

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/382146362>


Visual Question Answering (VQA)

Article · July 2024

CITATIONS
0

READS
229


5 authors, including:



Faaraan Farid Kazi
Yeshiva University

6 PUBLICATIONS 0 CITATIONS


SEE PROFILE



Shikshit Gupta
Yeshiva University

6 PUBLICATIONS 88 CITATIONS

SEE PROFILE



Onkar Kunte
Yeshiva University

5 PUBLICATIONS 0 CITATIONS

SEE PROFILE

Visual Question Answering (VQA)

Faaraan Farid Kazi	Shikshit Gupta	Onkar Kunte
fkazi1@mail.yu.edu	sgupta11@mail.yu.edu	okunte@mail.yu.edu
Yeshwanth Kesani	Venu Khare	
ykesani@mail.yu.edu	vkhare@mail.yu.edu	

Abstract

The creation of a comprehensive dataset from Yeshiva University's machine learning course and the creation of an advanced multimodal VQA model are the two goals of this work, which falls under the category of Visual Question Answering (VQA). Natural language processing and computer vision groups have both shown a great deal of interest in Visual Question Answering (VQA), in part because it provides insight into the connections between two crucial sources of data. Existing datasets and the models constructed from them have concentrated on answering questions that can be resolved by analyzing the query and image separately.

The three carefully carried out stages of the dataset compilation were the development of question-answer pairs, image and transcript collecting, and image processing. The endeavor succeeded in creating a strong basis for the analysis in spite of obstacles like the intricate structure of the picture collection and the worse quality of the original audio content. Visual Question Answering (VQA) is a difficult job that is gaining interest from the natural language processing and computer vision sectors. To determine the proper response given a picture and a natural language question, one must use general knowledge and reasoning skills to analyze the image's visual components. We compare contemporary methods to the problem in order to assess the state of the art in the first section of our review. We categorize techniques based on how they link the textual and visual senses.

The study's findings significantly advance the field of VQA, including the difficulties and lessons discovered during the dataset building and comparative analysis of the models. They shed information on the difficulty and potential of creating systems that can comprehend and generate responses in sophisticated ways in visually situated speech scenarios, opening the door for further developments in artificial intelligence and machine learning.

GitHub link: [VQA Special Topics](#)

1. Introduction

Image captioning and visual question answering (VQA) are two language and vision challenges that have become more well-known in recent years as computer vision research has moved past "bucketed" recognition and toward multi-modal problem solving. Problems in the nexus of vision and language are challenging due of the intricate compositional structure of language. However, recent research has shown that language also offers a strong prior, which may produce high surface performance even in cases when the underlying models do not fully comprehend the visual content. The majority of VQA benchmarks use word embedding techniques, recurrent neural networks (RNNs), and a collection of object descriptors made up of vectors representing visual attributes and bounding box coordinates to generate a question representation. Then, to train a VQA model, word and picture representations are combined and input into a network. These methods are useful, nevertheless, when no information other than the visual content is needed.

In this sense, machine learning (ML) lectures—a vital part of online learning—are especially difficult. Packed with intricate illustrations, mathematical formulas, and copious amounts of material, these lectures pose distinct difficulties for understanding and participation. Conventional Visual Question Answering (VQA) systems have been comparatively successful in providing answers to inquiries pertaining to still visual information since they mostly concentrate on static pictures. But the richness and complexity of the educational content—particularly in the ML lectures—requires a higher degree of comprehension than what the typical VQA systems can offer.

The purpose of this study is to broaden the scope of VQA to include complicated educational topics. In order to do this, we investigated a number of models and chose the best performing model that was customized for our particular

dataset. This dataset, which originates from Yeshiva University’s Katz School of Science and Health (YUKSSH), comprises of machine learning lectures supplemented with open-ended questions and responses [19].

Our approach outperforms approaches intended for more broad picture material by concentrating on the model that best handles the nuances of instructional content. We think that our study will make a big difference in how well people grasp complicated educational materials and how quickly academic question-answering systems are developed.

2. Related Work

A field of study centered on static pictures called Visual Question Answering (VQA) has arisen. Answering questions via video. In order to complete this duty, the intelligent system must automatically respond to a natural language query based on the information included in a particular movie. A number of recent studies have suggested models [9] [8] and datasets [5] [12] [17] [18] for visual question answering. A pioneering pair, Malinowski and Fritz [17], provided one of the earliest open-ended datasets for solving picture questions. For the VQA challenge, they also suggested using a recurrent neural network (RNN) model. This dataset served as the basis for further investigation. The availability of benchmark datasets significantly expedited the VQA field’s growth between 2015 and 2017. These datasets offered a standardized approach to solving VQA issues, which promoted a more coordinated research agenda.

Inspiration. In Andreas et al. [4], the Neural Module Networks (NMN) are shown, and in Andreas et al. [4], they are expanded. They are made especially for VQA, hoping to take use of the questions’ compositional language structure. There are wide differences in the difficulty of questions. As an illustration, is this a truck? just calls for obtaining a single piece of data from the picture, in contrast to How many things are to the left of the toaster? calls for several processing stages, including counting and identification. NMNs are a type of network that dynamically assembles to represent the intricacy of a given problem for every problem occurrence. The strategy is similar to textual quality assurance methods that transform inquiries into logical statements by using semantic parsers. Applying this logical reasoning to continuous visual characteristics rather than discrete or logical predicates is a major contribution of NMNs.

Architectural advancements have significantly enhanced picture VQA in recent years. For instance, multimodal transformer [21] networks simultaneously represent text and picture modalities, bottom-up and top-down attention [3] integrate high-level semantic and low-level visual data, and graph neural networks take structured knowledge into account. A noteworthy model is MiniGPT-4 [23], which delivers new state-of-the-art performance on numerous VQA benchmarks using a transformer-based architecture that was

pre-trained on huge multimodal databases. The goal of the next phase of VQA research was to deal with video’s dynamic input. VQA datasets were first created for animated films and television shows. These datasets include TGIF-QA [22], an animated dataset compiled by Jang et al. [11], and MovieQA, which is based on movie-related question-answer pairs and plot summaries. After establishing the fundamentals of VQA, the community turned its attention to difficulties unique to certain domains. VQA for instructional films has drawn more attention lately as it introduces new difficulties requiring sophisticated thinking abilities.

This project develops models based on online lectures and classes with the goal of advancing VQA for instructional videos. We offer a dataset that was gathered from open-ended questions and responses and machine learning courses at Yeshiva University’s Katz School of Science and Health [19]. Our approach outperforms prior systems built for various types of videos by using multistep reasoning, thorough modeling of textual and visual parts, and external knowledge resources to tackle difficult inquiries concerning course material. We think that our effort will help the educational sector comprehend videos more deeply and be able to respond to questions.

3. Methods

3.1. Data Collection

Three separate processes comprise the data collection for this study: the creation of questions and answers, the gathering of images and transcriptions. To create the data set required to examine the machine learning course, each step is essential.

3.1.1 Image Collection

Beginning with the picture gathering stage, this initial step appears to be the easiest. It entails gathering slides from the machine learning course’s PDF presentations, which led to the collection of photos.

3.1.2 Transcript Collection

Going ahead, transcript collecting presents a really difficult issue. This phase’s main goal is to transcribe the spoken information from the machine learning course’s recorded lectures. Due to the limitations of transcription tools, which often do not handle video inputs, the procedure begins with extracting audio from videos. The Silero model [10], the Wav2Vec Base model [6], the Wav2Vec2 large-lv60 model [6], and the Google Speech-to-Text API [20] are among the models that are based on Python. In addition, expert web resources including Cockatoo, Deepgram [10], Trint [10], Parrot [10], Veed, and Speechox are utilized.

After a thorough examination of 10 distinct tools, the expected conclusion is obtained: professional tools greatly outperform their Python model equivalents. Forming intelligible sentences is difficult with the transcriptions produced by Wac2Vec and other Python models; they can only be described as disastrous. This is not to argue that expert instruments consistently yield perfect outcomes. Even while a large percentage of their transcriptions result in proper sentences, a notable portion still contain mistakes, giving rise to statements that appear correct but are actually erroneous. Deepgram was determined to be the best option following a careful comparison and assessment because of its exceptional accuracy.

It is crucial to acknowledge that one of the biggest challenges is the poor quality of the original recordings. The audio is quite noisy, and the material can become unintelligible even if there is just a little gap between the speaker and the microphone. For this reason, manual content inspection takes a lot of time but is an essential aspect of this process. The duration of the manual inspection process is significant.

3.1.3 Formulating Queries and Responses

Making questions and answers for every slide is the last stage. In addition to information like the lecture week and slide page, it's critical to record contextual cues that lead to the solutions. Asking "What is the topic of this slide?" repeatedly helps cull through the ten question-answer pairs that are usually included on each presentation. The goal of this critical inquiry is to compile an extensive list of summary questions. The sets of question-answer pairings are meticulously arranged and saved in a structured JSON format for uniformity and quick retrieval.

Table 1. Field Descriptions for QA Pairs

Field	Description
Instruction	The question or instruction for the QA pair
Context	Contextual information, often including slide or transcript content
Response	The corresponding answer or response to the question
Category	The category of the QA pair, e.g., 'closed qa', 'information extraction'
Week	The week of the ML course to which the content belongs
Page	The page number of the slide

3.2. Dataset Summary

In conclusion, there are three crucial stages to the data gathering process, all of which have a major impact on the production of an extensive data set. Future analysis and learning are built upon the painstaking transcribing procedure, the methodical creation of question-answer pairings, and the careful curation of pictures.

3.3. Data Preprocessing

To maximize the formatting and preparation of raw data for later training and modeling, data preprocessing is essen-

tial.

3.3.1 Combining and Organizing

Originally, common attributes like week and page numbers were used to combine datasets made up of question-answer pairs, transcripts, and pictures. Having a cohesive view of all the data points required this integration. The merged dataset's columns were renamed and made simpler to more accurately represent their contents after the merge. For example, columns that were before titled "instruction" and "response" were appropriately renamed "question" and "answer," respectively.

3.3.2 Splitting Datasets

A crucial stage in the modeling process is partitioning the data into training and validation sets. For training reasons, we divided the data for this research into weeks 1–11 and week 14, totaling 8,681 samples—roughly 90% of the entire data. By doing this, a solid data set is provided for the model to learn from. The remaining 10% was set aside for validation in order to evaluate the model's performance on unobserved data critically.

Category	Number
General QA	1,886
Open QA	1,727
Information Extraction	1,541
Closed QA	1,500
Brainstorming	1,064
Summarization	492
Classification	382
Creative Writing	382

Table 2. Category-wise numbers

3.3.3 Creation of Datasets

To expedite the retrieval and processing of data, we created a tailored dataset structure. A diagram of the structure is shown below:

- Pictures were obtained and loaded for every question-answer pair. Every image's resolution was adjusted to 224 by 224 pixels.
- The question and the lecture transcript were combined to form a comprehensive textual prompt, which included a designated space for the model to generate the response.
- Any text that exceeds the token capacity of the model is carefully trimmed due to the inherent limits of the chosen model, especially when it comes to processing extended sequences.

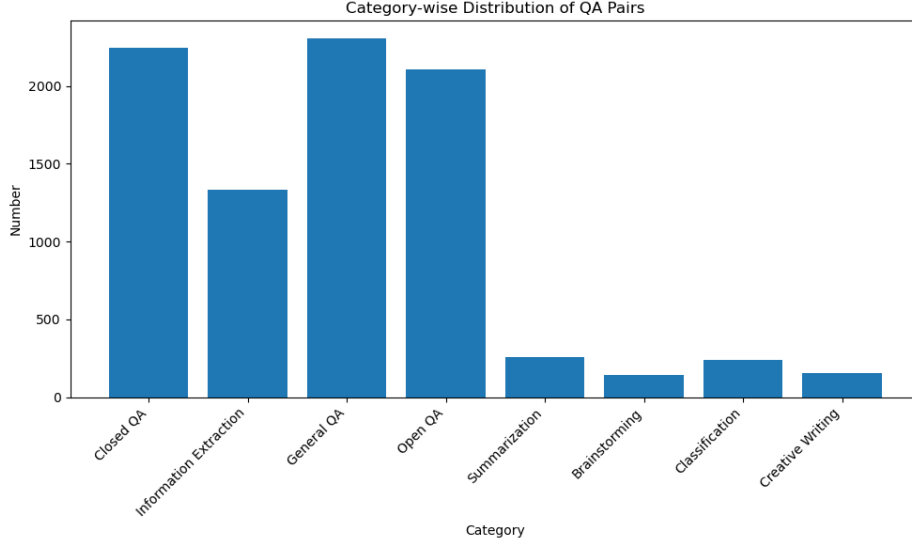


Figure 1. Category-wise numbers

- The right responses were transformed into a format that works with the model and enables the training and validation procedures.

Table 3. Data Type Statistics

Data Type	Mean String Length	Max String Length
Answer	220.1	400
Question	73.06	326
Transcript	707.31	3,034

3.3.4 Loading Data

A sophisticated data loading technique was used to obtain data effectively during training. The goal in creating this technique was to: Accurately assemble sets of data points. A cohesive structure appropriate for modeling should be created by combining various data items, such as textual cues and pictures, by using padding to standardize sequence lengths and guarantee consistency among batches.

The data was imported in many batches, each comprising a predefined amount of data points, due to hardware limits. In addition to improving training efficiency, the batching strategy revealed the structure and dimensions of the training data, which included pictures, verbal cues, and right responses.

3.4. Model Architecture

3.4.1 LLaVA

The LLaVA-1.5 model, a cutting-edge multimodal architecture intended for general-purpose visual and linguistic comprehension, is used in this work. Large Language and

Vision Assistant, version 1.5, created by [16], is a notable development in the merging of a vision encoder with a large language model for multimodal activities. The block architecture of the mode is shown in Figure 2.

The main reason for choosing the LLaVA-1.5 model was its robust and adaptable design, which combines many modules to manage a range of data kinds. The fundamental components of the model are as follows:

- **Vision Encoder:** It converts visual material from pictures into a complete collection of visual features by using a pre-trained CLIP ViT-L/14 as the vision encoder.
- **Text Encoder:** This component effectively encodes textual data, capturing the subtleties and complexities of the language, by using LLaMA 2 as the text encoder.
- **Projection Matrix:** To align the features from both modalities, a straightforward projection matrix connects the visual encoder and the LLM.
- **Multimodal Fine-Tuning:** To showcase the model’s versatility and efficacy, it is adjusted for two distinct use cases: Science QA and Visual Chat.

3.4.2 IDEFICS

Deepmind’s Flamingo visual language model is closed-source, while IDEFICS (Image-aware Decoder Enhanced à la Flamingo with Interleaved Cross-attentionS) [2] is an open-access version of it. Similar to GPT-4, the multimodal model generates text outputs and takes in any order of picture and text inputs. IDEFICS is based only on models and data that are made publicly available.

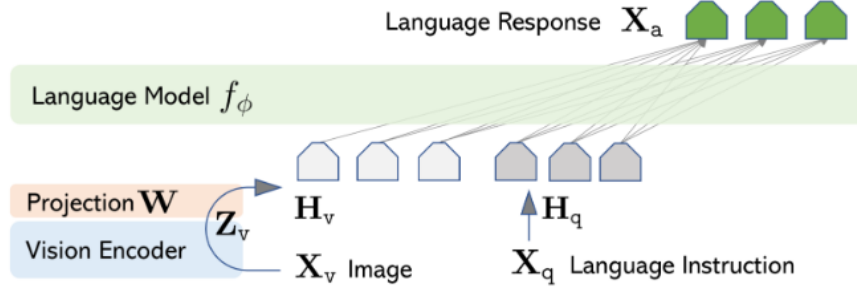


Figure 2. PAccording to the reference, the pre-trained model architecture of LLaVA-1.5 consists of a large language model (LLM) and a vision encoder. [16]

3.4.3 BLIP

The goal of closing the semantic gap between visual perception and linguistic representation is what inspired the creation of the BLIP (Bidirectional Language-Image Pre-training) model. Fundamentally, the BLIP model [14] provides evidence of how sophisticated language processing and high-resolution visual understanding may be combined to produce a more coherent and reciprocal understanding of content. The model deftly captures the nuances of visual cues and text correlations by combining advanced vision encoders with complex language decoders. This opens the door for a new wave of Visual Question Answering (VQA) and multimodal applications. The advancement of AI’s ability to comprehend and produce answers that are similar to those of humans depends on this harmonization of modalities, which is also essential for encouraging more natural interactions between human

3.4.4 Training Environment

The training environment for Pix2Struct was configured on Google Colab, leveraging the capabilities of a powerful Nvidia A100 Tensor Core GPU. The system RAM usage remained moderate at 5.0 GB out of the available 83.5 GB. Crucially, the training process effectively utilized the GPU memory, consuming 37.4 GB out of the 40.0 GB available. This ample GPU memory allocation likely facilitated the training of Pix2Struct with a sufficient number of trainable parameters and potentially larger batch sizes, which can improve model performance. The disk space remained moderate at 24.4 GB out of the total 201.2 GB, suggesting adequate storage for the training data and model checkpoints. Overall, the training environment on Colab provided a suitable platform with appropriate resource allocation for training Pix2Struct.

3.4.5 Optimizer

we employed the AdamW optimizer for training our model, which is a variant of the Adam optimizer that incorporates weight decay directly into its update step. The AdamW optimizer is well-suited for training deep learning models, particularly in scenarios where regularization is crucial for preventing overfitting. The update rule for the AdamW optimizer is defined as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \quad (1)$$

- θ_t represents the parameters of the model at iteration t ,
- η denotes the learning rate, which determines the step size during optimization,
- \hat{m}_t is the exponentially decaying average of past gradients,
- \hat{v}_t represents the exponentially decaying average of past squared gradients, and
- ϵ is a small constant introduced for numerical stability.

3.4.6 Loss Function

we utilized the Cross-Entropy Loss function for computing the loss during the training of our model. This loss function is particularly suitable for classification tasks, where the model is trained to predict one out of multiple classes for each input. The Cross-Entropy Loss function computes the difference between the predicted probability distribution and the true distribution of class labels, measuring the discrepancy between the two distributions.

The formula for the Cross-Entropy Loss function is given by:

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C \text{true}_{i,j} \cdot \log(\text{pred}_{i,j}) \quad (2)$$

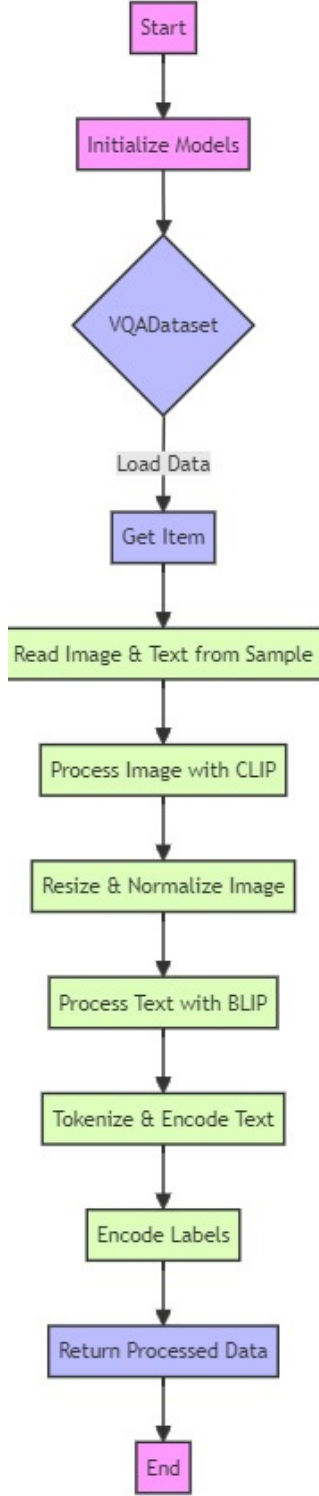


Figure 3. Training Flowchart

Where:

- N represents the number of samples in the batch,

- C denotes the number of classes,
- $\text{true}_{i,j}$ is a binary indicator (0 or 1) indicating whether class j is the true class for sample i ,
- $\text{pred}_{i,j}$ is the predicted probability that sample i belongs to class j ,
- \log denotes the natural logarithm, and
- The negative sign at the beginning ensures that the loss is minimized during training.

The 'ignore_index' parameter is used to specify a class label that should be ignored when computing the loss. This is particularly useful in scenarios where certain class labels are reserved for padding or special tokens and should not contribute to the loss computation.

3.4.7 Accumulation Steps

To address computational resource limitations during training, we employed gradient accumulation with a unique configuration: a batch size of 1 and 64 accumulation steps. This approach is particularly beneficial when memory constraints restrict the use of larger batches, yet training still requires updates based on a larger effective batch size. Gradient accumulation works by accumulating gradients from multiple mini-batches before updating the model's parameters. In our case, even though each mini-batch only contained a single data point (batch size = 1), gradients were accumulated over 64 such mini-batches. This effectively increased the magnitude of the update applied to the model parameters, mimicking the benefits of training with a batch size of 64 without the associated memory overhead. The accumulated loss is then scaled down by 64 to compensate for the gradient accumulation. After 64 mini-batches, the optimizer updates the model parameters based on the accumulated gradients, and the gradients are reset for the next iteration. This approach allowed Pix2Struct to leverage the advantages of larger effective batch sizes for training efficiency, even with the limitations of a single data point per batch.

3.4.8 Pix2Struct

Pix2Struct [13] is a novel approach intended to address the challenges associated with visually-situated language processing. In contrast to other approaches limited by specialized domain-specific recipes, Pix2Struct presents a revolutionary method that parses web page screenshots using masks into simplified HTML by using pretraining. This approach makes use of the huge diversity of visually appealing items that can be found on the internet and are neatly contained inside HTML structures. This results in a large amount of pretraining data that can be used for a variety of

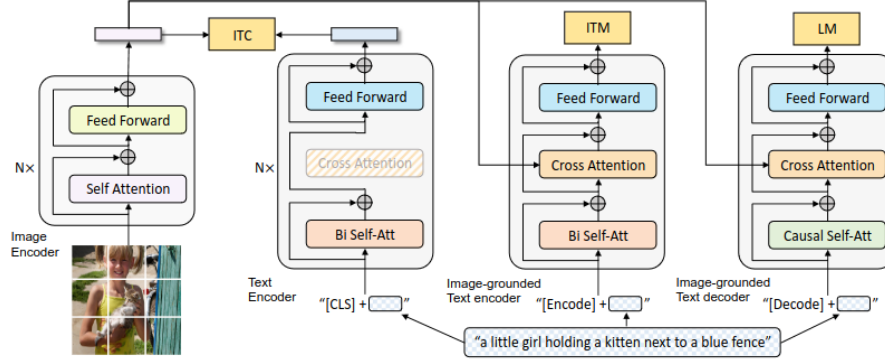


Figure 4. Pre-training model architecture of BLIP [14]

downstream tasks. However, Pix2Struct goes beyond standard pretraining signals like picture captioning and optical character recognition (OCR). Innovative features including the smooth integration of verbal prompts directly onto input images and variable-resolution input representation are introduced. This breaks down the barriers that have traditionally separated digital and physical languages, enabling a more flexible and thorough interpretation of visually-situated language.

The model may explain visual contents, generate narratives based on a series of photos, respond to inquiries concerning images, or operate as a pure language model in the absence of visual inputs. When examined using in-context few-shot learning, IDEFICS performs comparably to the original closed-source model on a number of picture-text benchmarks, such as visual question answering (both open-ended and multiple choice), image captioning, and image categorization. There are two versions available: one with a massive 80 billion parameters and another with just 9 billion.

4. Results

4.1. Training result

Three fundamental metrics—ROUGE and COSINE used in the comparative assessment of the models, which is summarized in Table 3. These metrics evaluated n-gram overlap, semantic similarity, and translation correctness, providing a thorough analysis across training and validation datasets.

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [15]: ROUGE is a set of metrics used to evaluate the quality of automatic summaries by comparing them to reference summaries. It measures the overlap of n-grams (sequences of words) and other text units between the system-generated summary and the reference summary.

- COSINE (Cosine Similarity) [1]: COSINE is a metric used to measure the similarity between two vectors by calculating the cosine of the angle between them. In natural language processing, COSINE similarity is often applied to measure the similarity between documents or text passages represented as vectors in a high-dimensional space.
- BLUE (Best Linear Unbiased Estimator) [7]: The BLUE evaluation matrix is a method used to evaluate the quality of machine translation. It's primarily used to compare the output of different machine translation systems to a set of reference translations.

Model	Rouge-1	Rouge-2	Rouge-L	COSINE	Epochs
Pix2struct-7B	0.22	0.10	0.20	0.2646	1
Pix2struct-Base	0.40	0.26	0.40	0.396	10
Blip	0.42	0.37	0.18	0.326	10

Table 4. Various model comparison

5. Discussion

The table 4 provides a comparative analysis of different models. Each model's performance is evaluated based on its scores in Rouge-1, Rouge-2, and Rouge-L metrics, which measure the overlap of unigrams, bigrams, and the longest common subsequence between the generated texts and the reference texts, respectively. Additionally, the COSINE similarity metric offers a view into the semantic similarity between the generated text and the reference, and the 'Epochs' column reflects the training duration or intensity for each model. Pix2struct-7B: This model variant shows the lowest performance across all Rouge metrics (Rouge-1 = 0.22, Rouge-2 = 0.10, Rouge-L = 0.20) and a moderate COSINE similarity score of 0.2646. It was trained for only 1 epoch, suggesting minimal training that might not have been adequate to capture complex patterns or relationships in the data. Its lower scores

might indicate it has less capacity or was less effectively trained compared to the other models. Pix2struct-Base: Exhibiting a significant improvement over the 7B variant, the Base model achieves markedly higher Rouge scores (Rouge-1 = 0.40, Rouge-2 = 0.26, Rouge-L = 0.40) and the highest COSINE similarity score of 0.396 among the models presented. Trained for 10 epochs, this suggests that extended training periods positively impact the model's ability to generalize and capture linguistic nuances, making it much more effective at the given task.

Blip: This model presents a balanced profile with strong performance, particularly in Rouge-2 (0.37), which is the highest among all the models. However, its Rouge-L score (0.18) is the lowest, which might indicate a difference in how it handles or prioritizes the sequence length or structure of the output compared to unigram and bigram overlaps. Its COSINE score of 0.326 is lower than the Pix2struct-Base but higher than the 7B variant. Also trained for 10 epochs, Blip shows robustness in capturing bigram relationships more effectively than the others.

The varying performance across different metrics and models can be attributed due to Training Duration. There is a clear correlation between the number of epochs and the performance in Rouge and COSINE metrics, as seen with Pix2struct-Base and Blip, indicating that longer training periods may be crucial for achieving higher quality results.

5.1. Comprehensive Data Gathering Process:

The data gathering process is depicted as comprising three critical stages: transcription, creation of question-answer pairs, and curation of images. Each stage contributes significantly to the development of a comprehensive dataset. By systematically transcribing lecture content and meticulously creating question-answer pairs, a rich resource for training and evaluation is established. Moreover, the inclusion of curated images adds another dimension to the dataset, enhancing its utility for various machine learning tasks

5.2. Importance of Contextual Cues:

One noteworthy aspect emphasized is the significance of contextual cues in crafting effective question-answer pairs. By repeatedly asking "What is the topic of this slide?" during the creation process, the aim is to ensure that the questions generated are relevant and closely aligned with the content of each slide. This approach not only enhances the quality of the dataset but also facilitates deeper understanding and engagement with the material.

5.3. Structured Data Format:

The use of a structured JSON format for organizing question-answer pairs allows for uniformity and efficient retrieval of information. Each QA pair is associated with specific metadata, including instruction, context, response, category, week, and page number. This structured approach facilitates easy navigation and management of the dataset, enabling seamless integration into machine learning pipelines.

5.4. Data Type Statistics:

The analysis of data type statistics provides insights into the characteristics of different types of data present in the dataset. By examining mean and maximum string lengths for answers, questions, and transcripts, researchers gain a better understanding of the data distribution and potential challenges associated with processing varying lengths of text.

6. Conclusion

The comparative analysis presented in this study provides valuable insights into the performance of different models in the context of visual question answering systems. Our findings highlight the inherent challenges in developing VQA systems that not only understand visual content but also generate sophisticated, contextually appropriate responses. While our models have shown promising results, there is a considerable gap that needs to be addressed to enhance both the accuracy of translations and the adaptability of models to diverse visual scenarios. Due to time constraints, our training periods were limited, which likely affected the models' ability to fully learn and generalize from the training data. Longer training periods could potentially improve model performance significantly, as evidenced by the initial improvements observed in models trained for more extended epochs. However, extensive training was not feasible within the scope of this initial study. Looking forward, we aim to explore advanced modeling techniques that could offer substantial improvements. Specifically, we plan to investigate the potential of emerging models such as LLaVA and IDEFICS, which have shown promise in related fields for their robustness and efficiency. Additionally, the application of 4-bit quantization techniques such as LoRA and QLoRA will be crucial. These methods not only reduce the model size, making them more practical for deployment in limited-resource environments but also potentially increase the computational efficiency without significant losses in performance. By integrating these advanced techniques, we hope to overcome the limitations observed in the current generation of VQA systems, enhancing both the precision of our models and their ability to function effectively across varied and complex visual contexts. The ongoing development and refinement of these systems are

critical, as they hold the promise of significantly advancing our ability to automate and enhance the interaction between computers and the visual world.

References

- [1] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974. [7](#)
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. [4](#)
- [3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2(4):8, 2017. [2](#)
- [4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016. [2](#)
- [5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. [2](#)
- [6] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020. [2](#)
- [7] Frankie KW Chan, HC So, Jun Zheng, and Kenneth WK Lui. Best linear unbiased estimator approach for time-of-arrival based localisation. *IET signal processing*, 2(2):156–162, 2008. [7](#)
- [8] Xinlei Chen and C Lawrence Zitnick. Mind’s eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2422–2431, 2015. [2](#)
- [9] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. [2](#)
- [10] Radek Holik, Ruslan Gokhman, and Manish Kumar Thota. Visual question answering (vqa) system for enhanced understanding of machine learning classes. [2](#)
- [11] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766, 2017. [2](#)
- [12] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. [2](#)
- [13] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023. [6](#)
- [14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. [5](#), [7](#)
- [15] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. [7](#)
- [16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. [4](#), [5](#)
- [17] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. *Advances in neural information processing systems*, 27, 2014. [2](#)
- [18] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. *Advances in neural information processing systems*, 28, 2015. [2](#)
- [19] Paul Russo. Katz school of science and health: Online newsletter: Fall 2023. 2023. [2](#)
- [20] Abdelrahman Salah, Ghada Adel, Hussein Mohamed, Youssef Baghdady, and Sherin M Moussa. Towards personalized control of things using arabic voice commands for elderly and with disabilities people. *International Journal of Information Technology*, pages 1–22, 2023. [2](#)
- [21] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vi-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. [2](#)
- [22] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016. [2](#)
- [23] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. [2](#)