# Visual Question Answering with Deep Learning

**Hongyu Zhang, Ning Wan, Yilin Li, Zhangkaiwen Chu**
University of Pennsylvania
{hz53, ningwan, yilli, chuzhkw}@seas.upenn.edu

## Abstract

In this project we explore visual question answering, which is to estimate a natural-language answer given an image and a natural-language question. We use the best model in the VQA paper as the strong baseline. We compare different image encoders and text encoders. We also try to add attention for the image encoder. We evaluate our model on VQA dataset. We use a truncated exact match rate as the metric. We achieved 44.37% accuracy with the strong baseline, and by adding attention, the accuracy is improved to 44.70%.

## 1 Introduction

Visual question answering is the task of predicting a natural-language answer given an image and a natural-language question. This task involves lots of aspects about machine learning. NLP is needed to parse the question, and extracting related information from the image involves various computer vision techniques, like objection detection and classification. Some question may also involve knowledge based reasoning.


Figure 1: Task Example

Figure 1 showns an example of visual question answering. The picture is about a pizza, and the questions are related to this piazza. A visual question answering system is expected to answer the questions given this picture. The first question can be answered with the information in the picture, while the second question needs extra knowledge.

Visual question answering aims at answering free-from, open-ended, natural-language question about the image. However, in our project, the target questions are limited to simple questions, which can be answered with only one word.

Visual question answering is a task happens frequently in real-life. The success in visual question answering should be a big progress in artificial intelligence. Also, it is a multi-discipline artificial intelligence, so we can combine what we formerly learned with the NLP knowledge in this course. We are also eager to see the effect of applying attention for extracting related information from image given the question.

## 2 Literature Review

Much prior work already exists on the topic of visual question answering.

Our project is inspired by the VQA challenge hosted by the VQA group in Virginia Tech and Georgia Tech[1]. Their VQA challenge is to build models to compete for accuracy in answering open-ended questions about images. In their paper, Antol et al. (2015) proposed the task of free-form and open-ended visual question answering. They provided a dataset built upon MS COCO image dataset with human proposed questions and answers. Models were evaluated by comparing the answers to human-proposed answers. Several baselines they tried such as 'random', 'prior yes', and 'nearest neighbor'. However, the best model utilized LSTM for the natural language question and VGGNet for the input image. The concatenated

[1]https://visualqa.org/index.html

vector was passed through a 2 layer MLP to generate a distribution over top K possible answers. It achieved 57.75% accuracy on all questions and 80.50% on yes/no questions.

In addition, we thought that it would be useful to map the input image to text and use the text information to help answer the question. Chen and Zitnick (2015) proposed a RNN based bi-directional mapping between images and text descriptions. The model has four layers: the hidden layer, the visual hidden layer, the output layer and the visual output layer. The hidden state is generated by the previous word, the previous state and the visual features. The visual features in the hidden state serve as visual memory, so that the visual hidden state can remember long term visual concepts. The proposed model outperformed RNN baselines on all the datasets. On MS COCO dataset, the proposed model achieved 18.99 BLEU and 20.42 METEOR, while the human performance is 20.19 BLEU and 24.94 METEOR.

Last but not least, we thought it would be helpful to review traditional question answering systems. Rajpurkar et al. (2016) discussed different strategies to deal with the question answering task. They created the Stanford Question Answering Dataset (SQuAD) which is based on a collection of Wikipedia articles. They used transitional models like logistic regression and sliding windows as baselines. For the sliding window approach, they calculated the unigram or bigram overlap between the sentence and the question containing it. The expanded approach is distance-based extension with the sentence containing the candidate's answer. Logistic regression is the best, which is based on the features extracted from each answer. The representative features are matching words, bigram frequencies and root match, length and span word frequencies, lexicalized features, and dependency tree path features. The evaluations are based on extract match and F1 score compared with human performance for those models, where logistic regression has F1 score over 50% which is much greater than the sliding window's 20% but is not as good as 90% through human evaluation.

## 3 Experimental Design

### 3.1 Data

We decide to use the VQA dataset collected by the VQA group discussed aboved in literature review (Antol et al., 2015). This dataset contains open-ended questions and answers about real images from Common Objects in Context (COCO). There are 3 kinds of datasets inside: image dataset, question dataset, and annotation dataset. The image dataset contains a total of 204721 real images from Common Objects in Conext (COCO). The question dataset contains a total of 1105904 questions generated by humans. And the annotation dataset contains a total of 11059040 answers, where each question has 10 answers generated by 10 human annotators. The questions and annotations are stored in JSON format. An example from the dataset is shown in Figure 2. The dataset is already splitted into training set, validation set, and test set. Table 1 shows the statistics in each set. There are at least 3 questions per image and 5.4 questions on average per image.

| Dataset | # Image | # Question | # Annotation |
|---|---|---|---|
| Training | 82783 | 443757 | 4437570 |
| Validation | 40504 | 214354 | 2143540 |
| Test | 81434 | 447793 | Not Given |

Table 1: Dataset Statistics

Image Dataset Example


```
{
    "image_id": 458752,
    "question": "What position is this man playing?",
    "question_id": 458752001
}
```


Question Dataset Example


```
{
    "question_type": "what",
    "multiple_choice_answer": "pitcher",
    "answers":
        [
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 1},
            {"answer": "catcher", "answer_confidence": "no", "answer_id": 2},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 3},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 4},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 5},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 6},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 7},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 8},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 9},
            {"answer": "pitcher", "answer_confidence": "yes", "answer_id": 10}
        ],
    "image_id": 458752,
    "answer_type": "other",
    "question_id": 458752001
}
```


Annotation Dataset Example

Figure 2: Dataset Example

| Answer Type | Accuracy % |
|---|---|
| yes/no | 64.42 |
| number | 26.93 |
| other | 9.38 |

Table 2: Simple Baseline Answer Type Accuracy

### 3.2 Evaluation Metric

We use the same evaluation metric as the original VQA paper proposed (Antol et al., 2015), as shown in Equation 1. The new "accuracy" metric for any answer is computed as the minimum value between (a): 1 and (b): the number of exact matches of the answer and the 10 annotated answers provided by humans divided by 3.

$$Accuracy = min(\frac{\#\ exact\ matches}{3}, 1) \quad (1)$$

For example, if an answer is the exact match with at least 3 out of 10 answers provided by 10 human annotators, we regard the answer as having $100\%$ accuracy. The overall accuracy of the dataset is the average accuracy of all the answers in the dataset. The reason for using this new "accuracy" metric is to cope with variability in the phrasing of annotated answers, as human annotators can provide answers in different wordings. This evaluation metric is robust to this variability and is frequently used in past VQA publications like Zhang et al. (2016) and Goyal et al. (2016).

### 3.3 Simple Baseline

We use the majority class baseline as our simple baseline. For each question type in the validation or test set, we choose the most frequent answer in the training set for that question type. The type of a question is determined by the first few words of the question, like "how many", "what color is the", "where is the", and so on. There are 65 different types of questions overall, according to the official VQA group. Since this simple baseline model does not need fine-tuning, we directly test the model on the validation set. It achieves $32.36\%$ overall accuracy on the validation set. Table 2 contains different answer types with their corresponding accuracies, and Table 3 contains question types with top 5 accuracies. It has $64\%$ accuracy on yes/no answers, which is decent. But it performs poorly on number answers ($27\%$) and open-ended answers ($9\%$).

| Question Type | Accuracy % |
|---|---|
| could | 76.62 |
| do | 68.66 |
| does the | 65.19 |
| does this | 65.02 |
| are | 64.57 |

Table 3: Simple Baseline Question Type Accuracy (Top 5)

## 4 Experimental Results

### 4.1 Published Baseline

We took the best model from the original VQA paper (Antol et al., 2015) as our strong baseline. The model uses a two layer LSTM to encode the questions and the last hidden layer of VGGNet(Simonyan and Zisserman, 2014) to encode the images. The image features are then normalized. Both the question and image features are transformed to a common space and fused via element-wise multiplication, which is then passed through a fully connected layer followed by a softmax layer to obtain a distribution over answers. The model's schematic is shown by Figure 3. After 10 epochs of training, the model achieved 44.37% overall accuracy, including 65.30% on yes/no questions, 29.82% on number related questions, and 32.28% on other types of questions. Unfortunately, the result was a lot worse than the implementation of the original paper, which has 57.75% overall accuracy. Multiple possible reasons for this:

1. We did not have enough time to tune our model to its best.

2. We only adopted the model structure of the original paper. Training details, questions like what optimizer to use and what batch size to take, are missing from the original paper. Therefore, we tried our best to tune a sound model from scratch.

3. To reduce the training time, we resized all training images into 224 by 224 and we only saved top 1000 answers as possible classes. This might result in significant loss of information.

Our results were directly comparable because we used the exactly same dataset and the evaluation metric to compare.

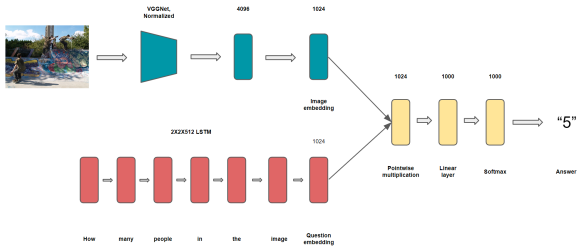


Figure 3: Strongbaseline model schematic.

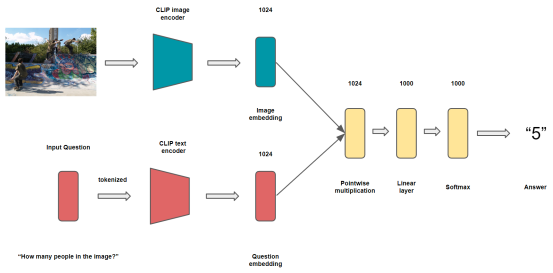


Figure 4: Extension 1 model schematic.

### 4.2 Extension 1

In our first extension, we adapted CLIP (Contrastive Language-Image Pre-training) (Radford et al., 2021) network and utilized its image encoder and question encoder. CLIP is a neural network model for the task of multimodal learning, which involves learning to map inputs from multiple modalities (such as text and image) to a common representation. It is trained using a combination of supervised and self-supervised learning, with the goal of improving the performance of a wide range of downstream tasks that involve multimodal inputs. One of the major reason that we chose CLIP is its ability to learn a common representation for text and image inputs without the need for explicit alignment between the two modalities.

CLIP is composed of several key components, including an image encoder, a text encoder, and a multimodal encoder. The image encoder is a convolutional neural network (CNN) that takes an image as input and produces a compact representation of the image in the form of a feature vector. The text encoder is a transformer-based language model that takes a sequence of words as input and produces a representation of the text. The above two representations are then passed through the multimodal encoder, which combines the representations of the image and text inputs to produce a single multimodal representation. In our adapted model, we only used the image encoder and the text encoder. We replaced them for VGG net and LSTM network from the original VQA paper. The model can be summarized by Figure 4. The performance of the model on the validation set is shown in Table 4. It performs better than the strong baseline in the number answers (33.05%), but its overall accuracy is about 0.2% lower than the strong baseline. This is expected, as our strong baseline follows the best model in Antol et al. (2015), which has very superb performance for our model to beat.

### 4.3 Extension 2

In our second extension, we used attention(Yang et al., 2015) to help extract information from the visual feature. We separated the visual feature to a group of visual features and add each visual feature with the encoded text to generate 196 multimodal features. This separation happened in the maxpooling of the original vgg model so that more multimodal features were passed to two attention layers. Inside an attention layer, features were passed to a tanh layer and a fully connected layer to generate attention score. A dot product between the visual features and the attention scores returned weighted visual features. Finally, we used the sum of the weighted visual feature and the encoded text as the final feature. The model can be summarized by Figure 5. Inside the Attention network, the model looks like Figure 6.

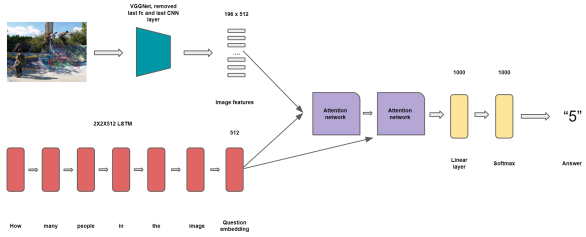


Figure 5: Extension 2 model schematic.

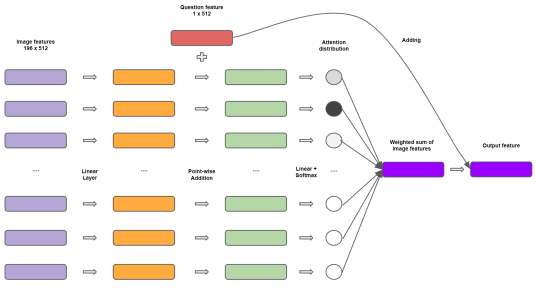


Figure 6: Extension 2 Attention module schematic.

| Accuracy % | | | | |
|---|---|---|---|---|
| Models | All | yes/no | number | other |
| Simple baseline | 32.36 | 64.42 | 26.93 | 9.38 |
| Strong baseline | 44.37 | 65.30 | 29.82 | 32.28 |
| Clip-adapted | 44.15 | 65.04 | 33.05 | 31.19 |
| Attention-adapted | 44.70 | 68.04 | 30.97 | 30.56 |

Table 4: Accuracy of our methods for the open-ended task on the VQA-2.0 val set for real images. Simple baseline is Majority class method. Strong baseline is the LSTM in the original VQA paper. Clip-adapted is our first extension and Attention-adapted is the second extension.

Table 4 shows the accuracy results of all our models, among which attention-adapted model performs the best with total accuracy 44.70%. Our attention model also has the best performance on yes/no answers (68.04%), while our clip model has the best performance on number answers (33.05%) and our strong baseline model has the best performance on open-ended answers (32.28%).

### 4.4 Error Analysis & Model Comparison

We select some failure examples from the validation set in Table 5 for our best model, attention adapted LSTM. We identify 2 categories of common errors. The first one is text recognition issue, which means it cannot adequately detect the apparent text information inside the picture. The second one is time recognition failure, the model does poorly on extracting the time information based on the shape and context provided in the photo. Both two errors are very common in my model, and our predicted results are generally "$< unk >$".

Compared to strong baseline, our best model does better in number type as shown in Table 4 because more features are detected related to the picture in the image encoder under attention-based model.

Apart from that, Yes/No type is significantly better in our best model, this might be due to the fact that attention-based model has more power in text encoding since attention is added to let the model focus more on important inputs. Since Yes/No question answering occupies a large proportion in the dataset, the features regarding those are better detected. In the meantime, some other types like "which", "where" and "how" are not handled better than our strong baseline as shown in Table 7.

## 5 Conclusions

Throughout our project, we implemented several models to complete the visual question answering task. We used majority classifier as the simple baseline, and we used the best model in the prior work as the strong baseline. We implemented Clip-adapted and attention-adapted models as our extensions. Our attention model outperforms the strong baseline by a little bit in overall accuracy, while our Clip model underperforms the strong baseline by a little bit. Overall, our attention model performs the best. However, we did not come close to the human performance or the state-of-the-art performance on the task. Our models do not perform well especially in the number answers and the open-ended answers, suggesting that we need to do further work to enhance the models' understanding and common sense knowledge about the images.

## Acknowledgments

We used the publicly released VQA API to load data and evaluate our models and we would like to express thanks to the VQA team[2]. We used OpenAI's CLIP model[3] in our project and we would like to thank them. We also would like to express our thanks to Yifei Li for his generous support and guidance throughout our project.

[2]https://visualqa.org/evaluation.html
[3]https://github.com/openai/CLIP

| Category | Question | Actual | Predicted |
|---|---|---|---|
| text recognition | 'What name is on the sign?" | "scott wells" | "$< unk >$" |
| text recognition | 'What name is in the picture?" | "emile swain" | "$< unk >$" |
| text recognition | 'What name is written on the building?" | "hawkins companies" | "$< unk >$" |
| text recognition | "What number is on the motorcycle?" | "750" | "$< unk >$" |
| time recognition | "What time is on the clock?" | "8:35" | "$< unk >$" |
| time recognition | "What time of the day is it?" | "afternoon" | "daytime" |
| time recognition | "What time does the clock say?" | "11:10" | "$< unk >$" |
| time recognition | "What time of year is this?" | "winter" | "spring" |

Table 5: Error Analysis with Examples

| Question Type (Yes/No) | Accuracy for Attention Model | Accuracy for Strong Baseline |
|---|---|---|
| could | 74.53 | 72.38 |
| does this | 68.28 | 66.98 |
| is it | 70.25 | 65.63 |

Table 6: Model Comparision (Yes/No)

| Question Type (Other) | Accuracy for Attention Model | Accuracy for Strong Baseline |
|---|---|---|
| which | 32.84 | 36.12 |
| where is the | 18.64 | 20.75 |
| how | 16.52 | 18.08 |

Table 7: Model Comparision (Other)

## References


Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

Xinlei Chen and C. Lawrence Zitnick. 2015. Mind's eye: A recurrent visual representation for image caption generation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2422–2431.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Making the v in vqa matter: Elevating the role of image understanding in visual question answering.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2015. Stacked attention networks for image question answering.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5014–5022.