Defence Science Journal* 76, no. 1 (2026): 41-54.
- [8] T. Jaunet et al., "VisQA: X-raying Vision and Language Reasoning in Transformers," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 976–986, 2022.
- [9] Gussain, Nutan, Ashok Kumar, and H. L. Mandoria. "Performance Analysis of Image and Audio Compression Technique Using Discrete Wavelet Transform." *International Journal for Research in Management and Technology* 4: 63-66.
- [10] A. Dosovitskiy et al., "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale," in *Proc. ICLR*, 2021.
- [11] W. Kim et al., "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision," in *Proc. ICML*, 2021.
- [12] C. Gao et al., "Chop Chop BERT: Visual Question Answering by Chopping VisualBERT's Heads," *arXiv preprint arXiv:2104.14741*, 2021.
- [13] L. Hackel et al., "LIT-4-RSVQA: Lightweight Transformer-Based Visual Question Answering in Remote Sensing," *arXiv preprint arXiv:2306.00758*, 2023. [
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
- [15] H. Touvron et al., "Training Data-Efficient Image Transformers & Distillation Through Attention," in *Proc. ICML*, 2021, pp. 10347–10357.
- [16] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proc. ICCV*, 2021, pp. 10012–10022.
- [17] J. Lu et al., "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks," in *Proc. NeurIPS*, 2019.
- [18] Nandan, A., Pradhan, L., Sharma, H., Bhola, A., & Srivastava, A. (2025, December). Multi-Task Learning Model for Fine-Grained Categorization of Emergency Tweets. In *2025 IEEE 17th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 2045-2050). IEEE.
- [19] Yousef, Rammah, Gaurav Gupta, Nabhan Yousef, and Manju Khari. "A holistic overview of deep learning approach in medical imaging." *Multimedia Systems* 28, no. 3 (2022): 881-914.
- [20] S. Ghosh and B. Barman, "Know Your Heart: A Multimodal Pipeline for Effective Detection and Inference of Heart Diseases," in *Proc. CODS-COMAD*, 2024.
- [21] Rajagopal, Aghila, Gyanendra Prasad Joshi, A. Ramachandran, R. T. Subhalakshmi, Manju Khari, Sudan Jha, K. Shankar, and Jinsang You. "A deep learning model based on multi-objective particle swarm optimization for scene classification in unmanned aerial vehicles." *IEEE Access* 8 (2020): 135383-135393.