Visual Question Answering using OpenAI's CLIP and linear layers

Khalid Osama Ziad Fahmy

August 2024

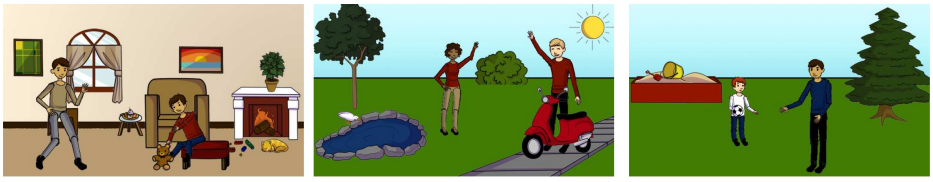
Abstract

The task of Visual Question Answering (VQA) is a challenging one as it requires models to integrate and reason across images and text. Many of the current architectures for VQA suffer from high complexity. As a result, these architectures are diffcult to train and require high computational resources. To address these problems a CLIP-based architecture that does not require any fine-tuning of the feature extractors, based on the previous work by Deuser et al., 2022, is introduced as our baseline architecture. A simple linear classifer is used on the concatenated features of the image and text encoder. During training an auxiliary loss is added which operates on the answer types. The resulting classifcation is then used as an attention gate on the answer class selection. Operating on the VizWiz 2023 dataset designed for visually impaired individuals, we achieve –% accuracy on the task of predicting the answer to a visual question.

1 Introduction

1.1 Problem Statement

Visual Question Answering is the task of answering open-ended —and close-ended— questions based on an image. They output natural language responses to natural language questions. This technology has significant applications in fields such as healthcare, education, and surveillance, where understanding and interpreting visual information is crucial. A key challenge in VQA lies in developing models that "can accurately interpret both the visual content of an image and the linguistic subtleties of a question, even when faced with real-world imperfections like unclear questions or challenging image conditions.



(a) **Q**: What is the boy playing with? **A**: teddy bear

(b) **Q**: Are there any animals swimming in the pond? **A**: No

(c) **Q**: How many trees? **A**: 1

Figure 1: VQA examples

1.2 Exploring the dataset

The VizWiz dataset was originally designed to aid visually impaired users. It includes real-world images and questions captured through the VizWiz app, presenting challenges like poor image quality and ambiguous content. We choose the updated version of January 10, 2023 to work with, which has the following structure: **image name, a question, 10 answer/answer-confidence pairs for the question, answer type, and whether the question is answerable or not**, as shown in Table 1.

Table 1: Sample from the training set

image	question	answers	answer type	answerable
VizWiz_train_0.jpg	What is this?	{'answer':'basil', 'answer_confidence': 'yes'...}	other	True/1

The dataset is already split with the training set consisting of 20,523 examples, validation set of 4319 examples, and test set of 8000 examples.

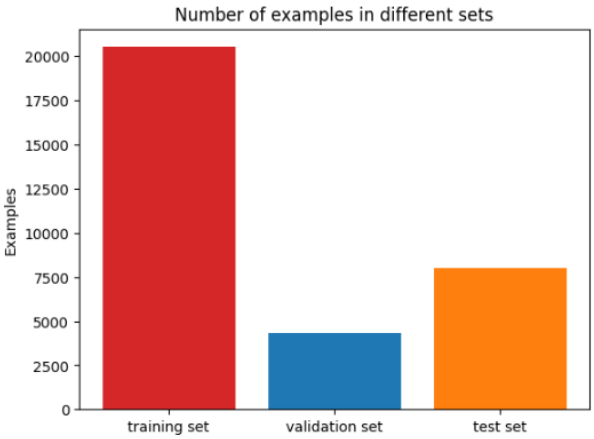


Figure 2: Number of examples in the different sets.

The answer types of the training and validation sets have the following distributions, shown in Figure. 3, with 'other' being the most common type of answers.

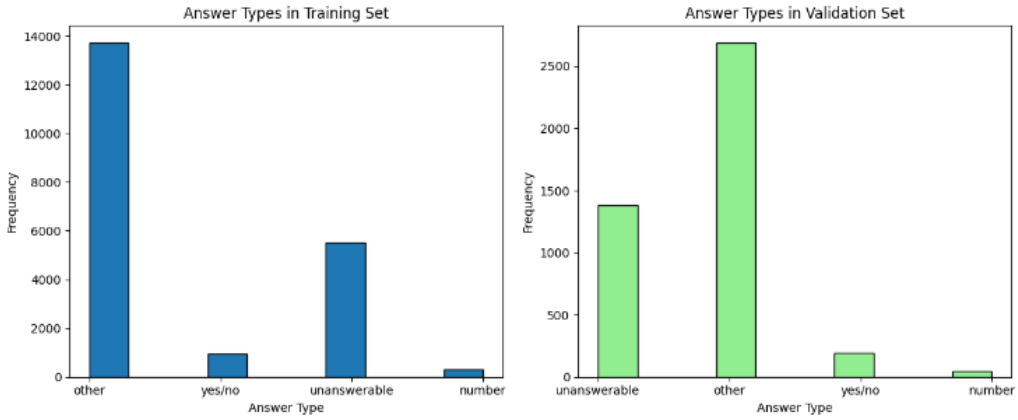


Figure 3: Answer Types Distribution

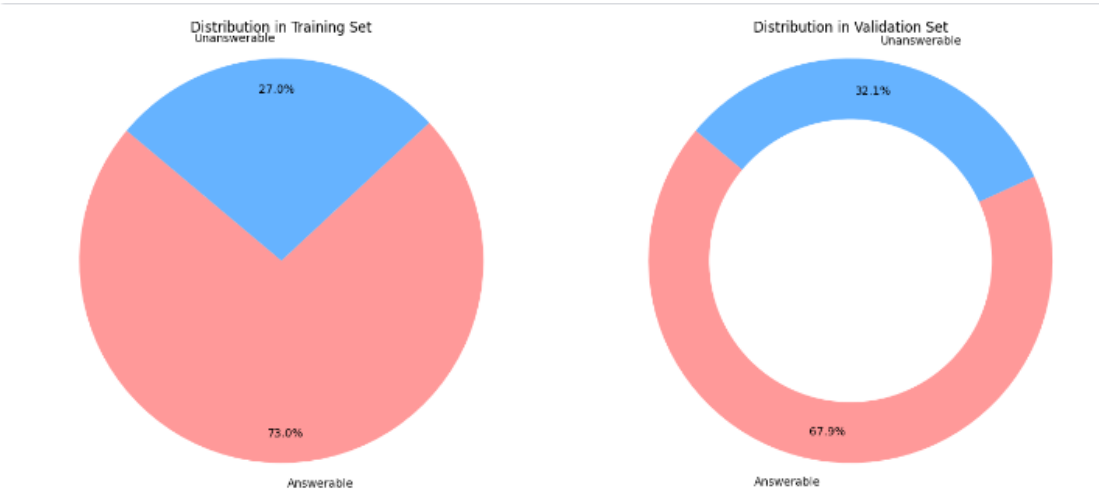


Figure 4: Distribution of answerable and unanswerable questions

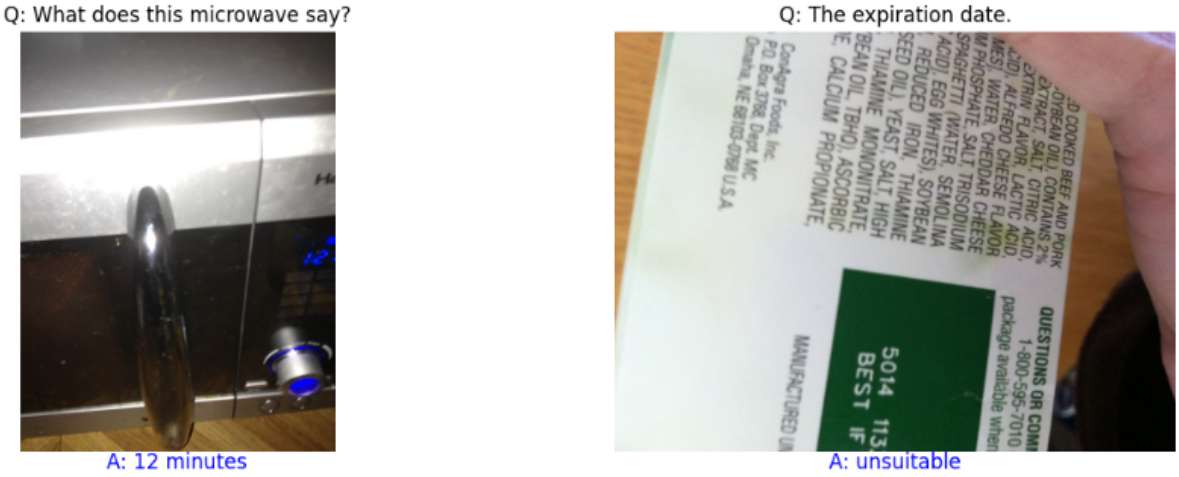


Figure 5: Two visual examples from the training set

2 Methodology

2.1 Model backbone

As our model heavily relies on CLIP, it is convenient to introduce how it works. CLIP (Contrastive Language–Image Pre-training) is a powerful model that connects images and text. CLIP can understand visual concepts and match them with descriptive language as it has been trained on a huge dataset of image-text pairs. This enables it to perform tasks like zero-shot image classification, where it can recognize objects without specific training on those categories.

CLIP takes images and text as inputs, generating feature vectors for each. It processes images through a vision model to produce image embeddings and text through a transformer to produce text embeddings. It then matches these embeddings by comparing their cosine similarities, enabling tasks like zero-shot image classification and retrieval by linking visual and textual information.

The contribution of this paper is divided into (i) creating a suitable vocabulary for the classification task, (ii) using CLIP features with linear layers for VQA.

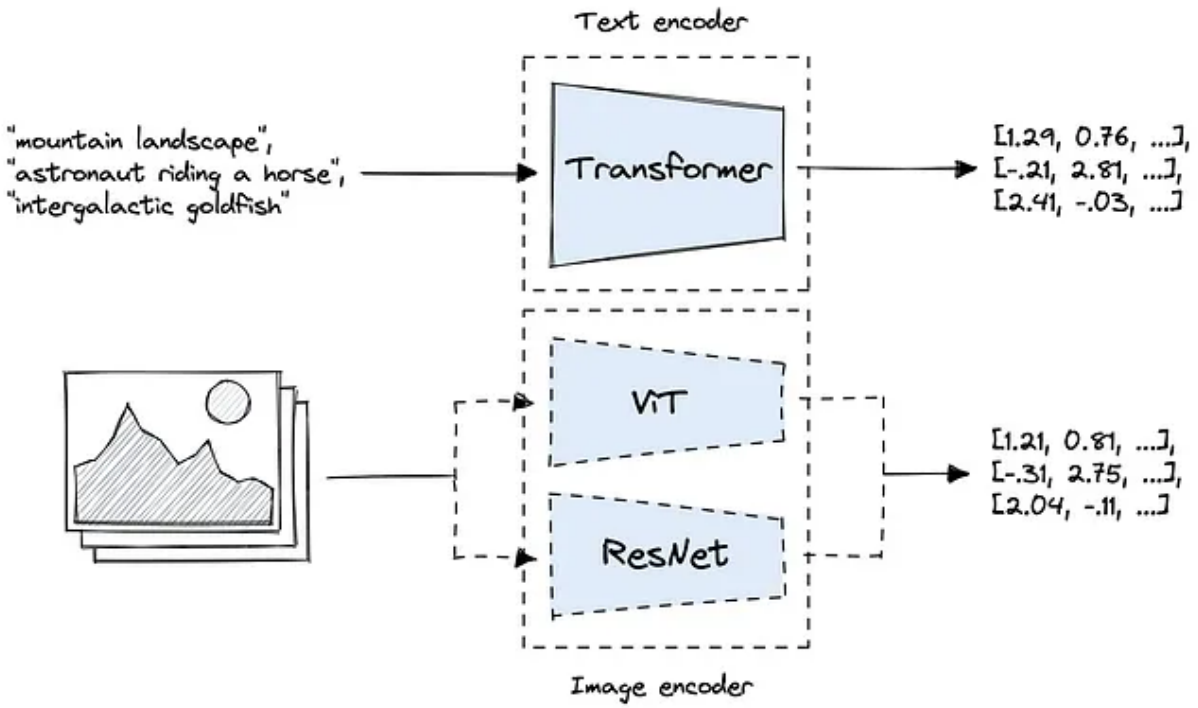


Figure 6: CLIP Architecture

2.2 Answer Vocabulary Generation

The vocabulary building process is designed to identify the most representative answer for each question from a set of 10 possible answers and to build a refined answer vocabulary. Initially, the answers with the highest confidence levels are prioritized. The selection process follows a structured policy to ensure the most relevant answer is chosen.

First, among the set of answers, the one with the highest frequency is selected. If this initial selection results in a tie, the process moves to the next step, where the answer with the highest frequency across the entire dataset is chosen. This step ensures that, if multiple answers are equally frequent within a specific question, the one that appears most often across all questions is selected.

If a tie persists even after considering the overall frequency, the Levenshtein distance is used to break the tie. This involves calculating the total Levenshtein distance between each tied answer and the others. The answer with the smallest total distance, indicating it is the most similar to the other tied answers, is chosen.

This thorough approach ensures that the final answer vocabulary is both representative and comprehensive. The process results in a curated vocabulary of 5731 distinct answers, effectively balancing frequency and relevance for each question in the VizWiz dataset.

2.3 Auxiliary Loss

The "Answer Type Gate" auxiliary loss improves the VQA model by introducing a mechanism that masks irrelevant answers based on their type. The model first identifies the answer type, such as "numbers," "yes/no," "others," or "unanswerable," by analyzing the most suitable answer for each image-question pair. This answer type is then used to predict a probability distribution through a linear projection, with the size of the vector matching the answer vocabulary (5731). Afterward, a sigmoid layer generates probabilities for each answer type, which are multiplied with the answer logits. This element-wise multiplication effectively masks out answers that don't match the identified type, allowing

the model to focus on relevant responses during inference.

Both the intermediate answer type prediction and the final answer classification are key components of the overall loss function, with two equally weighted cross-entropy losses driving the training process. By learning both tasks simultaneously, the model becomes more proficient at recognizing the appropriate type of answer and generating the final answer itself. This dual-task learning enables the model to make more accurate, contextually relevant predictions. The Answer Type Gate ultimately enhances the VQA model's ability to handle diverse types of answers, leading to better performance by filtering out irrelevant answers and refining the decision-making process.

2.4 Architecture

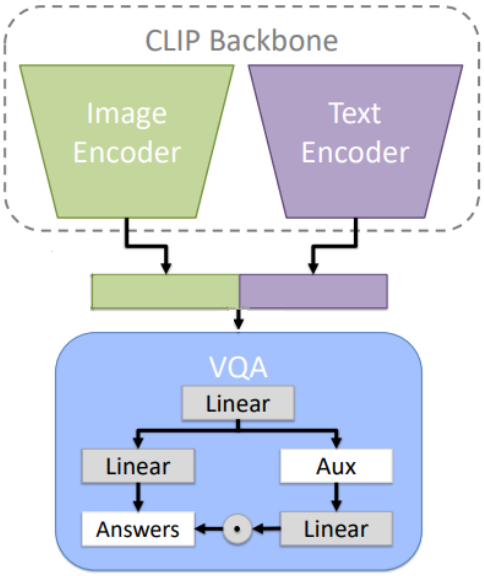


Figure 7: Proposed Architecture

Our architecture is mainly based on the (Deuser et al., 2022) paper. The training images are fed to CLIP along with the already prepared vocabulary. The features extracted from these encoders are concatenated and passed through linear layers. The architecture can be explained as follows:

- **CLIP Backbone:**
  - **Image Encoder:** The image encoder takes the raw image as input and processes it through a pre-trained vision model from the CLIP architecture (e.g., a Vision Transformer or CNN). The output is an image feature embedding that represents the visual content in a high-dimensional space.
  - **Text Encoder:** The text encoder processes the textual input (typically a question) using a pre-trained language model from CLIP (based on a transformer). It transforms the question into a text embedding that captures the semantic meaning of the input.
  - **Feature Concatenation:** After obtaining embeddings from both the image and text encoders, they are concatenated into a single feature vector, capturing the multi-modal information necessary for the downstream VQA task.
- **VQA Module:**

- **Linear Layer:** The concatenated features (from the image and text encoders) are passed through a fully connected linear layer, transforming the joint representation into a feature space for answer prediction and auxiliary tasks.
- **Answer Prediction Path:**
  - The features from the linear layer are passed through additional linear transformations to predict the possible answers based on the answer vocabulary.
  - The model compares the predicted answers with the ground truth during training, minimizing classification error using cross-entropy loss.
- **Auxiliary Task - Answer Type Prediction:**
  - This branch predicts the answer type (e.g., "yes/no," "numbers") for each image-question pair.
  - The predicted answer type is used to mask out irrelevant answers during inference, focusing on answers corresponding to the predicted type.

- **Answer Type Masking and Sigmoid:**
  - The answer type prediction is projected into a vector with the same dimension as the number of possible answer classes.
  - After applying a sigmoid layer, this vector is multiplied with the logits of the answer vocabulary to mask out irrelevant answers during inference.

- **Final Output and Loss:**
  - The model combines the outputs from both the answer prediction and auxiliary task paths.
  - Both cross-entropy losses, from the answer type prediction and final answer classification, are computed and equally weighted during training.

2.5 Loss Function

The linear classifier is trained using the **cross-entropy loss** function. This approach measures the performance of the classifier by comparing the predicted probabilities of each class to the true class labels. The cross-entropy loss quantifies the discrepancy between the predicted and actual values, guiding the model's learning process to minimize this error and improve classification accuracy.

3 Challenges and Setbacks

During the project, several setbacks arose, requiring adjustments to the model as progress continued. These challenges, though initially disruptive, led to improvements in the model's performance and overall project direction.

3.1 Initial model

In our initial attempt to build the model using PyTorch, storage allocation issues with the DataLoader forced us to train on a limited subset of the data, impacting the model's early performance. Although the model eventually achieved an acceptable loss, it struggled to generalize effectively.

Figure 8: Train and Validation loss for the initial model

The initial model's inability to generalize effectively can be clearly seen when given these three pictures along with the question "What is this?"

Figure 9: Running the model on samples from the train set

3.2 Improved model

This model is designed to process two inputs: an image encoding and a question encoding, each with a shape of $(1, 512)$. These inputs are concatenated into a combined feature vector of shape $(1, 1024)$. This combined vector is then passed through a batch normalization layer to stabilize the training process, followed by a dropout layer with a 50% dropout rate to prevent overfitting.

The model handles two tasks: answerability prediction and answer prediction. For answerability prediction, the normalized and dropped-out combined features are passed through a dense layer with 1 unit and a sigmoid activation function, which outputs a probability score between 0 and 1, indicating whether the visual question is answerable. This result is reshaped into a 1D tensor to match the expected output format.

For answer prediction, the model includes an auxiliary layer, which is a dense layer with 4 units and a softmax activation. This layer predicts the type of answer, which is

important for applying an answer mask. The output of this auxiliary layer is reshaped into a vector of size 4 to generate answer types. Next, a dense layer generates answer gates by projecting the answer types into a space of 5731 possible answers (the full vocabulary of potential answers).

Simultaneously, the model predicts unmasked answer logits using another dense layer with 5731 units and a softmax activation. These logits represent the likelihood of each possible answer being correct. To refine the answer prediction, the model uses a multiplication layer to combine the unmasked answer logits with the answer gates, applying the predicted answer type to mask out irrelevant answers.

Finally, the resulting masked answers are reshaped through the main output layer, producing the final answer output with shape $(None, 5731)$, representing the likelihood distribution over all possible answers. Thus, the model predicts both the answerability of the question and the specific answer to the visual question by combining image and question encodings, auxiliary answer types, and masked answer logits. The model summary can be seen in the figure below.

Layer (type)	Output Shape	Param #	Connected to
image_encoding (InputLayer)	(None, 1, 512)	0	-
question_encoding (InputLayer)	(None, 1, 512)	0	-
concatenate (Concatenate)	(None, 1, 1024)	0	image_encoding[0… question_encodin…
batch_normalizatio… (BatchNormalizatio…	(None, 1, 1024)	4,096	concatenate[0][0]
dropout_9 (Dropout)	(None, 1, 1024)	0	batch_normalizat…
aux_layer (Dense)	(None, 1, 4)	4,100	dropout_9[0][0]
unmasked_answer_li… (Dense)	(None, 1, 5731)	5,874,275	dropout_9[0][0]
answer_types_proj (Dense)	(None, 1, 5731)	28,655	aux_layer[0][0]
masked_answer (Multiply)	(None, 1, 5731)	0	unmasked_answer_… answer_types_pro…
answerability_outp… (Dense)	(None, 1, 1)	1,025	dropout_9[0][0]
main_output (Reshape)	(None, 5731)	0	masked_answer[0]…
aux_output (Reshape)	(None, 4)	0	aux_layer[0][0]
answerability_outp… (Reshape)	(None, 1)	0	answerability_ou…

Figure 10: Model summary using Tensorflow

4 Evaluation metrics

In addition to accuracy, our evaluation introduces a metric called VizWiz accuracy, which was introduced by the official VizWiz website. This metric reflects not only the correctness but also the context and relevance of the answers provided. It is calculated as follows:

$$\text{VizWiz accuracy}(\text{answer}) = \min\left(\frac{\text{number of humans that provided that answer}}{3}, 1\right)$$

This means if the predicted answer matches at least 3 human-provided answers, you get full accuracy (1.0) for that question. Otherwise, the score is proportional to how many of the 10 answers it maches, capped at 1.0.

We also introduce a metric for answerability prediction, which measures the model's ability to predict whether a question is answerable, based on a confidence score ranging from 0 to 1.

5 Results

The model was trained for a total of 50 epochs on Kaggle using the P100 accelerator. We used Adam as the optimizer and a batch size of 16. The model's performance can be seen in the following figures.

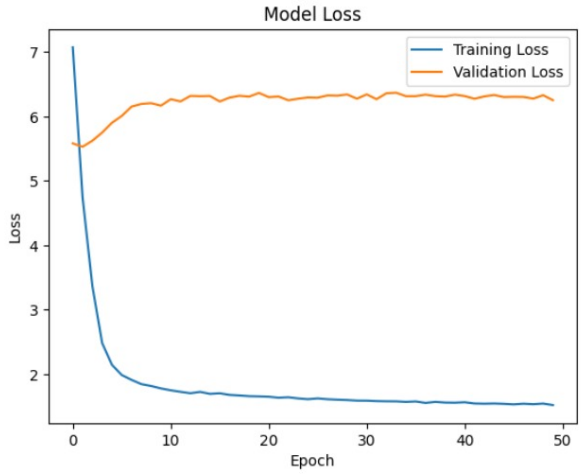


Figure 11: Train and validation loss

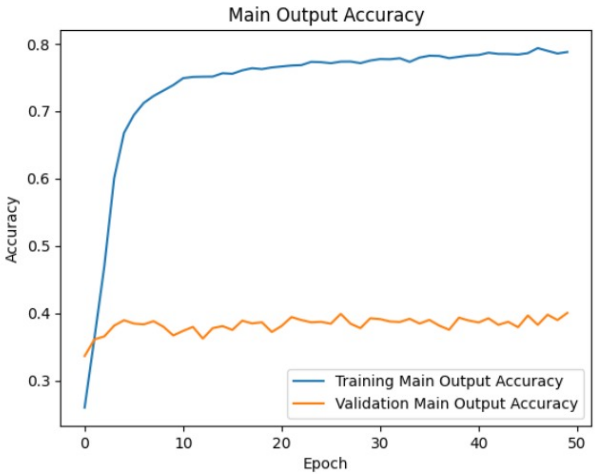


Figure 12: Main output (answer) accuracy

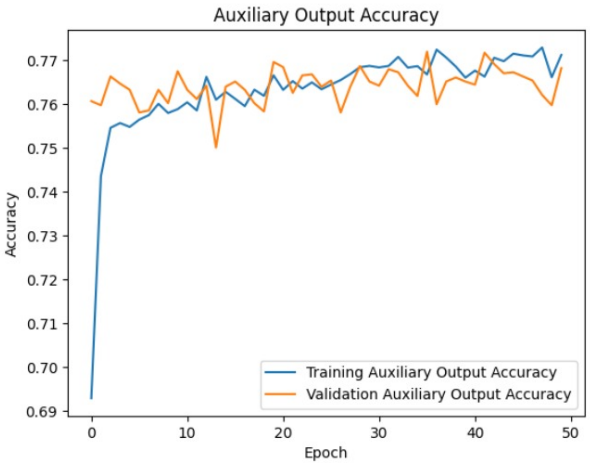


Figure 13: Auxiliary output (answer type) accuracy

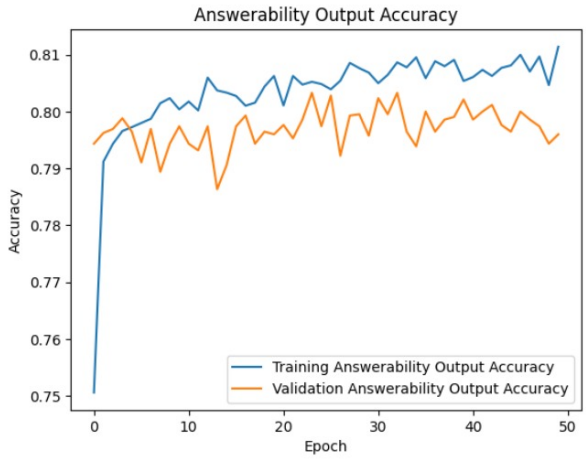


Figure 14: Answerability accuracy

The results table (Table 2) presents the performance of a VQA model on various metrics. The metrics include VizWiz accuracy, accuracy, and answerability. For VizWiz accuracy, the model achieved a training accuracy of 86.52% and a validation accuracy of 52.18%. This metric indicates how well the model performs on the VizWiz dataset, which is a real-world image dataset containing challenging visual question answering scenarios. The accuracy metric measures the overall correctness of the model's predictions. The training accuracy achieved by the model is 85.35%, while the validation accuracy is 40.02%. This

Metric	Training	Validation
VizWiz Accuracy	86.52	52.18
Accuracy	85.35	40.02
Answerability	82.9	79.6

Table 2: Results for different metrics

6 Samples

We present a selection of examples from the validation and test datasets. By showcasing 10 examples from each dataset, we aim to provide a comprehensive view of the data that our model has been evaluated on.

6.1 Validation Set Examples

question: Can you tell me what this medicine is please?
predicted answer: night time
expected answer: unsuitable image
predicted answer type: other

question: Which one is the blue one?
predicted answer: right
expected answer: right
predicted answer type: other

question: What color is this
predicted answer: white
expected answer: white
predicted answer type: other

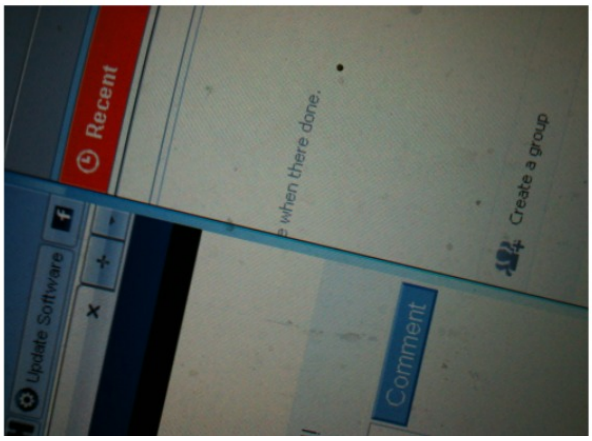
question: Which one of these items is the children's dictionary? Is it the one on the right, or the one on the left?
predicted answer: right
expected answer: left
predicted answer type: other

question: What is the computer doing?
predicted answer: laptop
expected answer: unanswerable
predicted answer type: other

question: I was wondering how ominous the sky is looking here at the northwest.
predicted answer: yes
expected answer: unanswerable
predicted answer type: yes/no

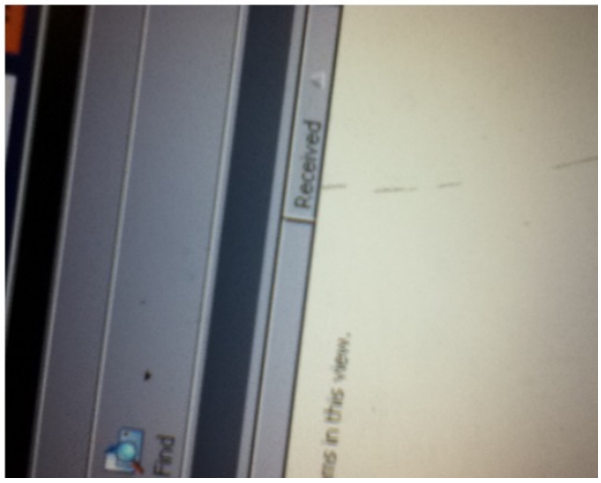
question: What store is this?
predicted answer: original pancake house
expected answer: original pancake house
predicted answer type: other

question: Can you tell me what this is a picture of?
predicted answer: computer screen
expected answer: unanswerable
predicted answer type: other



question: What's the name of the drink?
predicted answer: water
expected answer: irn bru
predicted answer type: other

question: What is this?
predicted answer: keyboard
expected answer: unanswerable
predicted answer type: other



6.2 Test set examples

It should be noted that the test set is unlabeled, meaning that no answers are publicly available for it in the dataset.

```
question: What is this? And what color is it?
predicted answer: black
predicted answer type: other
```

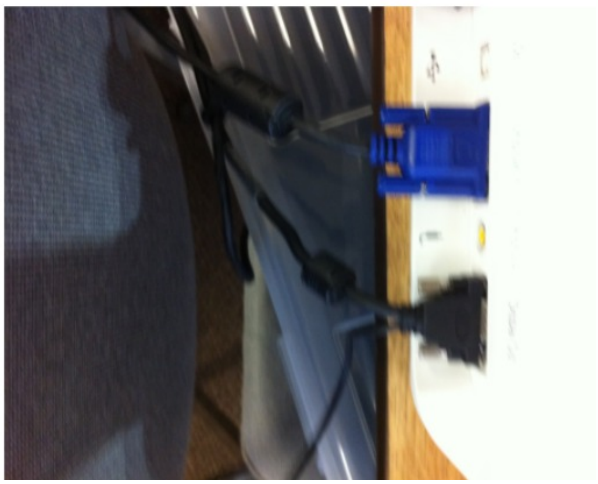
```
question: What is this?
predicted answer: laptop
predicted answer type: other
```

```
question: What is this?
predicted answer: half half
predicted answer type: other
```

```
question: Do these beans look like black beans or pinto beans?
predicted answer: yes
predicted answer type: other
```

question: what is this?
predicted answer: stove
predicted answer type: other

question: What color are these two connectors?
predicted answer: black blue
predicted answer type: other



question: Which cereal is this? Question mark.
predicted answer: cheerios
predicted answer type: other

question: What room is this?
predicted answer: 216
predicted answer type: other

question: What kind of coffee is this?
predicted answer: vanilla hazelnut
predicted answer type: other

question: What is this?
predicted answer: keyboard
predicted answer type: other