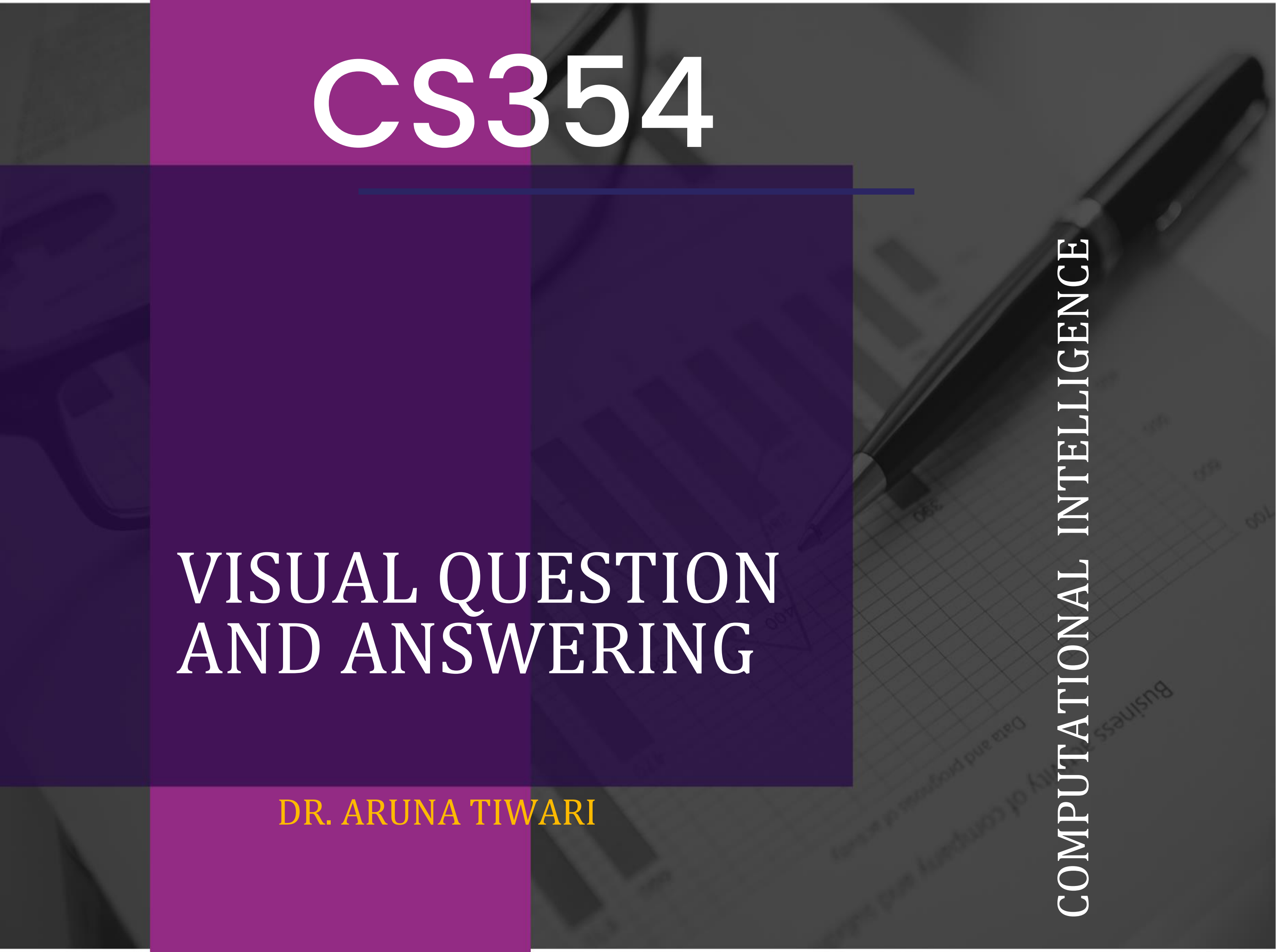


CS354

VISUAL QUESTION AND ANSWERING

DR. ARUNA TIWARI

COMPUTATIONAL INTELLIGENCE

M SRAVANTHI(210001042)
S PARIMALA (210001064)
M DIVYATEJA(210001040)

PROBLEM STATEMENT

Given a dataset consisting of images paired with corresponding questions and answers, the task is to develop a machine learning model capable of performing prompt-based visual question answering. The goal of the model is to accurately answer questions about unseen images based on the provided questions and answers during training.

DATA COLLECTION

We used the following dataset,

https://www.kaggle.com/datasets/lhanhsin/vizwiz/data

- 20,523 training image/question pairs
- 205,230 training answer/answer confidence pairs
- 4,319 validation image/question pairs
- 43,190 validation answer/answer confidence pairs
- 8,000 test image/question pairs

Dataset files to download are as follows:

- Images: training, validation, and test sets
- Annotations and example code:
  - Visual questions are split into three JSON files: train, validation, and test. Answers are publicly shared for

the train and validation splits and hidden for the test split.

- APIs are provided to demonstrate how to parse the JSON files and evaluate methods against the ground truth.

PREPROCESSING

Image

For image preprocessing, the initial step involves resizing the image to a square size of 336x336 pixels using bicubic interpolation to maintain quality and proportion. This is followed by a central crop to extract a 336x336 region from the center of the image, ensuring key content retention. If the image isn't already in RGB format, it's converted to RGB. Then, the image is transformed into a tensor for neural network processing. Finally, normalization adjusts pixel values based on dataset mean and standard deviation to stabilize training.

Text

For text preprocessing, it's typically processed as a sequence of tokens. After this step, the text is transformed into a tensor. This transformation involves encoding the text into numerical representations, such as indices or embeddings

Labels

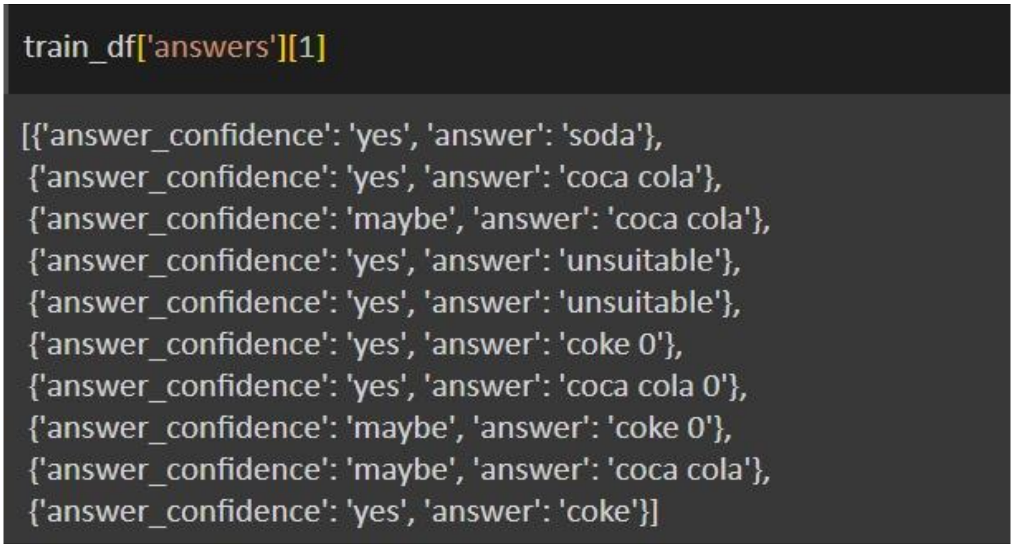
The frequency of answers in the overall dataset was determined by counting the occurrences of each unique answer.

It builds the answer vocabulary based on the most frequent answer for each question, handling tie-breaking using Levenshtein distance.

One-hot encoding is applied to answers and answer types using OneHotEncoder.

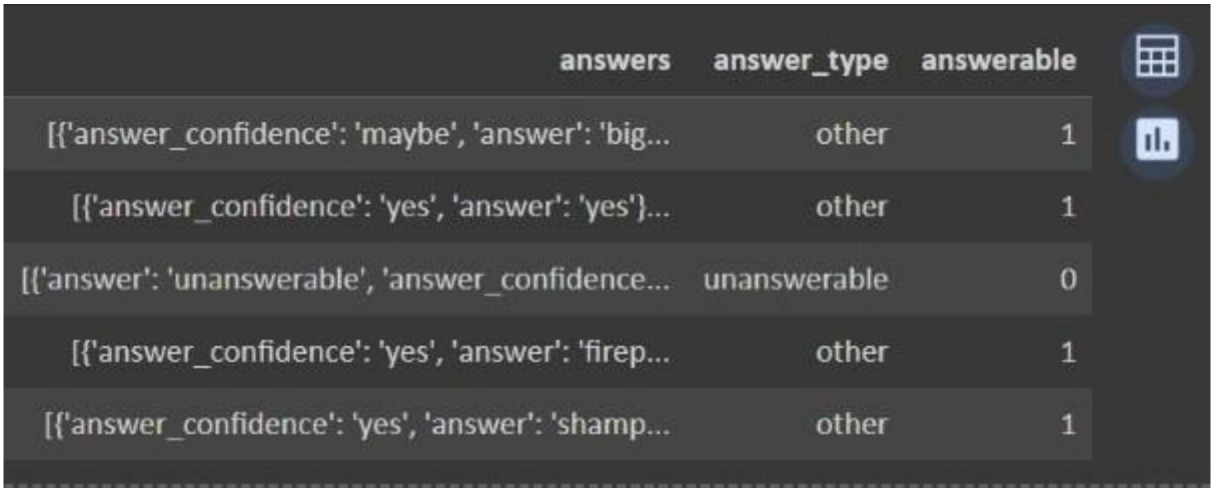
It saves image and question embeddings, as well as a binary indicator of whether each question is answerable or not.

Final Dataset contains individual samples, including image and question features, one-hot encoded answers and answer types, answer counters, and answerability labels.



```
train_df['answers'][1]
```

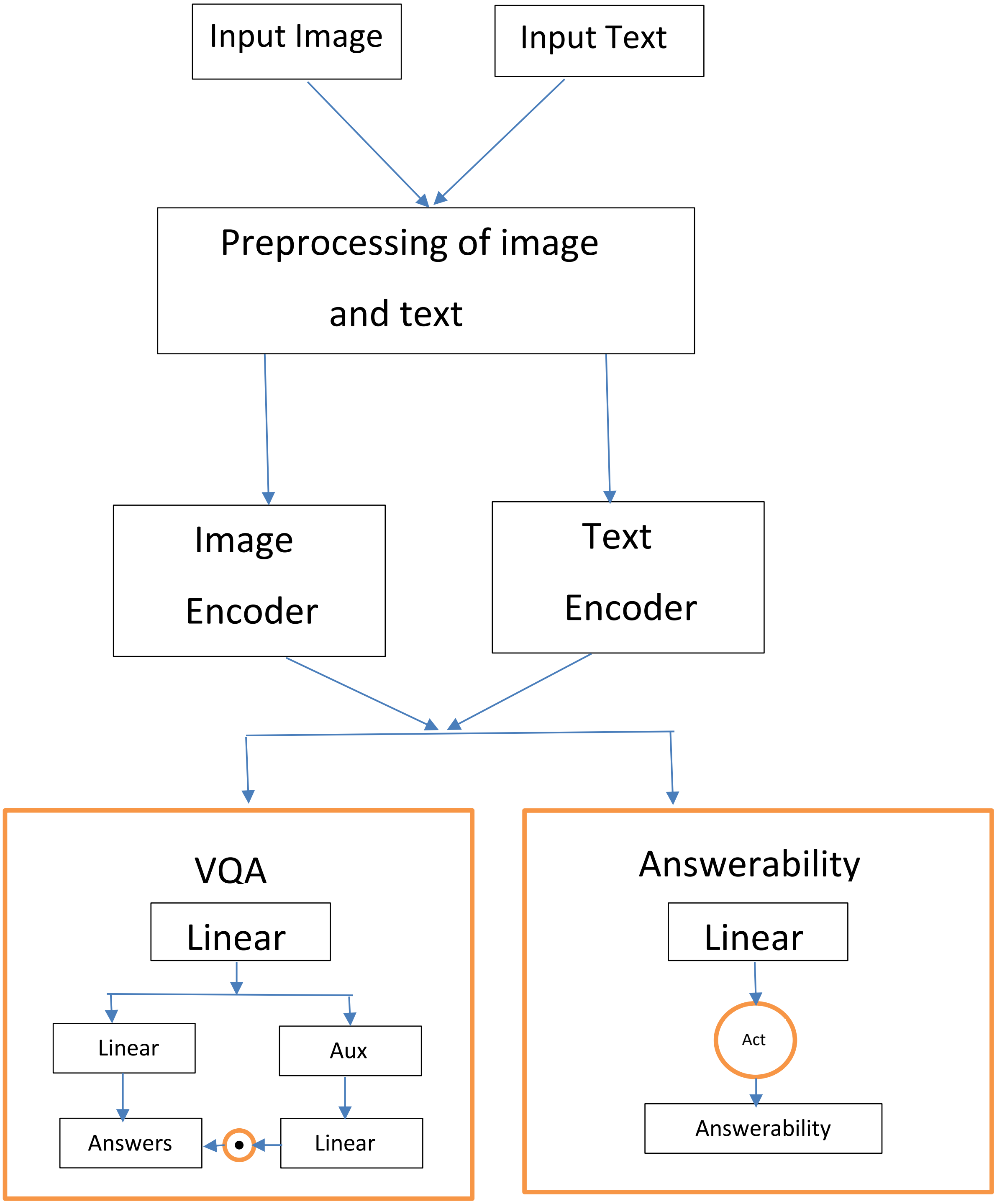
```
[{'answer_confidence': 'yes', 'answer': 'soda'},
 {'answer_confidence': 'yes', 'answer': 'coca cola'},
 {'answer_confidence': 'maybe', 'answer': 'coca cola'},
 {'answer_confidence': 'yes', 'answer': 'unsuitable'},
 {'answer_confidence': 'yes', 'answer': 'unsuitable'},
 {'answer_confidence': 'yes', 'answer': 'coke 0'},
 {'answer_confidence': 'yes', 'answer': 'coca cola 0'},
 {'answer_confidence': 'maybe', 'answer': 'coke 0'},
 {'answer_confidence': 'maybe', 'answer': 'coca cola'},
 {'answer_confidence': 'yes', 'answer': 'coke'}]
```



answers	answer_type	answerable
[{'answer_confidence': 'maybe', 'answer': 'big...	other	1
[{'answer_confidence': 'yes', 'answer': 'yes'}...	other	1
[{'answer': 'unanswerable', 'answer_confidence...	unanswerable	0
[{'answer_confidence': 'yes', 'answer': 'firep...	other	1
[{'answer_confidence': 'yes', 'answer': 'shamp...	other	1

ANALYSIS

Requires a deep understanding of both computer vision and natural language processing. The inputs to the model are images paired with corresponding questions, while the output is the predicted answer for each question. The model needs to learn the associations between images, questions, and answers from the training data. This includes feature extraction from images, text processing for questions, and learning the mapping between inputs and outputs.

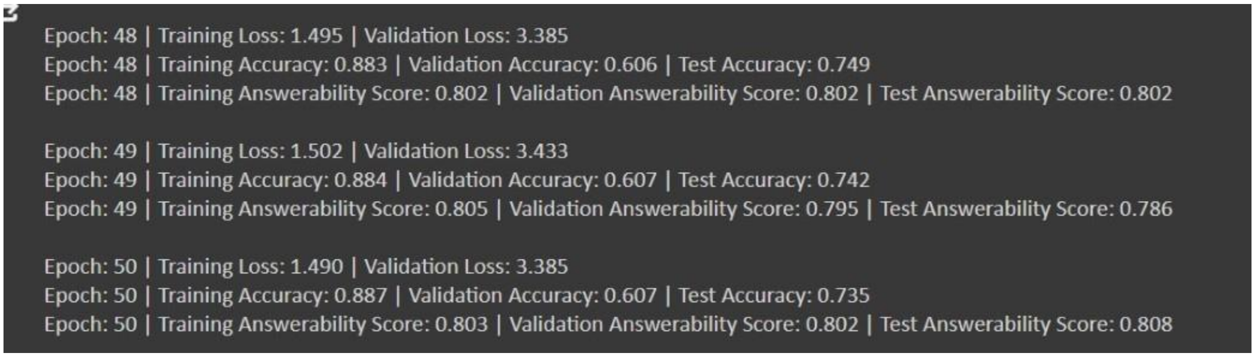

Input Image
Input Text
Preprocessing of image and text
Image Encoder
Text Encoder
VQA
Linear
Linear
Aux
Answers
Linear
Answerability
Linear
Act
Answerability

ALGORITHM

1. Image and text(question) are given as input. As images are not of same size, they are reshaped to (3 x 336 x 336).After preprocessing the text, it converts to (1 x 77)
   Sequence length=77
2. Encoder:
   - A sequence of residual attention blocks within a transformer architecture.
   - Each residual attention block includes a multi-head attention mechanism followed by a feedforward neural network (MLP) with layer normalization and GELU() activation.
   - This block utilizes layer normalization both before and after the attention. Additionally, the output is normalized again at the end of the block.
3. Image Encoder consists of a CNN layer which converts 3 channels to 1024 followed by 23 Encoder blocks. Outputs from this are image features of dimension 1024.
4. Text Encoder consists of a token embedding layer, followed by 11 Encoder blocks. Outputs from this are text features of dimension 768.
5. Features of image and text are concatenated and passed into two linear layers followed by additional layers for handling answer types and answerability.

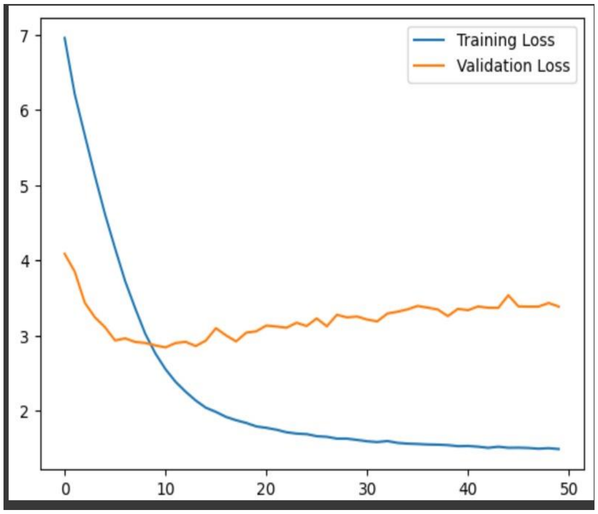
6. The first linear layer incorporates Layer Normalization and Dropout, accepting concatenated image and question features as input and outputting to a specified hidden size.
7. Another linear layer handles answer types prediction, followed by a sigmoid activation for answer masking.
8. An answerability linear layer with Layer Normalization precedes a final linear layer with sigmoid activation for answerability prediction.
9. Features from first linear layer passing through the second linear layer, the output is modulated by the answer mask, providing the final output along with answer types and answerability score.

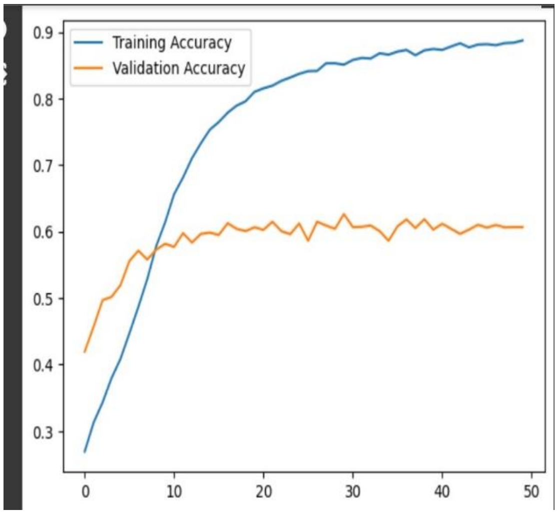
RESULTS

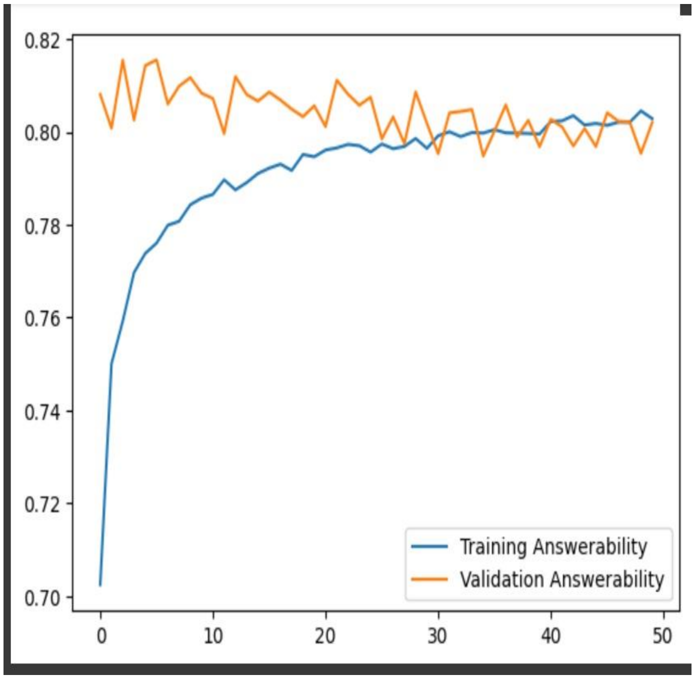


```
Epoch: 48 | Training Loss: 1.495 | Validation Loss: 3.385
Epoch: 48 | Training Accuracy: 0.883 | Validation Accuracy: 0.606 | Test Accuracy: 0.749
Epoch: 48 | Training Answerability Score: 0.802 | Validation Answerability Score: 0.802 | Test Answerability Score: 0.802

Epoch: 49 | Training Loss: 1.502 | Validation Loss: 3.433
Epoch: 49 | Training Accuracy: 0.884 | Validation Accuracy: 0.607 | Test Accuracy: 0.742
Epoch: 49 | Training Answerability Score: 0.805 | Validation Answerability Score: 0.795 | Test Answerability Score: 0.786

Epoch: 50 | Training Loss: 1.490 | Validation Loss: 3.385
Epoch: 50 | Training Accuracy: 0.887 | Validation Accuracy: 0.607 | Test Accuracy: 0.735
Epoch: 50 | Training Answerability Score: 0.803 | Validation Answerability Score: 0.802 | Test Answerability Score: 0.808
```

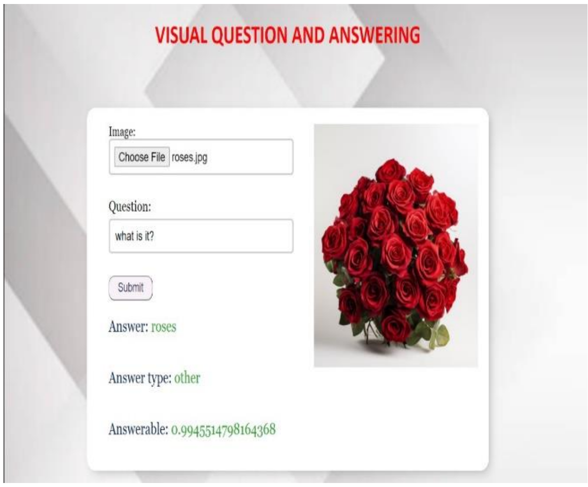
Plots


Training Loss
Validation Loss
7
6
5
4
3
2
0
10
20
30
40
50


Training Accuracy
Validation Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
10
20
30
40
50


0.82
0.80
0.78
0.76
0.74
0.72
0.70
Training Answerability
Validation Answerability
0
10
20
30
40
50

Samples



VISUAL QUESTION AND ANSWERING

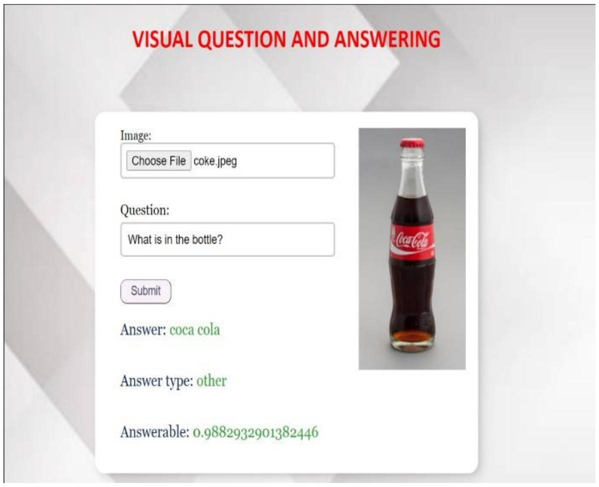
Image: roses.jpg

Question: what is it?

Answer: roses

Answer type: other

Answerable: 0.9945514798164368



VISUAL QUESTION AND ANSWERING

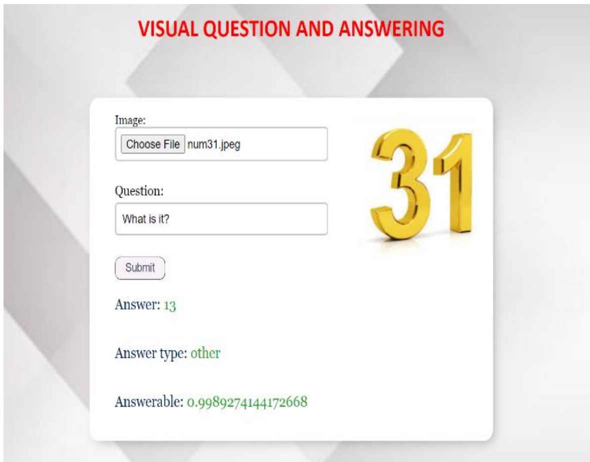
Image: coke.jpeg

Question: What is in the bottle?

Answer: coca cola

Answer type: other

Answerable: 0.9882932901382446



VISUAL QUESTION AND ANSWERING

Image: num31.jpeg

Question: What is it?

Answer: 13

Answer type: other

Answerable: 0.9989274144172668

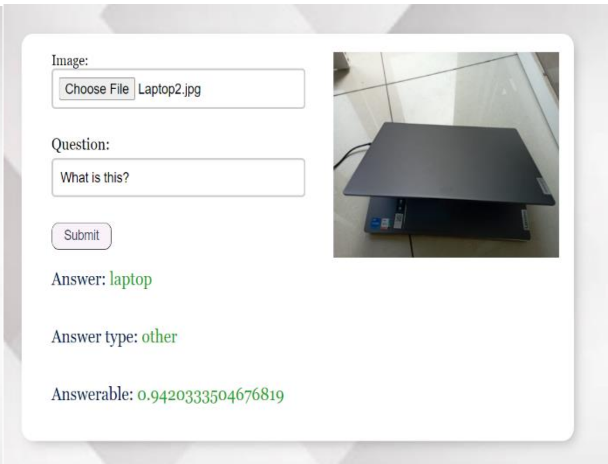


Image: Laptop2.jpg

Question: What is this?

Answer: laptop

Answer type: other

Answerable: 0.9420333504676819

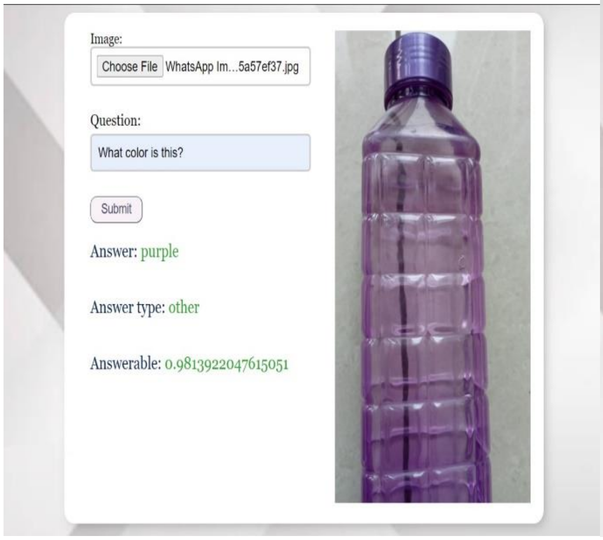


Image: WhatsApp Im...5a57ef37.jpg

Question: What color is this?

Answer: purple

Answer type: other

Answerable: 0.9813922047615051

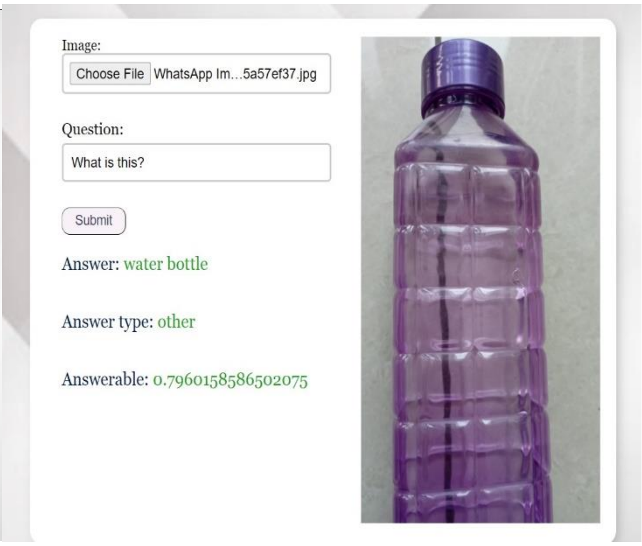


Image: WhatsApp Im...5a57ef37.jpg

Question: What is this?

Answer: water bottle

Answer type: other

Answerable: 0.7960158586502075

CONCLUSION

Our project explored the fascinating field of visual question answering (VQA), where computers are trained to understand and respond to questions about images. Through our work, we've demonstrated the potential of VQA systems to bridge the gap between images and natural language, enabling more intuitive interactions between humans and machines. By combining computer vision techniques with natural language processing, we've developed a system that can accurately interpret questions about visual content and provide meaningful answers.

Github Link

https://github.com/maratidivya/Visual-Question-Answering