50.038 Computational Data Science
Visual Question-Answering (VQA)

Bryan Tan (Chen Zhengyu) — 1004318
Christy Lau Jin Yun — 1005330
Lek Jie Wei — 1005007

Computer Science and Design (CSD)
Singapore University of Technology and Design

April 21, 2023

Abstract

We present the development of a multi-modal Visual Question Answering (VQA) model using a late-fusion approach, which processes both visual and textual information in images. In doing so, we analysed the impact of various hyperparameters (such as encoder choice and classifier structure) on the model's performance and made comparisons with existing baseline models. The model outperforms a naive CNN + LSTM baseline but falls short of the performance of the Vision-and-Language Transformer (ViLT) model. The research highlights the challenges and limitations of VQA and suggests potential future improvements, emphasizing the significance of multi-modal models in enhancing accessibility and practical applications across various domains. The code for our project can be found here.

Contents

1 Introduction

A large proportion of past mainstream advancements in Natural language Question Answering models are confined to text only. A model that can incorporate both visual and textual information in images can improve accessibility and image search results.

The multi-modal models developed through this project such as social media, e-commerce product recommendations and automated.

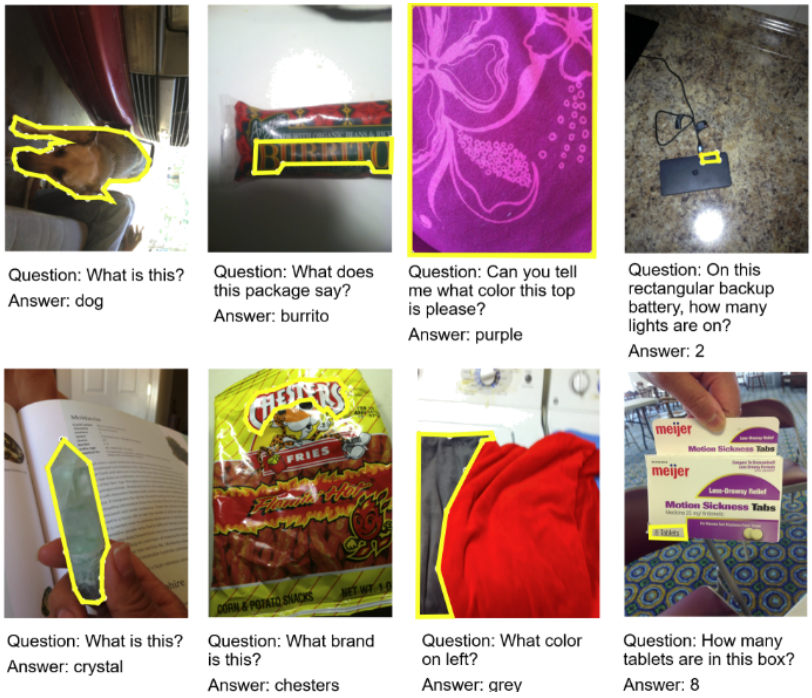


Figure 1: Example of a Visual Question Answering (VQA) task.

2 Dataset and Collection

We utilised the Visual Question Answering v2.0 (VQA v2.0) dataset, which consists of **443,757 questions for 82,783 images in training set**, as well as **214,354 questions for 40,504 images in the validation set**. The images are obtained from the Common Objects in Context (COCO) dataset, while the annotations include a multiple-choice answer and a list of 10 ground-truth answers for each question. In our project, we use the multiple-choice answers as the ground truth labels for the dataset.

Another dataset we considered was the DAQUAR (Dataset for Question Answering on Real-world Images), which includes 9,974 and 2,494 examples in the training and test sets respectively. However, we ultimately decided against using it due to several limitations. Firstly, the dataset size is relatively small, which could hinder the model's ability to generalize to diverse scenarios. Secondly, some questions within the dataset are poorly

constructed, nonsensical, or contain incorrect or subjective answers. Lastly, the dataset's images predominantly feature dimly-lit indoor scenes, which may not adequately represent the real-world use cases of VQA models.

2.1 Data Visualisation

Distribution of Answer Type based on Question asked
For every question in the train set, there is also a corresponding answer for it. For example:

	answer
question	
What is this photo taken looking through?	net

Figure 2: Example question in dataframe

Using this, we plotted the distribution of answer type based on the question using the following labels, which pattern appear more commonly in the dataframe:

- **yes/no question:** answers which are only either yes or no.
- **number question:** answers which contain a number only.
- **other question:** any answer that does not fall under the categories above.

When first started with the VQA 2.0 dataset, we get the following distribution:

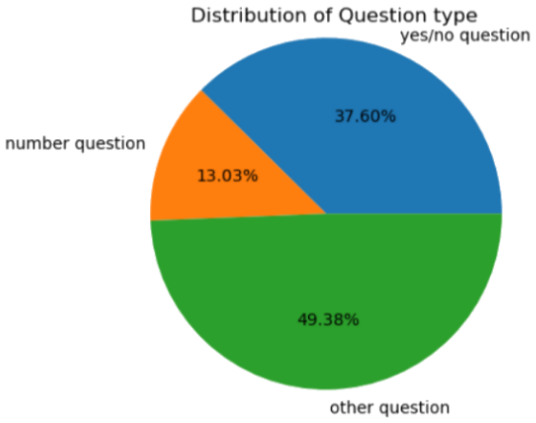


Figure 3: Piechart distribution of answer types from VQA dataset

It is still acceptable that we have a large proportion of the question being in the 'other' label, since they might also contain a wide repertoire of different open ended answers. However, in our findings, yes/no questions made up 37.6% of the dataset, which made up a relatively large proportion of the dataset, which was undesirable.

Cluster Label Distribution

Using this dataset, we also wanted to see what the different cluster label distribution the dataset had.

We used KMeans Clustering to give insight on what type of topic each question is asking for. To do so, we needed to tokenise the data and remove stop-words from the questions. Then we vectoriesed the data using the Tf-Idf technique. Afterwards, we determined a satisfactory number of clusters by plotting the sum of squared distance error against the number of clusters (Figure 4). After assigning each cluster number to every cluster, we can then assign each cluster a word to represent that cluster using Gensim's 'summarize' function. Since it is possible that using summarise and get the same summarised word across different clusters, we also allowed for a two-word summary that better splits the cluster labels if necessary (Figure 6).

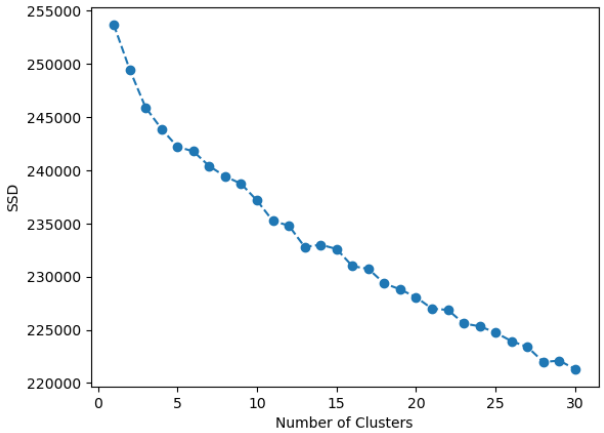


Figure 4: Graph of Number of Clusters against Sum of Squared Distances

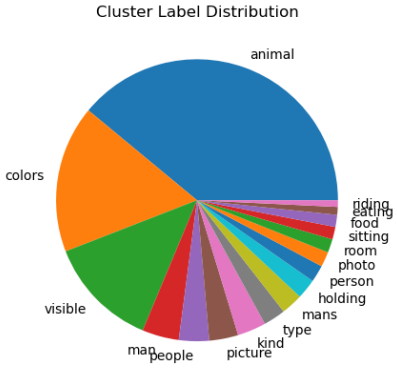


Figure 5: Cluster Label Distribution with labels summarised as one word.

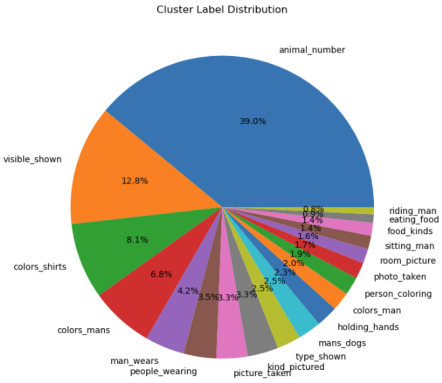


Figure 6: Cluster Label Distribution with labels summarised as two words.

Distribution of Question Headers (First 2 Words)

Another visualisation that we did was to find out the distribution of the different question header, which we defined as the first 2 words of every question. We believed that question headers also constituted what kind of answer it should give. For example, a "How many" question header will always give a number answer. With this dataset, we had the following distribution for 2 words (Figure 7), and the first words (Figure 8). As well as the individual distribution for each first word (Figure 9).

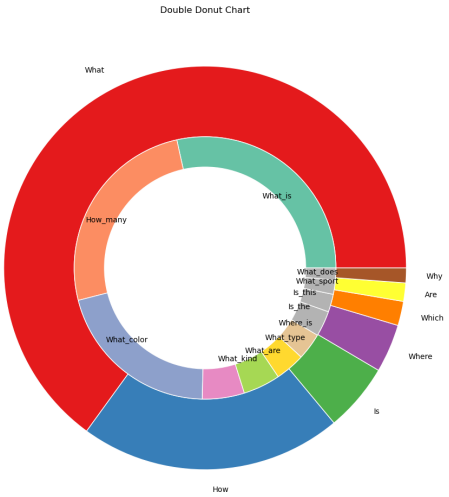


Figure 7: Distribution of first two words in each question as a double pie chart.

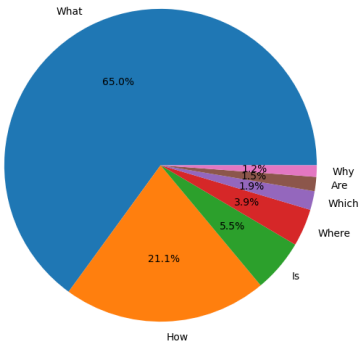
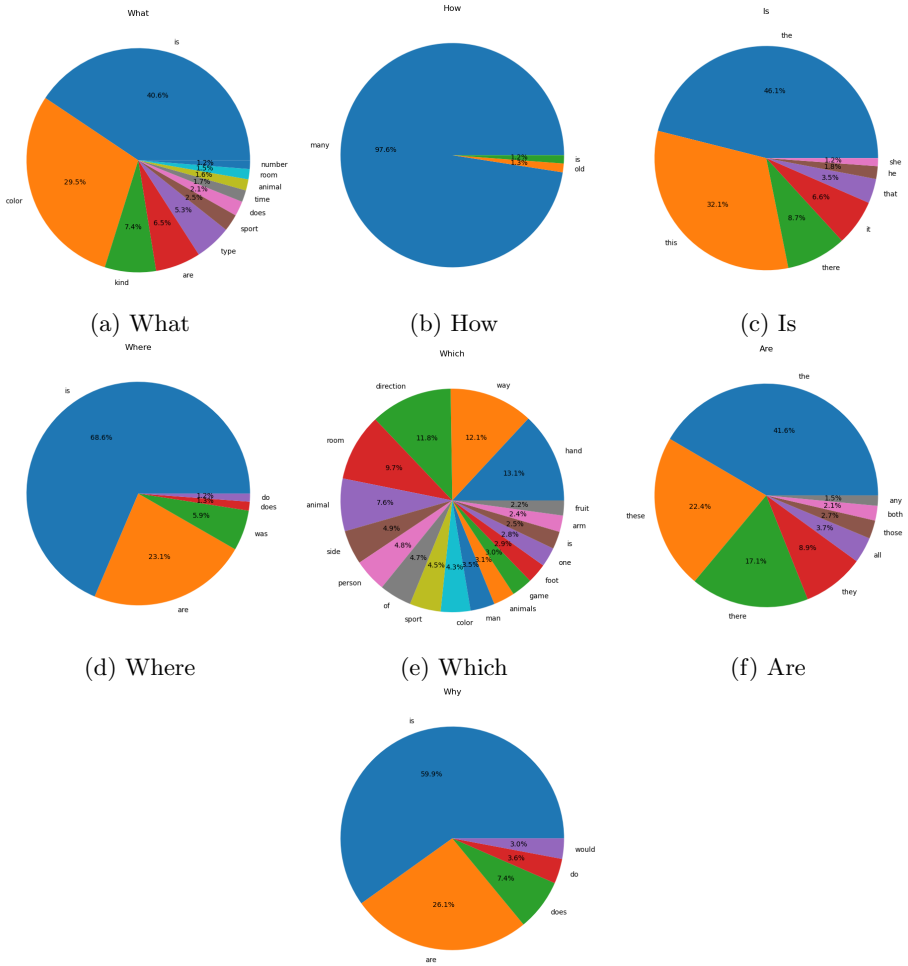


Figure 8: Distribution of first words in each question.

We omitted results that made up less than 1% of total results for clarity.



(a) What (b) How (c) Is

(d) Where (e) Which (f) Are

(g) Why

Figure 9: Distribution of second words in each question w.r.t. their first word.

2.2 Data Pre-Processing

Several steps were undertaken to ensure the data was appropriately cleaned and prepared for analysis. The original dataset comprises training examples, each consisting of an image, a question, and 10 answers provided by annotators, with an associated confidence level.

Following the common practice when using the VQA v2.0 dataset, we convert the visual question-answering task into a multi-class classification problem, where only answers that have appeared at least 9 times across the training and validation sets are included

in the labels. This led to a reduction from 29,332 unique answers to only 3,129 unique answers. These 3,129 unique answers covered 413,433 examples and 199,613 examples in the training and validation sets respectively (613,046 in total), thus 93.15% of the dataset was preserved and minimal data was lost in this pre-processing step.

Next, the training set of the VQAv2 dataset was split into Train and Test sets, consisting of 330,746 and 82,687 samples, respectively. Stratified sampling (by question type) was employed to ensure a representative distribution of examples, with a test-size parameter of 0.2 to allocate 20% of the samples to the Test set.

Lastly, we tested an optional preprocessing step whereby we reduced the number of yes/no questions in the dataset. Initially, yes/no questions accounted for approximately 40% of the samples. By limiting their representation to around 16%, the dataset became less biased towards these types of questions and answers, allowing for a more balanced evaluation of the VQA model's performance across different question types.

3 Existing Works

3.1 Basic CNN + LSTM (Naive Baseline)

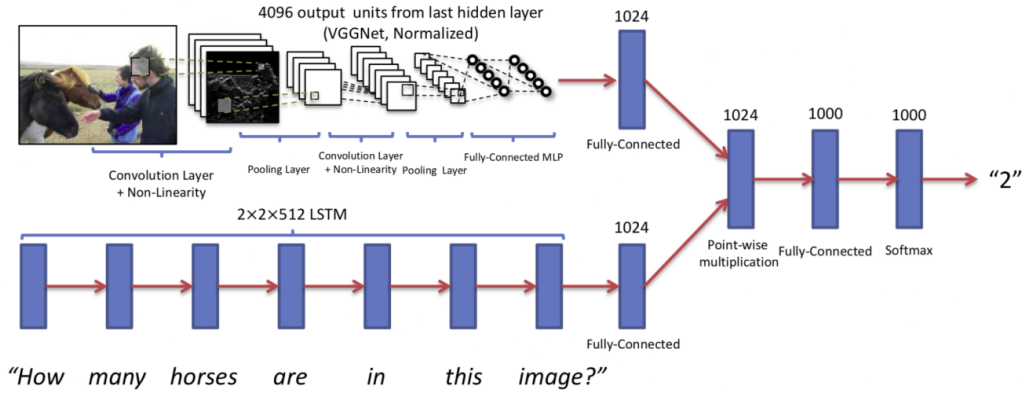


Figure 10: A graphical representation of the baseline VQA model in their original VQA dataset paper.

This approach was outlined by the original paper which proposed the task of free-form and open-ended Visual Question Answering (VQA). It is a simple naive approach that features the pointwise multiplication of the Fully-Connected (FC) output layers of a VGGNet CNN with that of an LSTM, followed by an FC classifier head with softmax as the final output layer.

3.2 ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision (Strong Baseline)

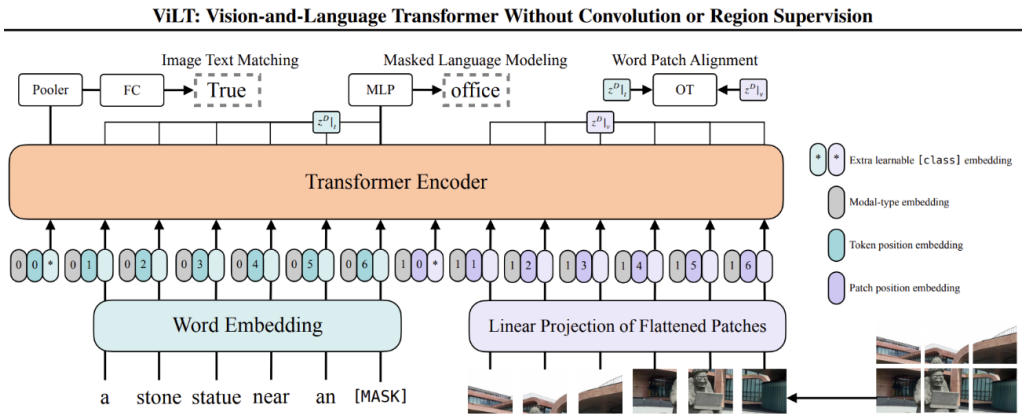


Figure 11: A graphical representation of the ViLT model in their original paper.

The Vision-and-Language Transformer (ViLT) is a novel model proposed for Visual Question Answering (VQA), which uniquely combines two modalities within a single unified architecture. Distinct from previous VLP models, ViLT employs simple, convolution-free embedding of visual inputs, resulting in significantly smaller model sizes and reduced running time. Model performance is increased by focusing computations on multimodal interactions.

4 Methods

4.1 Models

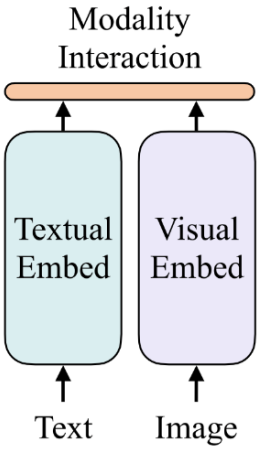


Figure 12: A graphical representation of our late fusion model featuring a simple classifier head.

Our time and resource constraint precludes the use of complex generative models due to their prohibitive size. Hence, our proposed fusion model for Visual Question Answering (VQA) is a relatively simple late fusion model that combines visual and textual features extracted independently. The model tokenizes and embeds questions using pre-trained encoder models like RoBERTa and BERT. Concurrently, image features are extracted using models like CNNs or pre-trained models like ViT and BEiT. The visual and textual features are then fused via concatenation and passed as inputs to a classification head for modality interaction. The classifier output is a vector of size 3,129, containing logits for each possible label. While the late fusion model is computationally heavier in the textual and visual embedding portions, it maintains simplicity in the classifier.

4.2 Loss and Evaluation Metrics

We evaluate the performance of our models based on Accuracy, F1 Score and Wu-Palmer Similarity (WUPs).

4.2.1 Accuracy

Naive measure of exact matches. It is the ratio of correct prediction to the total number of predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{All Samples}} \tag{1}$$

4.2.2 F1 Score

Harmonic mean of precision and accuracy. In a large multi-class problem however (e.g. 3,129 classes), F1 is likely to be very low and overly strict as a metric due to the high number of FPs leading to low precision

$$\text{F1 Score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{2}$$

4.2.3 Wu-Palmer Similarity

Wu-Palmer Similarity (WUPs) captures semantic similarities of strings/concepts based on the longest common subsequence in the taxonomy tree. This is our main evaluation metric as we wish to evaluate models based on their ability to give reasonable answers based on the type of question asked. Previous metrics may be overly strict due to the high number of possible classes and the lack of partial credits for predictions that are semantically similar to the ground truth.

We acknowledge that Wu-Palmer similarity may not be the most suitable evaluation metric for multiclass classification in VQA, as it measures semantic similarity on a continuous scale and does not inherently distinguish between correct and incorrect answers, making it less appropriate for classification tasks. For instance, if the answer space contains semantically similar words (e.g., only words describing types of colours), the similarity score could be high even for entirely incorrect answers. Moreover, WUPS does not work well with phrases.

However, we believe that there is a sufficient number of single-word answers among the 3,129 possible labels in our answer space. We also believe that the answer space encompasses a sufficiently diverse and comprehensive range of semantic meanings, ensuring the applicability and relevance of using WUPS as a supplementary evaluation metric for our VQA model.

$$\text{Wu-Palmer} = 2 \times \frac{\text{depth(lcs(s1,s2))}}{(\text{depth(s1)}) + (\text{depth(s2)})} \tag{3}$$

4.2.4 Cross-Entropy Loss

Since we are performing multiclass classification, we use Cross-Entropy Loss as the objective function.

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output} \\ \text{size}}} y_i \cdot \hat{y}_i \tag{4}$$

4.3 Training

For the optimization process, we employed Huggingface's implementation of the AdamW optimizer, which is a popular choice for training deep learning models due to its efficiency

and effectiveness in handling sparse gradients.

The model was trained with a batch size of 32 and the training process was performed over a total of 5 epochs, providing a reasonable trade-off between model performance and training time.

To ensure reproducibility and facilitate consistent results across different runs, we set the random seed to 12345. This approach enables other researchers to replicate our findings and validate our methodology.

Finally, the hardware used for the training process included an Intel i5-13600k processor, 32GB of RAM, and an RTX 3090 GPU with 24GB VRAM. The system was running Ubuntu 22.04 through the Windows Subsystem for Linux (WSL).

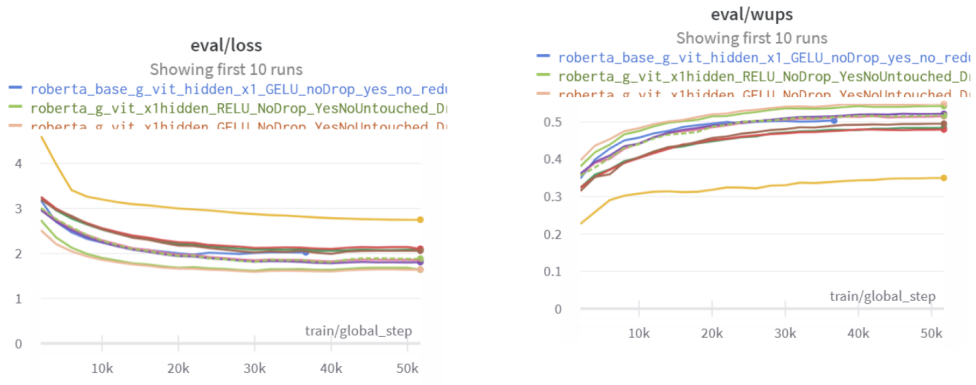


Figure 13: A graphical representation of the validation loss and Wu-Palmer Similarity (WUP) score for a selected subset of our trained VQA models, tracked over the course of five training epochs.

4.4 Hyperparameters Tested

The following are the hyperparameters we varied for evaluation:

- **Class Distribution For yes/no questions:**Untouched vs Reduced
- **Text Encoder:** BERT vs RoBERTa
- **Class Distribution For yes/no questions:**Untouched vs Reduced
- **Linear Classifier Head:** Google ViT vs Microsoft BEiT
  - **Intermediate layer size:** 512 vs hidden_size*1 vs hidden_size*2
  - **Layer norm:** Present vs Absent
  - **Dropout:** Present vs Absent
  - **Activation function:** ReLU vs GELU

**hidden_size = size of the output of text_encoder + image_encoder*

5 Results and Discussion

Bold represents best result within model family; **underline** represents best result across all models.

Model	Parameters	Eval loss	Eval wups	Eval acc	Eval f1	Eval time
Existing Works						
cnn_lstm_baseline (naive baseline)	1.66e+08	2.750	0.350	0.316	0.009	307
ViLT (strong baseline)	1.18e+08	2.371	**0.617**	**0.590**	**0.191**	None*
Fusion Model:Reduced yes/no						
bert_base_beit_x2_hidden	2.1e+08	1.828	0.523	0.492	0.143	193
bert_base_g_vit_x2_hidden	2.1e+08	1.765	0.529	0.497	0.165	**181**
roberta_ms_beit_x2_hidden	2.25e+08	1.773	0.522	0.490	0.154	192
roberta_g_vit_x2_hidden	2.25e+08	**1.722**	**0.530**	**0.498**	**0.173**	182
Fusion Model: Untouched yes/no						
bert_base_g_vit_x2_hidden	2.1e+08	1.957	0.509	0.478	0.126	**183**
bert_base_beit_x2_hidden	2.1e+08	2.062	0.495	0.462	0.084	196
roberta_ms_beit_x2_hidden	2.25e+08	2.268	0.414	0.378	0.032	211
roberta_g_vit_x2_hidden	2.25e+08	1.883	**0.517**	**0.485**	**0.144**	185
bert_base_g_vit_orig_classifier	1.98e+08	2.144	0.479	0.446	0.048	185
roberta_base_g_vit_orig_classifier	2.13e+08	2.085	0.484	0.450	0.053	207
bert_base_g_vit_hidden_x1	2.03e+08	**1.865**	0.514	0.482	0.126	**183**
Fusion Model: roberta_base_g_vit_hidden_x1						
yes_no_untouched	2.18e+08	1.805	0.521	0.489	0.130	186
noDrop_yes_no_untouched	2.18e+08	1.685	0.542	0.511	**0.208**	**182**
GELU_noDrop_yes_no_untouched	2.18e+08	1.644	**0.549**	**0.517**	0.196	**182**
GELU_noDrop_yes_no_reduced	2.18e+08	**1.629**	0.546	0.515	0.198	185

Table 1

**Eval time was not tested for ViLT. Additionally, training and evaluation for ViLT were performed with a batch size of 64 instead of 32.*

5.1 Findings

Text Encoder: Our experiment demonstrated that RoBERTa outperforms BERT as the text encoder for our VQA model, indicating that RoBERTa's architecture and pre-training strategy are better suited for the task.

Image Encoder: In the comparison of image encoders, Google's Vision Transformer (ViT) emerged as the superior choice over Microsoft's BEiT, suggesting that ViT's design and pre-training approach are more effective in extracting and representing visual features for VQA.

Classifier Parameters: The best number of parameters for the intermediate classifier layer was found to be equal to the hidden layer size (hidden*1). This configuration optimizes model capacity without introducing unnecessary complexity.

Classifier Head: Our results indicate that using the GELU activation function without dropout in the classifier head leads to improved performance. This suggests that

GELU offers better non-linearity for this task, while dropout may not be necessary for regularization.

Class Distribution for Yes/No Questions: The findings regarding the optimal distribution of yes/no questions in the dataset were inconclusive. Generally, most models trained on a reduced yes/no dataset showed better performance. However, the best-performing model was trained on the untouched yes/no dataset. This observation calls for further investigation into the impact of class distribution on model performance.

Our best model hence uses the following hyperparameters:

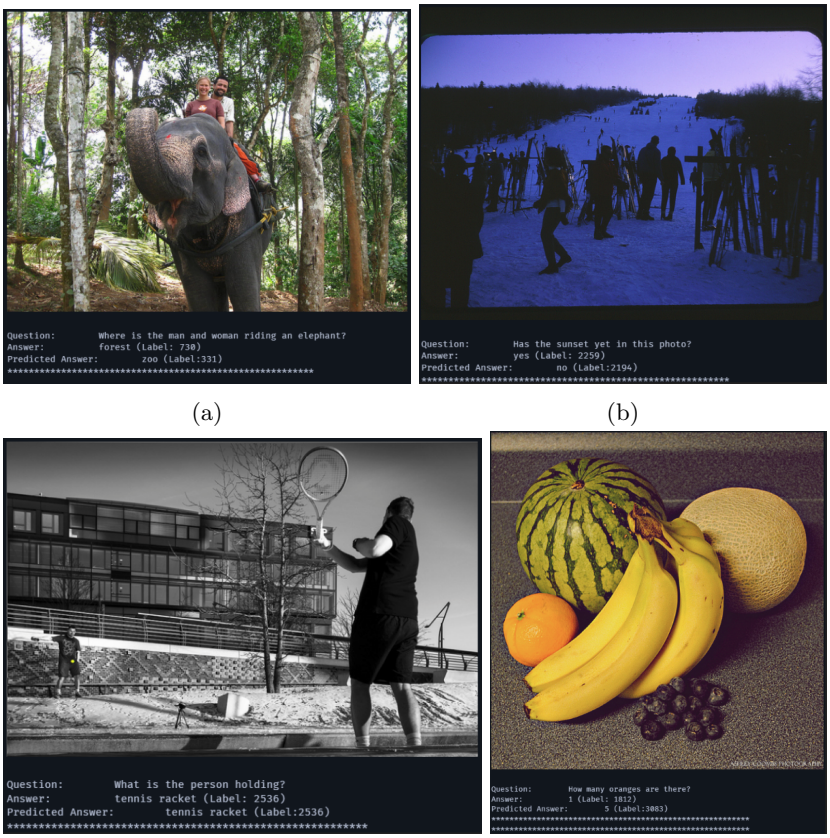
- **Class Distribution For yes/no questions:** Untouched
- **Text Encoder:** RoBERTa
- **Class Distribution For yes/no questions:** Google ViT
- **Linear Classifier Head:**
  - **Intermediate layer size:** Hidden*1
  - **Regularisation:** No dropout
  - **Activation function:** GELU

5.2 Findings

CNN + LSTM (Naive Baseline): Our best transformer-based fusion model achieved a WUPs score of 0.549, predictably surpassing the naive CNN + LSTM baseline, which had a WUPs score of 0.350. This demonstrates the superiority of the transformer-based fusion approach for the VQA task when compared to more basic architectures.

ViLT (Strong Baseline): Our model was unable to outperform the strong ViLT baseline, which achieved a higher WUPs score of 0.617 and demonstrated greater parameter efficiency. The reason for this difference in performance can be attributed to the well-tuned architecture of the released ViLT model by the authors, which is capable of directing computation towards modality interactions rather than embedding computations. ViLT's architecture features shallow and computationally light embedding layers for both raw pixels and text tokens, thereby concentrating most of the computation on modelling modality interactions.

Hence, while our transformer-based fusion model demonstrates improved performance over the naive CNN + LSTM baseline, it falls short of surpassing the state-of-the-art ViLT model, highlighting the importance of efficient architecture design and the focus on modality interactions in the VQA task.



(a) (b)

(c) (d)

Figure 14: Examples of our top-performing fusion model's predictions, highlighting its capacity to generate reasonable responses that are semantically similar to the ground truth answers

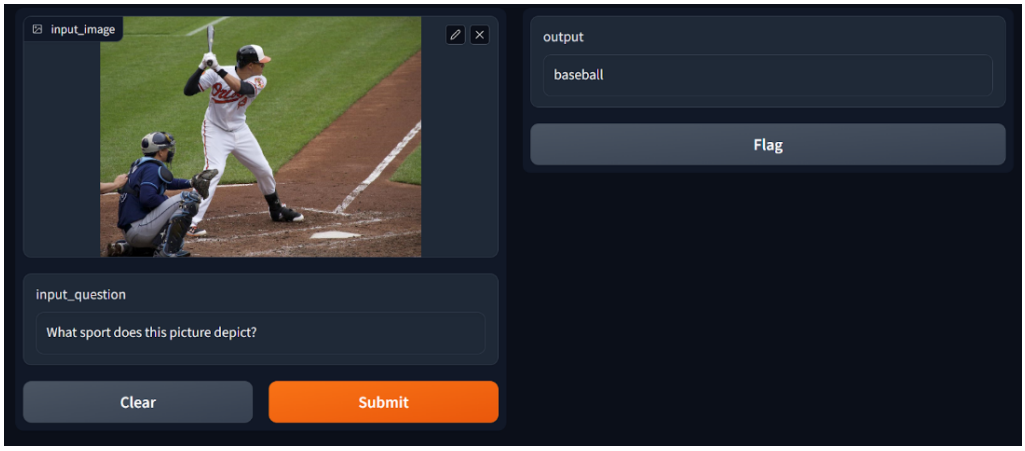


Figure 15: A user-friendly graphical user interface (GUI) for our Visual Question Answering (VQA) late-fusion model, developed using the Gradio framework.

6 Challenges and Limitations

6.1 Inherent difficulty of the task of VQA

The task of VQA is inherently challenging due to a variety of factors:

- **Abstract concepts:** Questions may involve abstract concepts (e.g., "end well") that are difficult for computers to interpret and answer accurately.
- **Contextual understanding:** Some questions require an understanding of context or implications arising from the current situation, which can be challenging for AI models.
- **Temporal reasoning:** Questions may necessitate predicting future outcomes, demanding temporal reasoning capabilities that are difficult for AI models to achieve.
- **Ambiguity:** VQA often deals with inherent subjectivity, even among humans, which complicates the task.
- **Compositionality:** The complex relationships between image and question elements can be challenging for AI models to decipher.

Figure 16: Example of a challenging question for VQA models, demonstrating the inherent complexities and difficulties faced by these models in certain situations.

6.2 Limited Time and Resources

Due to time constraints and lack of access to powerful GPUs, it was difficult to perform extensive experiments and hyperparameter tuning with the full dataset. This also limited our ability to explore larger and more computationally demanding models.

6.3 AI Alignment

Ethical concerns arise in VQA research, necessitating careful consideration of AI alignment. For example, there may be a need to filter out examples due to potential ethical concerns, such as instances of racism, to ensure that the AI model does not inadvertently propagate harmful content or biases.

7 Future Directions and Conclusion

While our model demonstrates promising results, there remains significant room for improvement, and more advanced model structures are yet to be explored. To achieve

meaningful, non-incremental advancements in the future, we suggest the following directions:

- Increase the size of the model and dataset by several orders of magnitude, following the brute-force approach employed by large-scale language models (LLMs) and ChatGPT. This strategy can help enhance the model's performance and enable it to tackle more complex VQA tasks.
- Develop and implement better ways to represent and ingest data used in VQA, focusing on the incorporation of abstract concepts, reasoning capabilities, and human values. By improving data representation and ingestion, future models may be better equipped to handle the inherent challenges of VQA tasks.
- Explore ways to emphasize multi-modal interactions in models, as espoused by the ViLT paper

In conclusion, we present the development and training of a late-fusion model capable of providing reasonable answers to most questions while maintaining a reasonable parameter count. We conducted a thorough investigation of various hyperparameters, such as encoder choice and classifier structure, and compared our model's performance with existing approaches in the field.