Visual Question Answering (VQA) System Video Understanding of Machine Learning Classes

Radek Holik rholik@mail.yu.edu
Yujie Wu ywu3@mail.yu.edu
Sheng-Han Yueh syueh@mail.yu.edu
Pinxue Lin plin3@mail.yu.edu

Yeshiva University, NYC, NY
Katz School of Science and Health

Abstract

This study focuses on the dual objectives of data collection and the development of a multi-modal Visual Question Answering (VQA) model. The dataset is meticulously curated from machine learning course at Yeshiva University. Data collection involves three distinct facets: Image collection, transcription compilation, and text assembly. Notably, this process is notably time-intensive due to the intricacies of execution and the audio's suboptimal quality.

The proposed multi-modal model is strategically designed to early-stage fuse slide images and transcript inputs, facilitating the generation of question-answer pairs. At its core, this project aims to develop a model that transcends traditional video-centric instruction, thereby elevating the learning experience for students. Employing the BLIP-2 model, a pre-trained Transformer-based architecture for image encoding, the model transforms visual data into feature sequences through extensive image dataset utilization.

With this comprehensive methodology, the VQA model endeavors to offer a more effective and interactive learning paradigm than conventional video-based methods. It is, however, acknowledged that the achieved results are not yet optimal. The complexity of lecture content, coupled with limitations in hardware and training capacities, transcript ambiguities, dataset challenges, and the specialized nature of the VQA model contribute to this outcome. Future recommendations underscore the utilization of more robust models and an expanded data collection effort to further enhance the project's outcomes.

1. Introduction

In an era where video content plays a pivotal role, especially for educational purposes, the demand for systems capable of comprehending and interacting with such content has grown significantly. Within this landscape, Machine Learning (ML) lectures hold a prominent position, shaping the online educational experience. These ML lectures, often dense with intricate information, diagrams, and mathematical equations, present distinct challenges in terms of comprehension and engagement.

Traditionally, Visual Question Answering (VQA) systems have concentrated on analyzing static images, achieving remarkable progress in addressing queries related to unmoving visual content. However, the transition from static images to dynamic video content, particularly within intricate subjects like Machine Learning, introduces an array of complexities. These evolved systems must not only grasp the visual components but also understand the evolution of concepts, their interrelationships, and the contextual nuances they exist within.

This study is dedicated to pushing the boundaries of VQA in the realm of educational videos, achieved by crafting models rooted in online lectures and classes. We introduce a dataset sourced from machine learning lectures, accompanied by open-ended questions and answers from Yeshiva University, Katz School of Science and Health (YUKSSH) [1]. To tackle the intricacies of in-depth inquiries about course materials, our model integrates diverse external knowledge sources, intricate modeling of both image and text segments, and a multi-step reasoning approach. As a result, our model surpasses previous methodologies tailored for different types of video content. We are confident that our work will pave the way for a deeper comprehension of video material and elevate the capabilities of question-answering within the educational domain.

2. Related Work

Visual Question Answering (VQA) emerged as a research area that focuses on static images. Malinowski and Fritz [2] are early pioneers who introduced one of the first open-ended datasets for image question answering. They

also proposed a recurrent neural network (RNN) model for the VQA task. This dataset became a foundation for succeeding research. Based on this foundation, Zhu et al. [3] incorporated spatial attention mechanisms into VQA models to enable the focus on relevant image regions. This combination of vision with attention mechanisms proved to be essential.

The expansion of the VQA field from 2015 to 2017 was notably hastened by the availability of benchmark datasets. These datasets offered a uniform methodology for addressing VQA problems, promoting a more cohesive research agenda. Notable examples that attained broad adoption in the community include VQA v1.0[4] and the Microsoft COCO-VQA dataset [5]. These benchmarks consisted of diverse and complex question-answer pairs that challenged the limits of VQA models. Besides affording a shared foundation for comparing different models, these benchmarks brought attention to issues encountered in practical contexts, paving the way for further research progress.

In recent years, architectural improvements have significantly advanced image VQA. For instance, bottom-up and top-down attention (Anderson et al., [6]) combine high-level semantic and low-level visual features, multimodal transformer networks (Su et al., [7]) jointly model image and text modalities, and graph neural networks (Norcliffe-Brown et al., [8]) incorporate structured knowledge. One noteworthy model is MiniGPT-4 (Zhu et al., [9]), which utilizes a transformer-based architecture pre-trained on large multimodal databases and achieves new state-of-the-art results on several VQA benchmarks.

The subsequent VQA research wave aimed to address the dynamic input of videos. Initially, VQA datasets were designed for movies and cartoons. Some examples of such datasets include MovieQA (Tapaswi et al., [10]) which is based on plot summaries and question-answer pairs related to movies, and TGIF-QA, an animated dataset collected by Jang et al. [11]. Jang et al. also proposed a dual-state recurrent model for video QA. More recent research has prioritized VQA for longer real-world videos, exemplified by the TV-QA dataset (Lei et al., [12]) consisting of question-answer pairs related to six popular TV shows. This research has also developed spatio-temporal VQA models specifically for this dataset.

With the foundational aspects of VQA well-established, the community shifted towards domain-specific challenges. Recently, VQA for educational videos has gained increasing interest, as this domain poses additional challenges requiring complex reasoning skills. Datasets in this genre include EgoVQA (Fan, [13]) with cooking video QA pairs, How2QA (Yi et al., [14]) with science video clips and questions, and VideoQuAD (Zhong et al., [15]) compiled from university machine learning lectures. Such specificity pushes models to be more versatile and adaptive to content variations. Hierarchical recurrent models and multistep reasoning frameworks are among the techniques used so far.

The main architecture of the model is based on BLIP2[16], which is a generic and efficient pretraining strategy that bootstraps vision-language pre-training from off-the-shelf frozen pre-trained image encoders and frozen large language models. This pioneering approach is executed in two pivotal stages, each contributing to the enhancement of vision-language understanding. The initial stage centers on the symbiotic evolution of vision and language representations, while the subsequent phase fosters the generation of textual content from visual inputs.

In the initial stage, BLIP-2 orchestrates a dynamic interplay between frozen image encoders, textual queries, and language descriptions. This vision-and-language representation learning stage establishes the foundation for effective cross-modal alignment. By harnessing the strengths of the frozen image encoders, this phase catalyzes the extraction of meaningful visual features, bridging the gap between visual and textual inputs.

The second stage of BLIP-2 involves the visionary collaboration between the framework and frozen large language models (LLMs). This visionary alliance empowers the framework to unlock the art of zero-shot instructed image-to-text generation. By invoking the dormant capabilities of LLMs, BLIP-2 transcends conventional boundaries and achieves the remarkable feat of generating textual outputs guided by natural language instructions.

The advantages of BLIP2 are as follows. The far-reaching advantages of BLIP-2 resoundingly affirm its groundbreaking status in the realm of vision-language pre-training. Foremost, BLIP-2 orchestrates a harmonious fusion of frozen pre-trained image models and LLMs. The innovative integration of a Querying Transformer (QFormer) in two distinct stages—representation learning and generative learning—propels BLIP-2 to the forefront of performance. Notably, BLIP-2 attains state-of-the-art proficiency across diverse vision-language tasks, encompassing visual question answering, image captioning, and image-text retrieval.

Moreover, BLIP-2 stands as a testament to the transformative prowess of LLMs, exemplified by OPT (Zhang et al., [17]) and FlanT5 (Chung et al., [18]). With the capacity for zero-shot image-to-text generation, BLIP-2 embarks on a new era of potential, emboldening capabilities such as visual knowledge reasoning and dynamic visual conversations.

BLIP-2's computational efficiency is enhanced by its strategic use of frozen unimodal models and a lightweight QFormer. This efficiency is clearly evident when compared to other state-of-the-art methods. Notably, BLIP-2 outperforms Flamingo (Alayrac et al., [19]) by a significant 8.7% on the zero-shot VQAv2 benchmark, while using only a

fraction of the training parameters (54× fewer).

A final testament to the universality of BLIP-2 lies in its adaptability to advanced unimodal models. The extensive experimental results showcase BLIP-2's capacity to effortlessly leverage superior unimodal models, affirming its status as a versatile tool for elevating vision-language pre-training performance.

This project aims to advance VQA for educational videos by developing models based on online lectures and classes. We present a dataset, compiled from machine learning lectures and open-ended questions and answers at Yeshiva University, Katz School of Science and Health [1]. To handle complex questions about course material, our model incorporates external knowledge resources, detailed modeling of visual and textual segments, and multistep reasoning, outperforming previous methods designed for other types of video. We believe our work will facilitate deeper video understanding and question answering capabilities for the educational domain.

3. Methods

3.1. Environment Implementation

To ensure an optimal implementation of this methodology, we employed Python v3.8.17 along with PyTorch v2.0.1 and CUDA v11.8. The model training was conducted on a desktop computer with the Microsoft Windows 11 operating system, equipped with an Intel i9-13900K CPU, 64 GB RAM, and a GeForce RTX 4090 GPU with 24 GB VRAM. Since the model training process is computationally demanding, all related tasks were offloaded to the GPU to achieve optimal performance.

3.2. Data Collection

The data collection for this study encompasses three distinct stages: image and transcription collection, as well as the development of questions and answers. Each phase is critical for creating the dataset necessary for analyzing the machine learning course.

3.2.1 Image Collection

Starting with the phase of collecting images, this first step appears to be the least difficult. It entails compiling slides from the PowerPoint presentations of the machine learning course, resulting in 257 images being collected. This easy task lays the visual foundation for future analyses.

3.2.2 Transcript Collection

Transcript collection poses a significantly challenging task when moving forward. The primary aim of this phase is to transcribe the speech content from the recorded lectures on machine learning courses. The process starts by extracting audio from videos, which is necessary due to the restrictions of transcription tools that do not usually support video inputs. Subsequently, two distinct categories of transcription tools are used. The Python-based models include the Silero model [20, 21], the Wav2Vec Base model, Wav2Vec2 large-lv60 model, and the Google Speech-to-Text API [22]. Besides, professional online tools such as Cockatoo [23], Deepgram [24], Trint [25], Parrot [26], Veed [27], and Speechtext [28] are also used.

Following an exhaustive trial of more than ten different tools, the expected result is attained: professional tools outperform their Python model counterparts by a noticeable margin. The transcriptions generated by the Wac2Vec and other Python models can only be described as catastrophic, making it impossible to form coherent sentences. However, this does not imply that professional tools always deliver flawless results. Although many of their transcriptions generate accurate sentences, a considerable number still contain errors, resulting in seemingly correct but inaccurate sentences. After a thorough comparison and evaluation, Deepgram is identified as the preferred choice due to its remarkable accuracy.

However, it is important to recognize that the low quality of the initial recordings is a significant challenge. The audio is permeated with noise and even a small distance between the speaker and the microphone can cause the content to be incomprehensible. Therefore, despite being time-consuming, manual inspection of the content is a necessary part of this phase. The time commitment required for manual inspection is significant. For example, for a two-hour video, manual inspection could take up to five times or more of its length.

All the tools used in this project are listed in Table 1.

3.2.3 Creation of questions and answers

In the final stage, questions and answers are created for each slide. Capturing contextual cues that lead to the answers is essential, along with metadata such as the lecture week number and slide page. Curating ten question-answer pairs for each slide includes consistently asking, "What is the topic of this slide?" This critical inquiry aims to gather a comprehensive set of summary questions. For consistency and efficient retrieval, the sets of question-answer pairs are meticulously organized and saved in a structured JSON format. Each JSON object is governed by the schema shown in Table 2. Similar to the previous phases, this phase demands a substantial investment of time due to its complexity.

In summary, the data collection journey involves three critical phases, each of which contributes significantly to the creation of a comprehensive data set. The meticulous curation of images, the intricate process of transcription, and the

Table 1. Speech-to-Text Tools Comparison

Tool Name	Features and Capabilities	Online Platform
Silero Model	Compact, supports multiple languages	
Wav2Vec Base Model	Unsupervised learning, speech representations from raw audio	
Wav2Vec2 large-lv60 Model	Improved architecture, large vocabulary support	
Google Speech-to-Text API	Neural network models, supports 120+ languages	
Cockatoo	Fast and accurate transcriptions, handles challenging conditions	✓
Deepgram	**Highly customizable, handles complex tasks**	✓
Trint	Supports multi-speaker and multi-language	✓
Parrot	Fast and accurate transcription, streamlined process	✓
Veed	Video editing tools, automatic subtitler	✓
Speechtext	Speed and accuracy, handles audio and video files	✓

systematic generation of question-answer pairs collectively serve as the foundation for future analysis and learning.

Table 2. QA Pair JSON Schema

Field	**Description**
Instruction	The question or instruction for the QA pair
Context	The contextual information, often comprising slide or video 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 the content belongs to
Page	The page number of the slide

3.3. Dataset Summary

The distribution of data types within this dataset is outlined in Table 3. The most significant component of the dataset is made up of 2,383 Question-Answer pairs. The dataset consists of 257 transcripts that provide detailed textual records of lecture content. Additionally, there are 257 images present in the dataset, each corresponding to a corresponding lecture slide. Each lecture slide has an associated transcript.

Table 3. Breakdown of the collected data.

Data Type	**Count**
Question-Answer pairs	2383
Transcripts	257
Images	257

The dataset primarily consists of lecture slides in a visual format. Each slide is recorded as an image with a resolution of 960 by 540 pixels. As lecture slides, they carry information pertaining to the specific subjects of the lecture. These slides serve a dual purpose: they provide a visual summary and reference to textual data, while also representing the narrative of the lecture.

Our textual dataset is categorized based on the nature of questions or tasks. As illustrated in the "Category Distribution" Table 4, we have distinct categories such as closed-ended questions (closed_qa), information extraction, open-ended questions (open_qa), and more. The highest number of entries fall under the "closed_qa" category, with 992 entries, while "creative_writing" has the fewest, with 22 entries.

Table 4. Category Distribution

Category	**Number**
closed_qa	992
information_extraction	499
open_qa	398
summarization	282
classification	64
brainstorming	64
general_qa	62
creative_writing	22

Additionally, Table 5, named "Length Stats for Instruction and Response", offers a comprehensive statistical analysis concerning the length of the instructions and responses included in the dataset. On average, instructions had a word count of 11.37, while responses had a word count of approximately 51.94. The recorded standard deviation values indicate a wide variety of complexities inherent in the questions and answers presented in the dataset.

Table 5. Length Stats for Instruction and Response

Statistic	Instruction	Response
count	2383.00	2383.00
mean	11.37	51.94
std	4.13	28.78
min	4.00	1.00
25%	9.00	29.00
50%	11.00	49.00
75%	14.00	71.50
max	62.00	266.00

3.4. Data Preprocessing

The data preprocessing is critical to optimize the formatting and preparation of raw data for subsequent training and modeling.

3.4.1 Merging and Structuring

Initially, datasets consisting of question-answer pairs, transcripts, and images were integrated based on shared attributes such as week and page numbers. This integration was crucial in obtaining a unified perspective of each data point.

Following the merge, the columns in the consolidated dataset were simplified and renamed to better reflect their content. For instance, columns previously labeled as 'instruction' and 'response' were accurately retitled 'question' and 'answer', correspondingly.

3.4.2 Dataset Splitting

Partitioning data into training and validation sets is a fundamental step in the modeling process. For this project, a significant 80% of the combined data was designated for training purposes, ensuring sufficient data for model learning. The remaining 20% was reserved for validation, which critically evaluates model performance on unseen data. The precise numbers, divided into training and validation sets, are presented in Table 6.

Table 6. Distribution of samples across training and validation datasets.

Dataset	Number of Samples
Training	1916
Validation	480
Total	2396

3.4.3 Dataset Creation

To streamline data retrieval and processing, we developed a customized dataset structure. The structure is illustrated below:

- Images corresponding to each question-answer pair were retrieved and loaded. The resolution of each image was resized to 224 by 224 pixels.
- A detailed textual prompt was created by combining the lecture transcript with the question and including a specific placeholder for the model to produce the answer.
- Due to the inherent limitations of the selected model, particularly when it comes to handling long sequences, any text that exceeds the token capacity of the model is shortened carefully.
- The correct answers were converted to a format which is suitable for the model and allows for both training and validation processes.

Separate datasets were curated for training and validation purposes, utilizing a customized dataset structure.

3.4.4 Data Loading

In order to retrieve data efficiently during training, a sophisticated data loading mechanism was employed. The objective of designing this mechanism was to:

- Sets of data points should be compiled accurately.
- Padding is used to standardize the length of sequences and ensure consistency across batches.
- Different types of data elements, such as images and textual cues, should be combined in a coherent structure that is suitable for modeling purposes.

Due to hardware limitations, the data was loaded in several batches, each containing a predetermined number of data points. The batching approach not only enhanced the efficiency of the training process but also uncovered the organization and dimensions of the training data, including images, textual prompts, and correct answers.

3.5. Training Hyperparameters

To train our Visual Question Answering (VQA) system to systematically and effectively understand ML classes, we carefully selected and fine-tuned a set of hyperparameters. These choices guaranteed optimal model performance, convergence, and generalizability.

1. **Batch Size**: A batch size of 1 was chosen due to hardware limitations. The number of training examples processed before updating the model's internal parameters is determined by the batch size. While larger batch sizes can offer more accurate and stable gradient estimates, hardware limitations require compromises that may impact training speed and stability [29].

2. **Epochs**: We performed training on the dataset for a total of 150 epochs. Each epoch represents a complete forward and backward pass of all training samples. This number was chosen to ensure that the model uncovers underlying patterns without overfitting to the training data [30].

3. **Loss Function**: We chose the Cross Entropy Loss function, which is optimal for classification tasks. It effectively quantifies the difference between the true labels and the model predictions, and excels in class imbalance scenarios by giving more weight to underrepresented classes [31].

4. **Optimizer Selection and Stability**: First, we tested widely used optimizers such as Adam, AdamW, and RMSprop with different parameter configurations. Despite their well-known effectiveness in numerous deep learning applications, these optimizers exhibited instability during our specific training process. This observation led us to the Stochastic Gradient Descent (SGD) optimizer augmented with Nesterov dynamics. This combination consistently provided stability throughout the training phase. The following parameters were used:

   - Learning Rate (lr): 1×10^{-4}
   - Momentum: 0.9
   - Weight Decay (L2 regularization): 1×10^{-3}

   The reliability of the Nesterov-accelerated gradient combined with weight decay ensured a consistent and smooth convergence trajectory for our VQA system [32].

5. **Learning Rate Scheduler**: The Cosine Annealing with Warmup Scheduler was used [33]. This two-stage approach consists of

   - **Warmup Phase**: In the early stages, a linear escalation of the learning rate is observed, which promotes model stability prior to the main training phase.
   - **Cosine Annealing**: After warm-up, the learning rate is modulated using a cosine curve, allowing for a gentle reduction. This nuanced approach helps avoid premature convergence to local minima.

6. **Gradient Accumulation**: To simulate the benefits of a larger batch size without incurring the associated computational cost, we used gradient accumulation. This technique involves accumulating gradients over 16 mini-batches before performing a weight update [34].

3.6. Model Architecture

The study employs the *BLIP-2* model, a state-of-the-art multimodal architecture designed to understand and generate content from both textual and visual data. Pioneered by [16], *BLIP-2* (Bidirectional Link between Image and text model, version 2) represents a significant advancement in bootstrapping language-image pre-training. Figure 1 depicts the block architecture of the model.

We used the *BLIP-2* model (Bidirectional Link between Image and text model, version 2) was used for our experiments.. The model was initialized with 2.7 billion parameters, out of which 107,132,288 were trainable. We selectively froze certain layers of the model to optimize its performance for our specific dataset and use-case. We froze all layers of the vision model except for its last layer. To ensure stability during the training process, we kept the entire language model static.

The *BLIP-2* model was selected mainly because of its robust and versatile architecture, which integrates different modules to handle various data types. Specifically, the model at its core includes:

1. **Vision Encoder**: This is an advanced convolutional vision transformer designed for translating visual content from images into a comprehensive set of visual features. It follows the architecture of ViT or DeiT.

2. **Text Encoder**: This component is built on principles similar to those of the transformer architecture and BERT. It efficiently encodes textual data while capturing the nuances and intricacies of the language.

3. **Q-Former**: The Q-Former is a customized module of the BLIP-2 that specializes in the comprehension and encoding of questions. This ensures that the model can effectively manage Question and Answer contexts.

4. **Cross-Modal Projection**: This component acts as a crucial bridge within the model and learns to formulate joint embeddings from the visual and textual features that are processed by the earlier modules. As a result, it ensures seamless integration and understanding of the content from both visual and textual modalities.

5. **Text Decoder**: The transformer decoder is built upon the established architecture of the OPT Large Language Model. The aim of this model is to generate text that is both contextually relevant and coherent by using encoded representations from previous modules.

The brilliance of the *BLIP-2* model's brilliance stems from its ability to integrate context from various data modalities, particularly images and text, and produce precise and contextually relevant natural language responses.

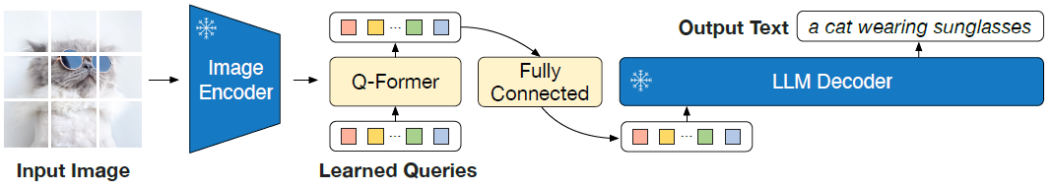

Figure 1. The pre-trained model architecture of BLIP-2 includes a frozen Image Encoder and Large Language model (LLMs), as described in the reference [16].

This modular and adaptive design of the *BLIP-2* made it a suitable choice for our research, aligning well with the multimodal nature of our dataset and the objectives of our study.

4. Results

4.1. The workflow of this study

The study process is illustrated in Figure 2. The initial data set consists of videos and slides. The videos are transformed into images and audio tracks. In addition, transcripts are extracted using Deepgram. By combining details from both the transcripts and the slides, we have formulated approximately 10 question-answer pairs for each slide. Our plan is to use a Visual Question Answering (VQA) model. This model accepts images, transcripts, and questions as input, and then produces answers to the questions posed.

4.2. Dataset preparation

In this study, we included the entirety of lectures 1, 4, 9, and 11, as well as a segment of lecture 2. This compilation resulted in a collection of 257 slides accompanied by their corresponding transcripts extracted from the videos. By merging insights from both the slides and the transcripts, we curated a comprehensive dataset containing a total of 2383 question-answer pairs, as shown in Table 7.

Week	QA pairs	Images/ Transcripts
1	880	92
2	285	25
4	598	61
9	237	38
11	383	41
Total	2383	257

Table 7. Number of data collected by week

4.3. Training result

Prior to initiating the training process, our approach was to feed the pre-trained model a combination of transcripts, images, and questions as inputs. However, the initial results, as shown in Table 8, disclosed a recurring pattern of sentences, but with a substantial inclusion of the main theme in the generated responses. This interesting finding motivated us to delve further into the fundamental mechanisms that are operating within the architecture of the model, aiming to decode the intricate interplay between the input modalities and the generation of subsequent text.

In our analysis, we present a graphical representation (see Figure 3) of the historical loss and accuracy after 150 epochs of training, which took approximately 16 hours. The left panel in the figure presents an exhaustive record of training and validation losses during the training process. On the other hand, the right panel of the same figure shows a graphical representation of the F1 scores for training and validation sets, which aids in providing a valuable perspective on the performance of the model across different stages of the training process. The model begins with a learning rate of 0 and incrementally increases it. During the initial stages, both the training and validation losses are substantially high but decrease as the epochs progress. Nevertheless, the validation loss appears to level off or even rise at some point, which might suggest the model is overfitting. The accuracy consistently remains low, indicating suboptimal performance of the model. Possible improvements might require adjustments to the architecture, learning rate, or other hyperparameters to enhance its learning and generalization abilities.

Following the training phase, the model underwent rigorous testing using 20 distinct datasets sourced from the validation set. This was done to comprehensively assess the model's generalization ability. Unfortunately, a persistent problem appeared in the form of recurring <s> tokens across all generated outputs. This study concludes with a thought-provoking inference. The model's performance was evaluated across a spectrum of validation datasets, revealing a perplexing and consistent reappearance of <s> tokens within the synthesized results. The multifaceted evaluation process consisted of data preprocessing, architectural refinements, meticulous hyperparameter tuning, and targeted post-training interventions. Through this journey, the intricate nature of this formidable challenge has been revealed. Further research is imperative to navigate this intricate enigma and expand the boundaries of model orchestration.

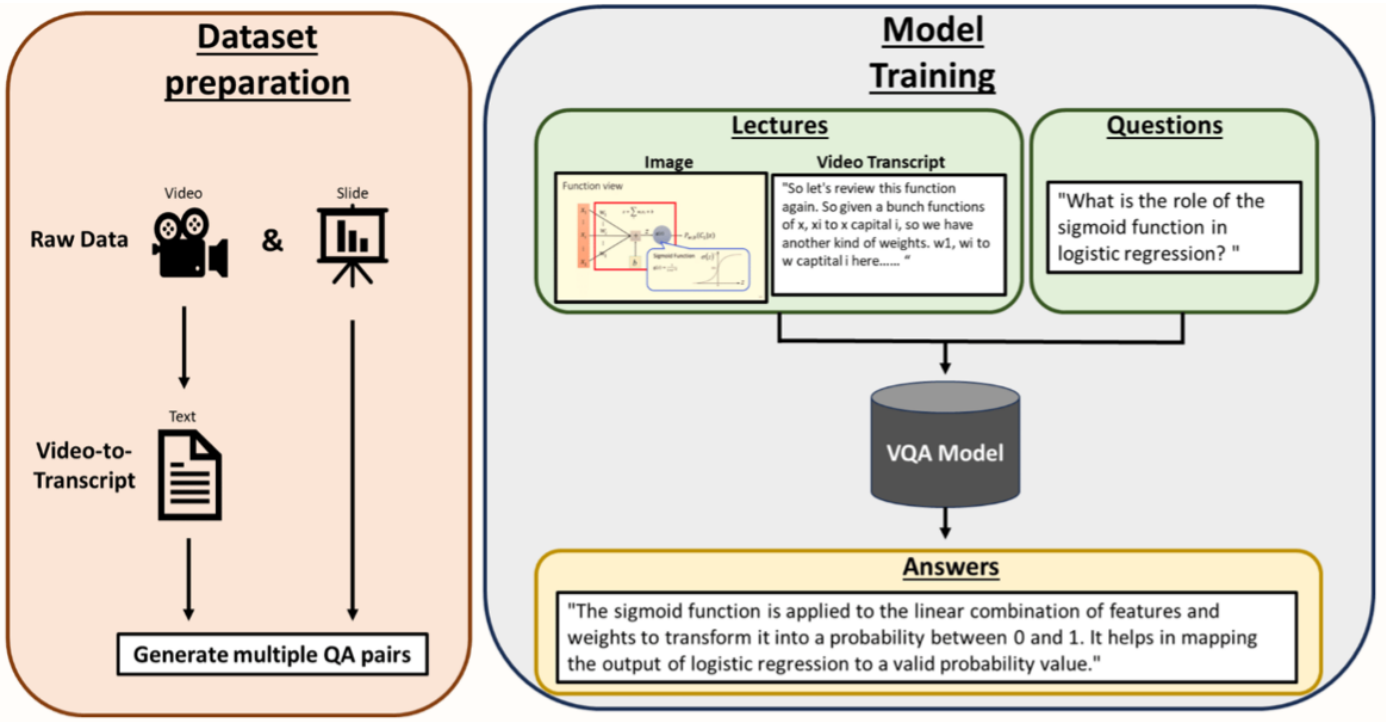

Figure 2. The workflow of this study.

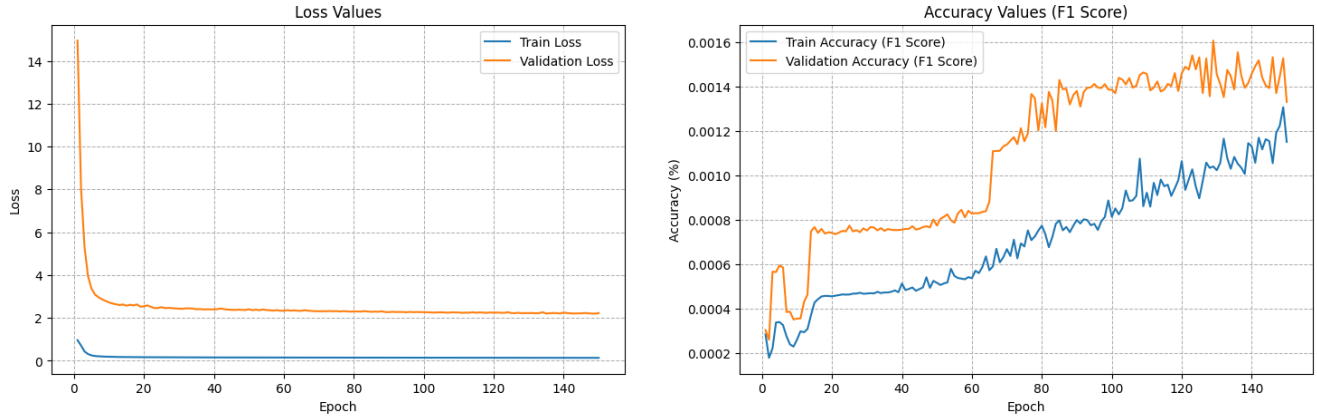

Figure 3. The presented graph shows how the historical loss and accuracy have changed over the training process.

5. Discussion

In this section, we address the challenges encountered during the development and deployment of a Visual Question Answering (VQA) model used to comprehend lecture content in video and slide format.

Complexity of Lecture Content: This study's main focus was complex topics, particularly those containing mathematical formulas and machine learning concepts. These elements, particularly the mathematical formulation, are innately difficult to transcribe, making the model's learning process more challenging. Our project is significantly more challenging due to its intricacies, unlike traditional VQA tasks that typically involve objects, actions, or straightforward textual content.

Hardware and Training Limitations: The model's architecture is strong, but it might not be sufficient for the task's complexity. The data set, with its comprehensive transcripts and intricate answers, is computationally demanding. Due to hardware limitations, we were unable to utilize the luxury of obtaining the significant computational power necessary to train a sophisticated model. Moreover, 150 epochs, even though seemingly substantial, might not be adequate for such a multifarious task.

Transcript Ambiguity: The colloquial and ambiguous

Table 8. Predicted answers before training

Question	True Answer	Generated Answer
Why does the professor ask about the meaning of zero-centered data?	The professor asks about the meaning of zero-centered data to gauge the students' understanding of this concept, which is an important preprocessing step in Machine Learning that adjusts the mean of the data to zero.	. So, the mean of the data is equal to 0. So, the mean of the
How does the professor describe the process of data normalization?	The professor describes data normalization as the process of adjusting data to fit within a certain range, often between -1 and 1, to standardize the scale of the data. This process helps with model training.	. So, the mean of the data is equal to 0. So, the mean of the
What does the professor ask about normalization and why?	The professor asks about the reasons for normalizing the data to get students thinking about and understanding the benefits and necessity of this preprocessing step in Machine Learning.	. So, the mean of the data is equal to 0. So, the mean of the
What does the professor suggest is another advantage of data normalization?	The professor suggests that one of the benefits of data normalization is that it can reduce the time it takes to train a machine learning model, as the data is scaled down and concentrated closer together.	. So, the mean of the data is equal to 0. So, the mean of the
What are the key points presented in the slide?	The key points presented in the slide include the geometric interpretation of vectors as points in space in the context of machine learning, the role of this interpretation in visualizing classification problems, and the importance of data preprocessing steps such as zero-centering and normalization.	. So, the mean of the data is equal to 0. So, the mean of the
Can you explain this slide?	This slide focuses on the concept of 'dot product' and its relationship to angles. It explains how to compute the dot product, its symmetric properties, and how it can be interpreted geometrically as the cosine of the angle between the two vectors. The slide also talks about the importance of cosine similarity in Machine Learning for measuring the similarity between two data instances.	So, this is the dot product of u and v times the cosine theta. So

nature of the transcriptions could impede the model's capability to extract useful information. Ideally, the transcriptions would provide a clear and concise depiction of the images or, at least, be condensed summaries.

Possible Causes of <s> Recurrence in Trained Model Output: The frequent repetition of <s> in the outputs of the trained models may be due to dataset inconsistencies or limitations in the model's architecture or training process.

Specialized Nature of the VQA Model: Traditional VQA models excel at understanding images. However, lecture slides often contain textual content that may be beyond the analytical capabilities of a standard VQA model.

6. Conclusion

Our attempt to apply a VQA model to lecture comprehension, while ambitious, did not yield the desired results. It became clear that the complexity of such a task, due to its inherent challenges, exceeds that of ordinary VQA undertakings.

To ensure the success of future attempts, a phased approach is recommended. Starting with more general courses, especially in the area of linguistics, may provide an easier entry point. Once a base is established, one can expand to more specialized courses.

To address the perceived inefficiency of the model in handling text, using an automated OCR tool to extract text information could be a potential solution. Integrating this extracted text as an extended data input to the model could fill the gap and improve the model's performance.

Ultimately, while the results of this study may not be exemplary, they serve as a testament to the complexity of the challenge and pave the way for future research in this area.

References

[1] Yeshiva University, Katz School of Science and Health. https://www.yu.edu/katz/ai, 2023. Accessed: 2023-08-09. 1, 3

[2] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. *CoRR*, abs/1410.0210, 2014. 1

[3] Yuke Zhu, Ce Zhang, Christopher Ré, and Li Fei-Fei. Building a large-scale multimodal knowledge base for visual question answering. *CoRR*, abs/1507.05670, 2015. 2

[4] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2016. 2

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 2

[6] 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. *CoRR*, abs/1707.07998, 2017. 2

[7] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: pre-training of generic visual-linguistic representations. *CoRR*, abs/1908.08530, 2019. 2

[8] Will Norcliffe-Brown, Efstathios Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. *CoRR*, abs/1806.07243, 2018. 2

[9] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. 2

[10] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. *CoRR*, abs/1512.02902, 2015. 2

[11] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: toward spatio-temporal reasoning in visual question answering. *CoRR*, abs/1704.04497, 2017. 2

[12] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. TVQA: localized, compositional video question answering. *CoRR*, abs/1809.01696, 2018. 2

[13] Chenyou Fan. Egovqa - an egocentric video question answering benchmark dataset. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4359–4366, 2019. 2

[14] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. *CoRR*, abs/1910.01442, 2019. 2

[15] Yaoyao Zhong, Junbin Xiao, Wei Ji, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges, 2022. 2

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 2, 6, 7

[17] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. 2

[18] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. 2

[19] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. 2

[20] Silero AI Team. Silero speech-to-text models. https://pytorch.org/hub/snakers4_silero-models_stt/, 2023. Accessed: 2023-06-30. 3

[21] Silero. Silero models: Pretrained enterprise-grade stt models. https://github.com/snakers4/silero-models, 2023. Accessed: 2023-06-30. 3

[22] Google cloud speech-to-text api. https://cloud.google.com/speech-to-text, 2023. Accessed: 2023-06-30. 3

[23] Cockatoo: Online transcription service. https://www.cockatoo.com/, 2023. Accessed: 2023-06-30. 3

[24] Deepgram. Ai-powered speech recognition. https://www.deepgram.com/, 2023. Accessed: 2023-06-30. 3

[25] Trint. Automated transcription software. https://www.trint.com/, 2023. Accessed: 2023-06-30. 3

[26] Parrot: Online transcripts. https://www.parrot.us/, 2023. Accessed: 2023-06-30. 3

[27] Veed.io: Simple online video editing. https://www.veed.io/, 2023. Accessed: 2023-06-30. 3

[28] Speechtext: Fast and accurate audio transcription service. https://www.speechtext.ai/, 2023. Accessed: 2023-06-30. 3

[29] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016. 5

[30] Lutz Prechelt. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 6

[31] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005. 6

[32] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. 6

[33] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. 6

[34] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 6