Independent Study

Evaluating Object Detection and Hallucinations in Fine-Tuned MLLM (LLaVA-1.5) Using Diverse Metrics

Instructor – Prof. Naveen Kumar

Student – Varun Chowdary Sayapaneni

Course – DSA 5990

Abstract

This research explores the fine-tuning and assessment of the Multimodal Large Language Model (MLLM) LLaVA 1.5, utilizing the OK-VQA dataset, which is tailored for Visual Question Answering (VQA) with a focus on object detection. Multimodal Large Language Models (MLLMs) have made significant progress in tasks that integrate visual and linguistic processing, such as image captioning and visual question answering, audio translation, and many more.

This report seeks to tackle these issues by fine-tuning the LLaVA 1.5 model to enhance its object detection accuracy and reduce hallucinations. The fine-tuning process is designed to diminish these hallucinations, thereby improving the model's ability to deliver precise objects in the given images and thereby reducing hallucinations.

The dataset employed comprises 9,100 training samples and 5,050 testing samples, alongside an evaluation of 250 images to gauge the model's performance. Three distinct tools—ALOHa, MMHal-Bench, and GroundingDINO—were used to evaluate the fine-tuned model's performance in detecting object hallucinations. The findings revealed a marked improvement in object detection accuracy and a reduction in hallucinations. The fine-tuned model attained an average ALOHa score of 0.886, reflecting enhanced object detection capabilities. The MMHal-Bench results indicated a hallucination score of 4.428, further validating the improvement in detection accuracy. GroundingDINO identified hallucinations in only 4% of the responses, emphasizing the model's increased reliability.

Table of Contents

1. Introduction

The rapid advancement of large language models (LLMs) has significantly influenced various natural language processing (NLP) tasks, leading to notable improvements in language understanding (Hendrycks et al., 2021; Huang et al., 2023), generation (Zhang et al., 2023; Zhu et al., 2023), and reasoning (Yu et al., 2023; Liu et al., 2023). Leveraging these advanced capabilities, multimodal large language models (MLLMs), also known as large vision-language models (LVLMs) (Liu et al., 2023; Ye et al., 2023; Dai et al., 2023), have emerged with extensive applications across multiple domains. MLLMs have shown strong performance in tasks that integrate both visual and linguistic data, such as image captioning (Li et al., 2023) and visual question answering (Liu et al., 2023; Ye et al., 2023). Prominent examples of these models include MiniGPT-4 (Dai et al., 2023), LLaVA (Liu et al., 2023), GPT-4Vo, and LLaVA-1.5 (Liu et al., 2023). These models are increasingly utilized in diverse sectors such as healthcare, defense, and education.

Despite their impressive capabilities, LVLMs encounter significant challenges, particularly in generating accurate responses—a phenomenon referred to as hallucinations. In the context of LVLMs, hallucinations occur when there is a mismatch between the actual content of images and the generated textual descriptions, similar to the hallucinations observed in traditional LLMs (Huang et al., 2023). These hallucinations can arise during various interactions or when producing outputs across different modalities, such as audio, images, or videos, complicating the reliability and accuracy of the model's outputs.

2. Objective

The objective of this study is to fine-tune the open-source Large Vision-Language Model (LVLM) LLaVA 1.5 using a publicly available dataset focused on object detection. After the fine-tuning process, the model's performance will be evaluated against 250 new images to assess its zero-shot capabilities. The results will be compared using three custom hallucination detection tools: ALOHa, MMHal-Bench, and GroundingDINO.

3. Literature Survey

3.1 Multimodal Large Language Models (MLLMs)

Vision Transformers (ViTs) (Cout, n.d.) represent a cutting-edge approach in artificial intelligence specifically designed for image processing, marking a significant shift from the traditional Convolutional Neural Networks (CNNs) that have long been the standard in this domain.

The key features of ViTs include:

1. **Transformer Architecture**: Originally developed for natural language processing (NLP) tasks, transformers excel at capturing relationships and dependencies within data by maintaining a context window. Vision Transformers (ViTs) adapt this architecture to process visual data effectively.

2. **Image Processing Method**: Unlike CNNs, which utilize localized filters (convolutions) to analyze images, ViTs divide an image into a sequence of smaller, fixed-size patches. Each patch is then flattened and transformed into an embedding, which is processed by the transformer model.

3. **Attention Mechanism**: ViTs employ the attention mechanism (Vaswani et al., 2017), allowing the model to focus on specific parts of the image sequentially, thereby understanding the relationships and contexts among various image patches.

4. **Scalability and Efficiency**: ViTs are highly scalable and benefit greatly from increased data and computational resources. They have demonstrated exceptional efficiency and accuracy, particularly in large-scale image recognition, image captioning, and visual question-answering tasks.

5. **Data-Intensive Nature**: A key challenge for ViTs is their requirement for large amounts of data to achieve optimal performance, which can be a limitation in tasks where data availability is restricted (Cout, n.d.).

3.2 Vision Transformers

Vision Transformers (ViTs) function by partitioning an image into smaller patches and creating embeddings for each flattened patch. These embeddings are subsequently processed through a transformer encoder, similar to the ones used in models like GPT, enabling the model to generate responses to prompts. Each modality, whether visual or textual, has its encoder that converts input data into corresponding tokens, allowing the model to efficiently integrate and process multimodal information (Cout, n.d.).

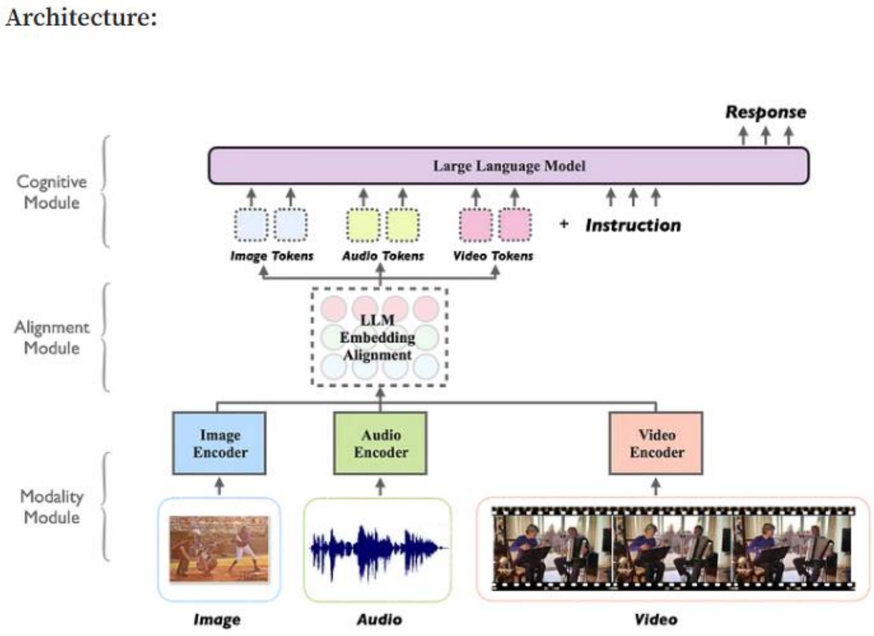


Fig 1: Architecture of ViT (Cout, n.d.)

In Figure 1, the breakdown of the image modality is as follows. The process begins by dividing the input image into fixed-size patches, typically 16x16 or 32x32 pixels. These patches are then flattened into 1D vectors, with each patch treated similarly to tokens in natural language processing (NLP) tasks. For each patch, an embedding is created by applying a linear transformation. To retain positional information, Vision Transformers (ViTs) utilize positional embeddings, which are added to each patch embedding, ensuring that the model considers spatial relationships within the image.

Once the embeddings are constructed, they are passed into a transformer encoder, often based on architectures like GPT. The transformer encoder employs self-attention mechanisms to model the relationships between the patches, enabling the network to capture both local and global features of the image. The multi-head self-attention layers process the entire image simultaneously, allowing the model to comprehend complex visual structures across all patches. The final output of the transformer encoder can then be sent through a classification head or used for downstream tasks such as object detection or segmentation, depending on the specific application. ViTs' ability to handle multiple modalities, including images, videos, and audio, highlights their flexibility compared to traditional convolutional neural networks (CNNs) in large-scale vision tasks (Cout, n.d.).

3.3 LLaVA: Large Language and Vision Assistant

LLaVA is an advanced multimodal model that integrates a vision encoder with Vicuna, an open-source model, to deliver comprehensive visual and language understanding. This model demonstrates impressive conversational abilities, derived from the multimodal capabilities of GPT-4, setting a new benchmark for accuracy in scientific question answering. LLaVA leverages GPT-4 to generate diverse, context-rich responses from images, enhancing both data quality and the model's conversational depth (Bai et al., n.d.).

The model's final 13-billion parameter checkpoint is trained on 1.2 million publicly available data points, with the training process completed in approximately one day on a single 8-A100 node. A key aspect of LLaVA's architecture is the seamless integration of visual features from CLIP with language embeddings from large language models like GPT-4. This integration is achieved through a projection matrix that effectively maps visual data into the language domain, allowing the model to process and understand multimodal inputs cohesively (Bai et al., n.d.).

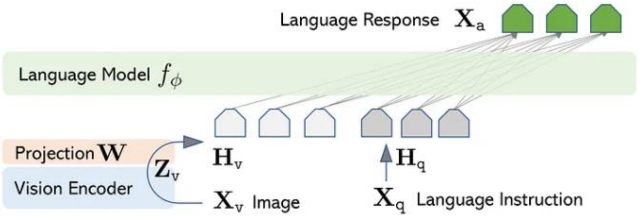


Fig 2: Architecture of LLaVA 1.5 (Bai et al, n.d.)

LLaVA's training process, illustrated in Figure 2, involves two primary stages. The first stage focuses on aligning features between the vision and language models to ensure a cohesive multimodal understanding. Following this alignment, the model undergoes fine-tuning, which is customized for specific applications like visual chat and scientific question answering, using multimodal data to boost its overall performance (Bai et al., n.d.).

3.4 Hallucination Types

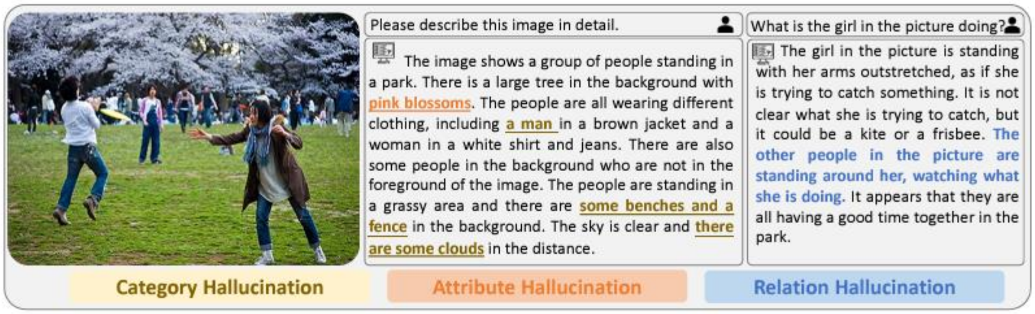
In object detection, hallucinations generally occur in two primary forms:

1. **Missing Objects**: The model fails to detect and describe objects that are actually present in the image.
2. **Nonexistent Objects**: The model describes objects that do not exist in the image or provides incorrect information about the objects that are present.

Of these, describing nonexistent objects is generally less acceptable to users. Object hallucinations can further be categorized into three specific types:

1. **Object Category**: Misidentifying or inventing the category of an object.
2. **Object Attribute**: Incorrectly describing an object’s attributes, such as color, shape, or size.

3. **Object Relation**: Misrepresenting the spatial or relational context between objects (Bai et al., n.d.).



Please describe this image in detail.

The image shows a group of people standing in a park. There is a large tree in the background with pink blossoms. The people are all wearing different clothing, including a man in a brown jacket and a woman in a white shirt and jeans. There are also some people in the background who are not in the foreground of the image. The people are standing in a grassy area and there are some benches and a fence in the background. The sky is clear and there are some clouds in the distance.

What is the girl in the picture doing?

The girl in the picture is standing with her arms outstretched, as if she is trying to catch something. It is not clear what she is trying to catch, but it could be a kite or a frisbee. The other people in the picture are standing around her, watching what she is doing. It appears that they are all having a good time together in the park.

Fig 3: Types of Object Hallucination (Bai et al., n.d.)

From Figure 3, different types of object hallucinations can be identified:

1. **Category**: MLLMs may identify object categories that do not exist or incorrectly label categories in the provided image. For example, text responses might describe "some benches and a fence" or "some clouds," even though these elements are absent in the image.

2. **Attribute**: While MLLMs may accurately identify object categories, they can incorrectly describe attributes such as color, shape, material, content, counting, or action. An example of an attribute hallucination is describing "pink blossoms" when the actual color is different.

3. **Relation**: In some cases, all objects and their attributes may be identified correctly, but the relationships between them—such as human-object interactions or their spatial arrangement—are inaccurately described. For instance, the model might describe "...standing around her, watching..." when the objects are present, but their positions or interactions do not match the actual image content (Bai et al., n.d.).

3.5 Hallucination Causes

Hallucinations in multimodal large language models (MLLMs) can stem from several factors throughout the capability acquisition process. The primary causes can be classified into four key areas:

1. **Data**: Inadequate or biased training data can result in hallucinations, as the model may learn incorrect associations or fail to generalize effectively to new, unseen data.
2. **Model**: The architecture and design of the MLLM itself can contribute to hallucinations, particularly when the model has limitations in representing and processing multimodal inputs accurately.
3. **Training**: Issues during training, such as insufficient training duration, inappropriate loss functions, or poorly tuned hyperparameters, may prevent the model from learning correct associations between different modalities.
4. **Inference**: During the inference phase, errors can occur if the model struggles to interpret and generate outputs based on the provided inputs, leading to hallucinations (Zhang et al., 2023).

3.6 Hallucination Benchmarks and Tools

Unlike standard LVLM benchmarks that assess overall capabilities, LVLM hallucination benchmarks are specifically designed to evaluate non-hallucinatory generation or to detect and discriminate against hallucinations. Various benchmarks have been developed, each using different evaluation methods. These benchmarks are generally categorized by their approach: Discriminative (Dis) or Generative (Gen) (Liu et al., n.d.).

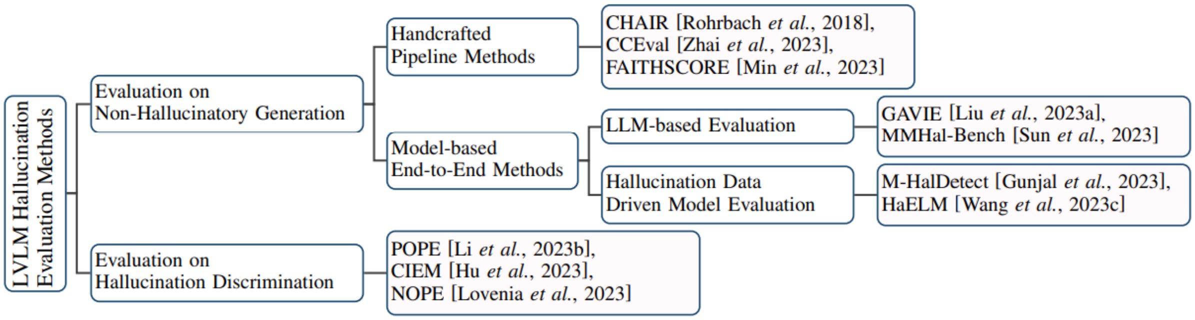

LVLM Hallucination Evaluation Methods
Evaluation on Non-Hallucinatory Generation
Handcrafted Pipeline Methods
CHAIR [Rohrbach et al., 2018], CCEval [Zhai et al., 2023], FAITHSCORE [Min et al., 2023]
Model-based End-to-End Methods
LLM-based Evaluation
GAVIE [Liu et al., 2023a], MMHal-Bench [Sun et al., 2023]
Hallucination Data Driven Model Evaluation
M-HalDetect [Gunjal et al., 2023], HaELM [Wang et al., 2023c]
Evaluation on Hallucination Discrimination
POPE [Li et al., 2023b], CIEM [Hu et al., 2023], NOPE [Lovenia et al., 2023]

Fig 4: Hallucination Evaluation Methods

Figure 4 illustrates various categories of hallucination evaluation methods, which can be broadly classified as follows:

1. **Discriminative Benchmarks**: Examples include POPE (Li et al., 2023), NOPE (Lovenia et al., 2023), and CIEM (Huang et al., 2023). These benchmarks focus specifically on object hallucinations and use accuracy as the primary evaluation metric. This accuracy is measured by querying the presence of objects in images and comparing the model's responses to ground-truth data (Liu et al., n.d.).

2. **Generative Benchmarks**: These benchmarks assess a wider range of hallucinations, including those related to object attributes and relationships (Gunjal et al., 2023; Liu et al., 2023; Ji et al., 2023; Wang et al., 2023). A notable example is AMBER (Wang et al., 2023), a comprehensive benchmark that incorporates both generative and discriminative tasks, providing a more in-depth evaluation of hallucination phenomena.

3.7 Hallucination Tools

In this study, we will be primarily working with 3 different hallucination tools.

1. MMHal-bench
2. Aloha
3. GroundingDINO

MMHal-bench

In this report, we utilized MMHal-Bench, focusing on its evaluation component to assess hallucinations in images. The original template was adjusted to fit the specific task of evaluating object hallucinations. This process required three key elements: a truth reference, a response generated by the large language model (LLM), and a directive instructing the LLM to identify and evaluate any object hallucinations present in the responses relative to the image. By modifying the format and evaluation prompts, the MMHal-Bench tool effectively facilitated the detection of object hallucinations, contributing to the overall assessment of the LLM's performance in handling image-based tasks (Liu et al., 2023).

Aloha

Aloha is a modern, open-vocabulary metric designed to utilize large language models (LLMs) for measuring object hallucinations. It employs an LLM to extract ground-truth objects from a candidate caption, assess their semantic similarity to reference objects identified in captions and object detections, and then applies matching techniques to generate a final hallucination score (ALOHa, 2024).

ALOHa enhances the reliability and localization capabilities of earlier tools, such as CHAIR, by incorporating in-context learning from LLMs and using semantically rich text embeddings for more accurate object parsing and matching. For each image caption, ALOHa provides two key measures:

1. **ALOHao:** A numeric score for each object, indicating the likelihood of the object being a hallucination.
2. **ALOHa:** An aggregated score that evaluates the overall extent of hallucinations present within the entire caption (ALOHa, 2024).

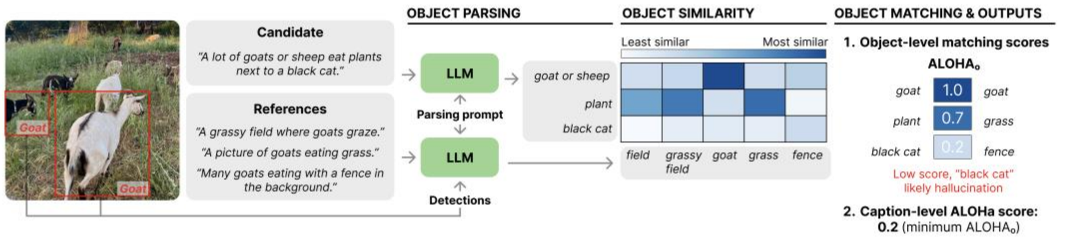

OBJECT PARSING
OBJECT SIMILARITY
OBJECT MATCHING & OUTPUTS
Candidate
"A lot of goats or sheep eat plants next to a black cat."
References
"A grassy field where goats graze."
"A picture of goats eating grass."
"Many goats eating with a fence in the background."
Goat
Goat
LLM
Parsing prompt
LLM
Detections
goat or sheep
plant
black cat
Least similar
Most similar
field
grassy field
goat
grass
fence
1. Object-level matching scores
ALOHA$_o$
goat
1.0
goat
plant
0.7
grass
black cat
0.2
fence
Low score, "black cat" likely hallucination
2. Caption-level ALOHa score:
0.2 (minimum ALOHA$_o$)

Figure 2: Overview of ALOHa. We prompt an LLM to extract visually grounded nouns from a candidate's machine-generated description and a set of references. We consider uncertain language (e.g., *"goat or sheep"*), add reference objects with and without modifiers (e.g., both *"field"* and *"grassy field"*), and avoid non-visual nouns (e.g., *"picture"* and *"background"*). Then, we compute a maximum-similarity linear assignment between candidate and reference object sets, the weights of which form the ALOHa$_o$. Matched pairs with low ALOHa$_o$ are likely hallucinations (e.g., *"black cat"*, ALOHa$_o$ $=0.2$). We additionally output the minimum ALOHa$_o$ as a caption-level ALOHa score.

Fig 5: Overview of Aloha

GroundingDINO

Grounding DINO is an advanced zero-shot object detection tool that integrates the powerful DINO architecture with grounded pre-training. It is designed to detect arbitrary objects based on human inputs, such as references, names, or specific attributes (Ahmad, n.d.).

Grounding DINO utilizes a self-supervised learning algorithm that merges DINO (DETR with Improved deNoising anchOr boxes) with grounded pre-training (GLIP). DINO is a transformer-based detection method known for its cutting-edge object detection and end-to-end optimization, eliminating the need for handcrafted modules like Non-Maximum Suppression (NMS). Meanwhile, GLIP excels in phrase grounding, effectively linking text descriptions to corresponding visual elements in images or videos, thus connecting textual descriptions with visual representations (Ahmad, n.d.).

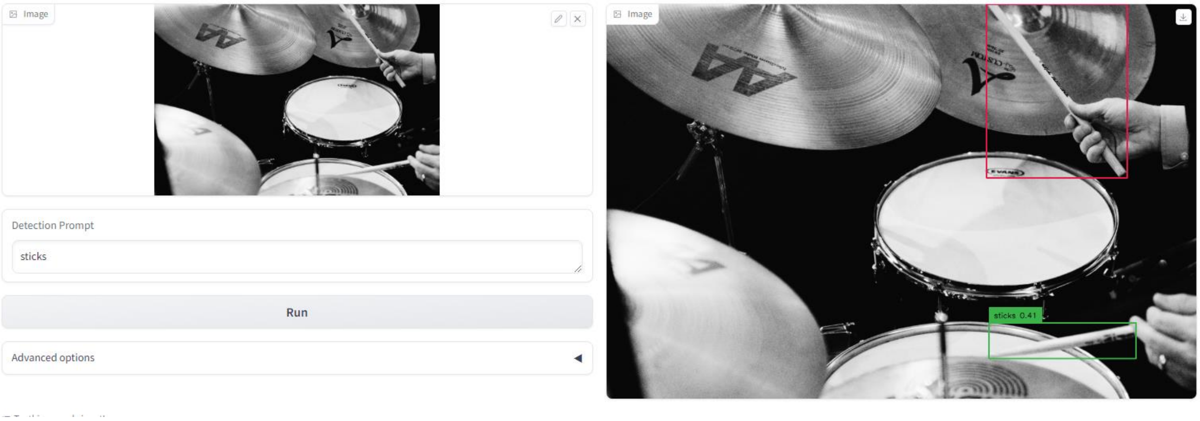
Grounding DINO is particularly effective in zero-shot object detection, aiming to generalize to unseen objects with minimal data, overcoming the limitations of traditional models. The architecture of Grounding DINO includes the following components:

1. **Image Backbone:** Extracts essential features from input images.
2. **Text Backbone:** Extracts text-based features from the corresponding descriptions.

3. **Feature Enhancer:** Fuses image and text features, enabling cross-modality information exchange.

4. **Language-Guided Query Selection:** Initializes queries based on language information.

5. **Cross-Modality Decoder:** Predicts bounding boxes using the fused features and queries (Ahmad, n.d.).



Detection Prompt

sticks

Fig 6: GroundingDINO example

Figure 6 shows an example of GroundingDINO where the object “sticks” is detected in the user-uploaded image. This example is a demo or an example that is tested on the HuggingFace platform.

We will manually evaluate all the objects generated by the model on the HuggingFace platform. If an object is present in the image, the model will detect it and provide a probability score.

4. Data

Dataset for Fine-Tuning LLaVA:

The LLaVA model will be fine-tuned using the OK-VQA dataset, which is specifically designed for Visual Question Answering (VQA) with an emphasis on object detection. The dataset is divided into two subsets:

1. Training Set (OK-VQA_train): Comprising 9,100 samples.

2. Testing Set (OK-VQA_test): Comprising 5,050 samples.

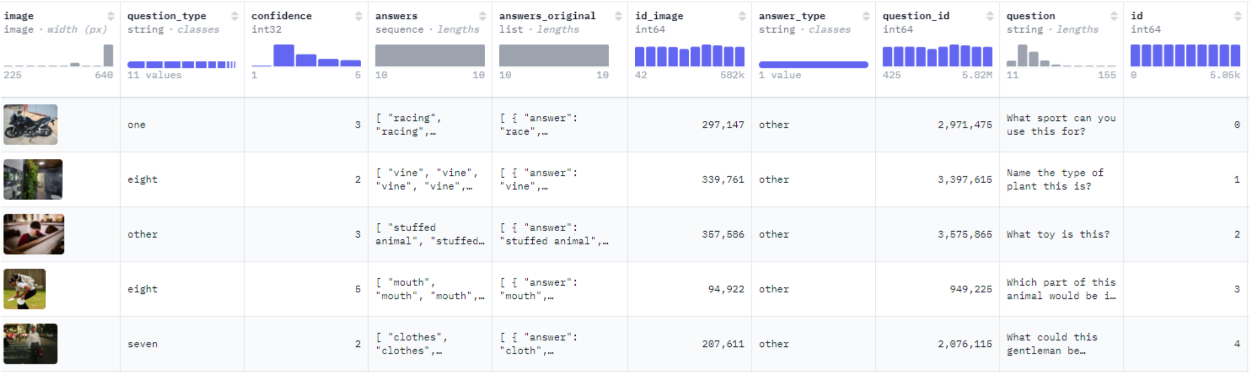


image	question_type	confidence	answers	answers_original	id_image	answer_type	question_id	question	id
image · *width (px)*	string · *classes*	int32	sequence · *lengths*	list · *lengths*	int64	string · *classes*	int64	string · *lengths*	int64
225 640	11 values	1 5	10 10	10 10	42 582k	1 value	425 5.82M	11 155	0 5.05k
	one	3	["racing", "racing",…	[{ "answer": "race",…	297,147	other	2,971,475	What sport can you use this for?	0
	eight	2	["vine", "vine", "vine", "vine",…	[{ "answer": "vine",…	339,761	other	3,397,615	Name the type of plant this is?	1
	other	3	["stuffed animal", "stuffed…	[{ "answer": "stuffed animal",…	357,586	other	3,575,865	What toy is this?	2
	eight	5	["mouth", "mouth", "mouth",…	[{ "answer": "mouth",…	94,922	other	949,225	Which part of this animal would be i…	3
	seven	2	["clothes", "clothes",…	[{ "answer": "cloth",…	207,611	other	2,076,115	What could this gentleman be…	4

Fig 7: Overview of the training dataset

All the columns of the dataset, as displayed in Figure 7, are as follows. The dataset is taken from the HuggingFace platform.

Image: Contains pictures.

Question type: Categorizes different types of questions, with 11 distinct categories.

Confidence: Indicates the difficulty level of the question.

Answers: Shortened versions of the answers_original column.

Answers_original: A list of original answers in dictionaries, each containing:

1. answers: The provided answers.
2. raw_answer: The raw form of the answer.
3. answer_confidence: The confidence level of the answer.
4. answer_id: A unique identifier for each answer.

id_image: Unique identifiers for images.

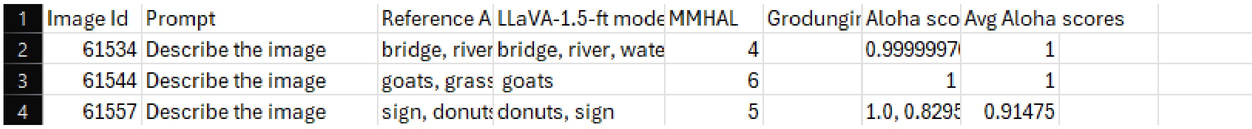
question_id: Unique identifiers for questions.

question: The question related to the image.

id: Unique identifiers for all rows.

Zero-Shot Image Testing

To evaluate the model's object detection capabilities, a collection of 250 zero-shot images, entirely distinct from the training or the testing set, was gathered. These images will be utilized to test the model across three different detection tools.



1	Image Id	Prompt	Reference A	LLaVA-1.5-ft mode	MMHAL	Grodungir	Aloha sco	Avg Aloha scores
2	61534	Describe the image	bridge, river	bridge, river, wate	4		0.9999997	1
3	61544	Describe the image	goats, grass	goats	6		1	1
4	61557	Describe the image	sign, donut	donuts, sign	5		1.0, 0.8295	0.91475

Fig 8: Overview of the data

Figure 8 provides an overview of the dataset, where the Image Id represents a unique identifier for each image. The prompt refers to the input used to query the fine-tuned model. The Reference Answer is manually gathered, while the response from the fine-tuned model is listed as LLava-1.5-ft model answer. Scores from the MMHAL-Bench, GroundingDINO, and Aloha tools are recorded as MMHAL, GroundingDINO, and Aloha scores, respectively.

5. Methodology

This study employed a systematic approach to fine-tune and evaluate the LLaVA 1.5 model, structured into two key phases:

1. Fine-Tuning the LLaVA 1.5 Model

Dataset Utilization: The LLaVA 1.5 model is fine-tuned using the OK-VQA dataset with an emphasis on object detection. This dataset ensured that the model is trained in tasks that require precise identification and description of objects within images.

Hyperparameter Tuning: The fine-tuning process involved careful hyperparameter tuning to optimize model performance. This includes selecting appropriate learning rates, batch sizes, and epochs to ensure the model is effectively trained for the target task.

2. Customizing and Applying Three Hallucination Detection Tools

Tool Selection: To evaluate the model's ability to accurately detect objects without hallucinations, three specialized tools are employed: ALOHa, MMHal-Bench, and GroundingDINO. These tools are chosen for their ability to assess different aspects of hallucination in the model's output.

Tool Customization: MMHal-Bench is tailored to improve its evaluation capabilities. This customization includes modifying the prompt template to more accurately compare and categorize objects detected by the model against the ground truth, as well as classify them on a scale from 0 to 6, enabling a thorough assessment of hallucinations.

Tool Application: Each tool is applied to the output generated by the fine-tuned model when tested on a set of 250 zero-shot images.

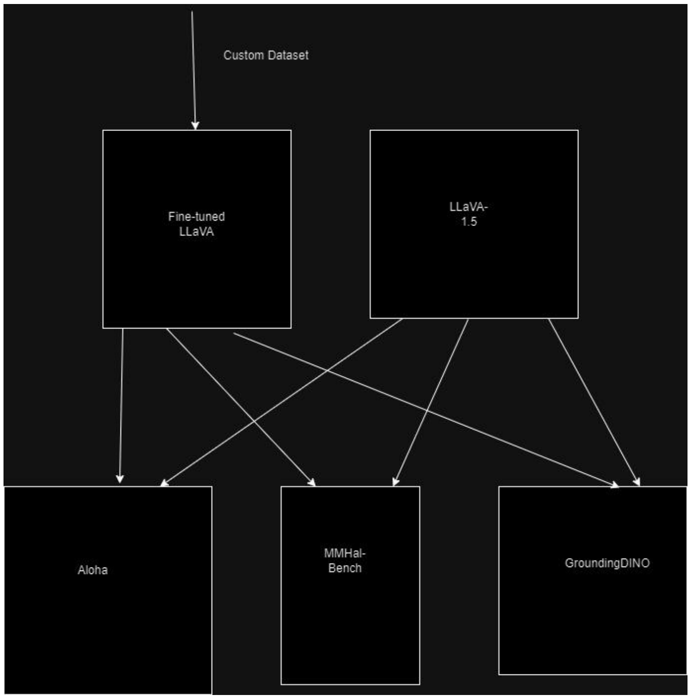


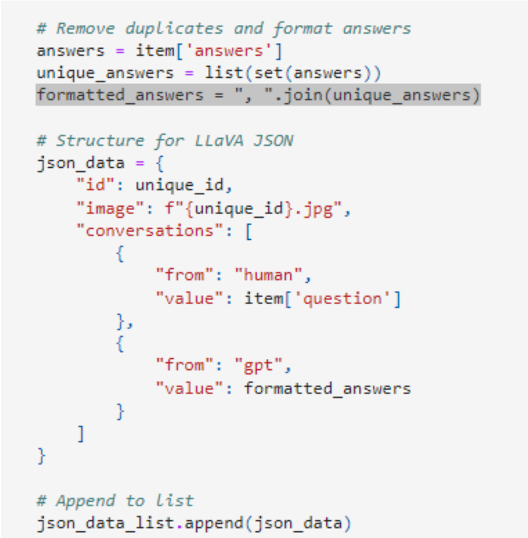
Fig 9: Overview of the Study

1. Fine-Tuning LLaVA Model

Purpose: The primary objective of fine-tuning the LLaVA 1.5 model is to detect or describe only objects present in visual images accurately. Hallucination in MLLMs poses a significant challenge, particularly in object detection, where models can be easily misled or confused. By fine-tuning the model to focus solely on object detection and presenting outputs as a list, we aim to mitigate object hallucination in MLLMs. We will also find the right hyperparameters that give us higher accuracy and other metrics.

Dataset: To achieve this goal, the model will be fine-tuned using the OK-VQA dataset, specifically designed for Visual Question Answering (VQA) with a focus on object detection.

We then take the dataset and convert it into an LLaVA fine-tuning compatible format. The format is given below, in Figure 10, and we save our formatted dataset called the 'dataset' directory.



```
# Remove duplicates and format answers
answers = item['answers']
unique_answers = list(set(answers))
formatted_answers = ", ".join(unique_answers)

# Structure for LLaVA JSON
json_data = {
    "id": unique_id,
    "image": f"{unique_id}.jpg",
    "conversations": [
        {
            "from": "human",
            "value": item['question']
        },
        {
            "from": "gpt",
            "value": formatted_answers
        }
    ]
}

# Append to list
json_data_list.append(json_data)
```

Fig 10: Format for Fine-tuning

We extract the 'unique_id', 'question', and 'formatted_answers' (which contains all the values from the answers column) and save them in the specified format for fine-tuning. The same procedure is applied to the test dataset.

Hyperparameter Selection and Tuning

The selection and tuning of hyperparameters were critical to optimizing the performance of the LLaVA 1.5 model. The key hyperparameters considered during the fine-tuning process included learning rate, batch size, and the number of epochs.

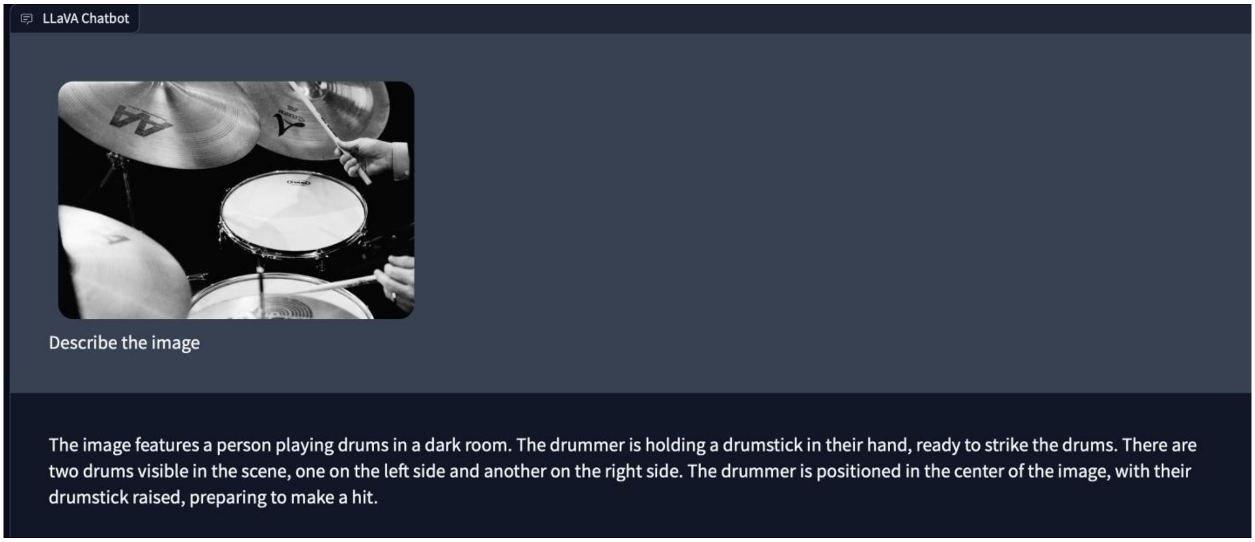
1. **Learning Rate:** The initial learning rate was set at 0.0001. A gradual decay approach was applied, reducing the learning rate as the number of epochs increased. This strategy helped prevent overfitting and ensured steady learning progress. The learning rate was halved approximately every epoch, starting from a moderately high value and decreasing as the model's performance stabilized.
2. **Batch Size:** The batch size was optimized to balance computational efficiency and the stability of the training process. Smaller batch sizes were avoided to reduce variability in the

gradient estimates, while excessively large batches were avoided to prevent memory issues and convergence problems.

3. **Epochs:** The model was trained for 5 epochs. Early stopping criteria were applied to avoid overfitting, and the log loss was closely monitored. The fine-tuning process revealed that training beyond 5 epochs led to diminishing returns, with no significant improvements in loss reduction.

Comparison of fine-tuned model vs original model

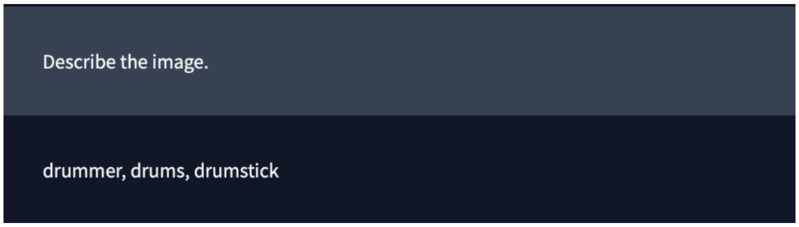
After fine-tuning and hosting our model on Gradio, we compared the output of the fine-tuned model with the original model.



Describe the image

The image features a person playing drums in a dark room. The drummer is holding a drumstick in their hand, ready to strike the drums. There are two drums visible in the scene, one on the left side and another on the right side. The drummer is positioned in the center of the image, with their drumstick raised, preparing to make a hit.

Fig 11: Original LLaVA 1.5 model demo



Describe the image.

drummer, drums, drumstick

Fig 12: Fine-tuned LLaVA 1.5 model demo

The fine-tuned model can detect only objects when prompted to describe an image as shown in Figure 12, whereas the original model tends to follow general instructions and describes the image generally as shown in Figure 11. Additionally, our model is more specific in listing various types of objects and combinations of objects compared to the original model.

Training results of Fine-tuned model

Epoch	Loss	Learning Rate
0.1	3.8382	0.0001
0.2	4.2314	0.0002
0.3	3.2841	0.0001997
0.4	2.2376	0.0001991
0.5	1.99	0.0001980
0.6	1.9696	0.0001965
0.7	1.7079	0.000194
0.8	1.6628	0.000192
0.9	1.8193	0.0001896
1.0	1.7132	0.000186
1.1	1.5226	0.000183
1.2	1.3258	0.000179
1.3	1.5440	0.000175

1.4	**1.3895**	**0.000170**
1.5	**1.2073**	**0.000165**
1.6	**1.3857**	**0.00016**
1.7	**1.1802**	**0.000155**
1.8	**1.227**	**0.000150**
1.9	**1.1097**	**0.000144**
2.0	**0.9525**	**0.000138**
2.1	**0.7680**	**0.0001321**
2.2	**0.7490**	**0.0001258**
2.3	**0.6389**	**0.000119**
2.4	**0.6445**	**0.000113**
2.5	**0.6803**	**0.000106**
2.6	**0.7108**	**0.0001**
2.7	**0.6077**	**0.00009**
2.8	**0.6856**	**0.000086**
2.9	**0.6194**	**0.00008**
3.0	**0.4654**	**0.00007**
3.1	**0.3437**	**0.000067**
3.2	**0.3490**	**0.000061**
3.3	**0.3334**	**0.000055**
3.4	**0.3175**	**0.00005**
3.5	**0.33**	**0.000044**
3.6	**0.2682**	**0.000039**
3.7	**0.2633**	**0.000034**

3.8	**0.2260**	**0.000029**
3.9	**0.3165**	**0.000024**
4.0	**0.2463**	**0.00002**
4.1	**0.1233**	**0.000016**
4.2	**0.1276**	**0.000013**
4.3	**0.1334**	**0.000010**
4.4	**0.1296**	**0.0000076**
4.5	**0.1297**	**0.0000053**
4.6	**0.1447**	**0.000003**
4.7	**0.1740**	**0.0000019**
4.8	**0.1419**	**0.0000008**
4.9	**0.1186**	**0.0000002**
5.0	**0.1035**	**0.0000000**

Table 1: Results of Fine-tuned model

Detailed Analysis

Epochs 0.1 to 1.0: The loss decreases significantly from 3.8382 to 1.7132. During this period, the learning rate is around 0.0001 to 0.000186, indicating that a moderately high learning rate helps in the initial training phases.

Epochs 1.1 to 2.0: The loss continues to drop, reaching 0.9525 by epoch 2.0, with the learning rate decreasing to around 0.000138. This indicates that reducing the learning rate further helps in achieving better loss values.

Epochs 2.1 to 3.0: The loss values drop to 0.4654, showing significant improvement, while the learning rate drops to around 0.000074. This phase shows that a lower learning rate is beneficial for further fine-tuning.

Epochs 3.1 to 4.0: The loss continues to drop to 0.2463, with a further reduced learning rate of around 0.000020. The lower learning rate continues to be effective.

Epochs 4.1 to 5.0: The loss reaches its lowest at 0.1035 by epoch 5.0, with the learning rate almost 0. This suggests that the fine-tuning phase is very effective with an extremely low learning rate.

For effective fine-tuning of the model, starting with a moderate learning rate and gradually decreasing it as training progresses is key. Training for around 5 epochs should be sufficient, with the learning rate halving approximately every epoch. This approach ensures the model steadily improves and avoids overfitting, leading to optimal performance. Training over 5 epochs with log loss almost equal to 0, has stabilized the performance of the model.

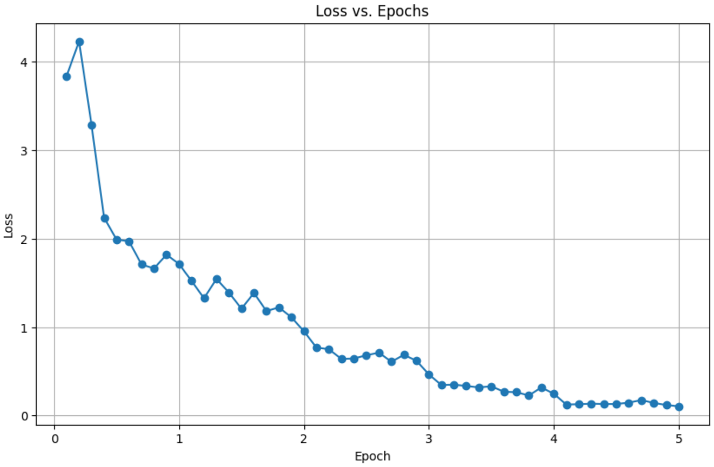


Fig 13-a: Loss vs Epoch

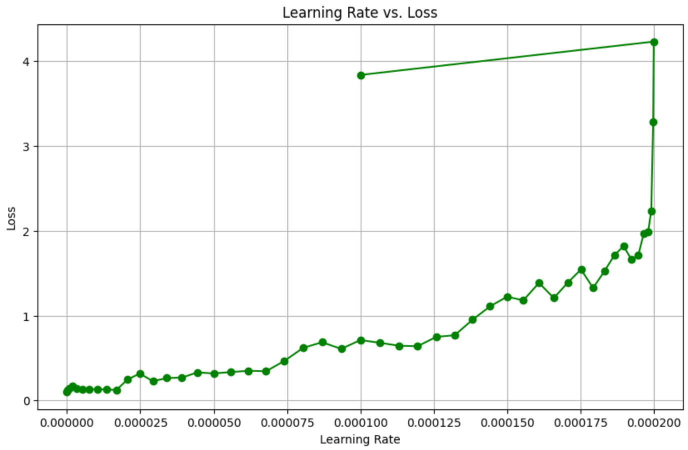


Fig 13-b: Learning rate vs Loss

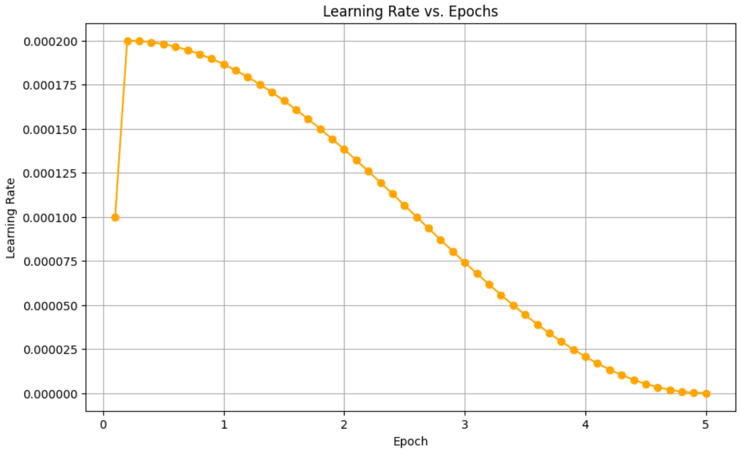


Fig 13-c: Learning rate vs Epochs

Figure 13-a (Loss vs. Epochs): This graph displays the model's loss as it progresses through training epochs. Initially, the loss decreases significantly, indicating that the model is learning and improving its performance. As the epochs continue, the rate of improvement slows, and the loss stabilizes, suggesting that the model is approaching optimal performance and further training yields diminishing returns.

Figure 13-b (Learning Rate vs. Loss): This graph illustrates the relationship between the learning rate and loss during training. Initially, a higher learning rate is associated with a rapid decrease in loss, but as the learning rate decreases, the model's loss continues to improve at a slower pace. This shows that reducing the learning rate over time helps fine-tune the model, improving its performance and stabilizing the loss towards the end of training.

Figure 13-c (Learning Rate vs. Epochs): This graph depicts the learning rate as it changes across epochs. It starts with a relatively high learning rate, which gradually decreases over time. This controlled reduction in learning rate helps the model refine its parameters more accurately, allowing for effective training in the early stages and fine-tuning in later epochs for better performance without overshooting optimal solutions.

Confusion Matrix

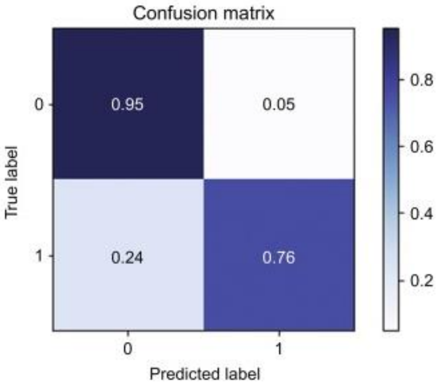


Fig 14: Confusion Matrix

The provided confusion matrix, in Figure 14, demonstrates the performance of a binary classification model. The model's predictions for object labels were compared with the actual labels from the test data (OK-VQA_test). If the predicted label did not match any object in the test list, it was classified as 0 (negative). The matrix is organized as follows:

1. **True Negative (TN):** 95% (Top left) – The model correctly predicted the negative class.
2. **False Positive (FP):** 5% (Top right) – The model incorrectly predicted the positive class.
3. **False Negative (FN):** 24% (Bottom left) – The model incorrectly predicted the negative class.

4. **True Positive (TP):** 76% (Bottom right) – The model correctly predicted the positive class.

Precision, Recall, F1 Score, and Accuracy

1. **Precision** (Positive Predictive Value)**:**

Precision=TPTP+FP=0.938 or 93.8%

2. **Recall** (Sensitivity or True Positive Rate)**:**

Recall=TPTP+FN=0.76 or 76%

3. **F1 Score** (Harmonic Mean of Precision and Recall)**:**

F1 Score=≈0.839

4. **Accuracy:**

Accuracy=0.855 or 85.5%

2. Customizing and Applying Three Hallucination Detection Tools on 250 Images

Hallucination Detection Tools

1. **ALOHa**
2. **MMHal-Bench**
3. **GroundingDINO**

Testing the Model Using Hallucination Detection Tools

To evaluate the performance of the fine-tuned LLaVA 1.5 model, a customized dataset consisting of 250 images, each accompanied by corresponding questions and reference truths. The objects within this image dataset were manually labeled to ensure accuracy. These images, along with their associated questions or prompts, were then processed by the fine-tuned LLaVA 1.5 model.

The responses generated by the model were subsequently analyzed using the three selected hallucination detection tools: ALOHa, MMHal-Bench, and GroundingDINO. These tools were employed to assess how effectively the model performs in object detection, particularly in identifying whether the detected objects are accurately represented or if hallucinations are present. This testing process provides a comprehensive evaluation of the model's capabilities in accurately detecting objects within visual inputs.

Aloha

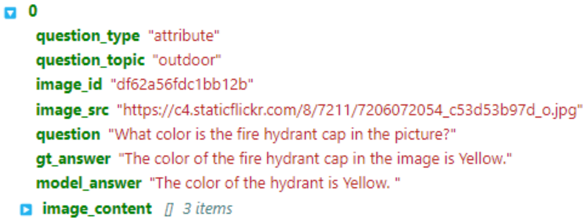
ALOHa was installed locally to test the outputs of the responses generated by our fine-tuned model compared to the original LLaVA 1.5 model. By running the test script with our unique OpenAI API key and setting the Object parser to GPT35TurboObjectParser and similarity measure to MPNetSimilarity, provided both candidate and reference captions, representing the LVLM-generated responses and the ground truth, respectively.

Upon executing the test script on the responses generated by LLaVA against the reference truth, we observed the following results:

The object similarity score ranges from 0 to 1. A score closer to or more than 1 indicates that the object matches the reference truth, suggesting the object is present in the image. Conversely, a score closer to 0 indicates a higher probability of a hallucinated answer.

MMHal-Bench

The Data is given in the following format in Figure 15.



```
0
   question_type "attribute"
   question_topic "outdoor"
   image_id "df62a56fdc1bb12b"
   image_src "https://c4.staticflickr.com/8/7211/7206072054_c53d53b97d_o.jpg"
   question "What color is the fire hydrant cap in the picture?"
   gt_answer "The color of the fire hydrant cap in the image is Yellow."
   model_answer "The color of the hydrant is Yellow. "
 image_content [] 3 items
```

Fig 15: Format of MMHal-bench

After modifying the template and adjusting some evaluation methods in MMHal-Bench, we collected hallucination scores for 250 different images using both the fine-tuned model. The results indicate a slight decrease in the hallucination rate for the fine-tuned model compared to the original. The data has been formatted to align with LLM analysis, using a scoring scale where:

0 – Not informative, with hallucinations

…

6 – Very informative and no hallucination

GroundingDINO

The GroundingDINO object detection tool identifies object labels from the fine-tuned model. We manually reviewed and detected all the objects for both models to assess how accurately each model predicted the objects.

6. Results

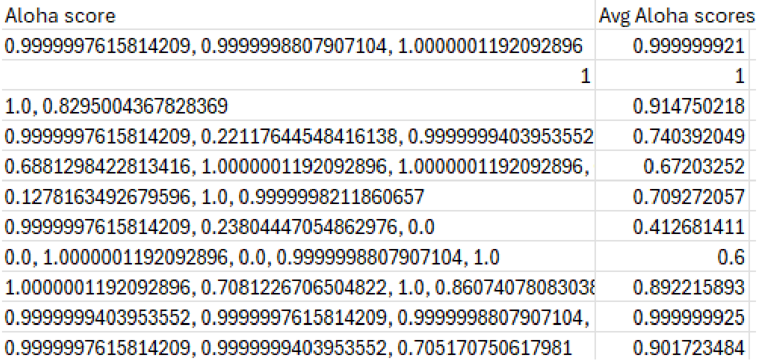
Fine-tuned model performance

The fine-tuned LLaVA 1.5 model demonstrates a strong overall performance, achieving an accuracy of 85.5%, indicating that it correctly predicts the classes in most cases. The model's precision is notably high at 93.8%, signifying that when it predicts a positive instance, it is highly likely to be correct. However, the model's recall is relatively lower at 76%, suggesting that while it is precise, it does miss some positive instances. The F1 score, which balances both precision and recall, stands at 83.9%, reflecting a well-rounded performance in object detection tasks.

During the fine-tuning process, training the model for 5 epochs proved to be optimal, resulting in a log loss of 0.1035. This low log loss indicates that the model has effectively learned from the data, achieving high accuracy and precision. Beyond 5 epochs, the log loss stabilizes, indicating that further training would not significantly enhance the model's performance. This stability suggests that the model has reached a point of diminishing returns, where additional training offers minimal improvement.

Overall, the fine-tuned model performs effectively in detecting objects within visual images, demonstrating a balanced trade-off between precision and recall. The findings suggest that the model is well-suited for tasks that require accurate and reliable object detection.

Fine-tuned model on Aloha



Aloha score	Avg Aloha scores
0.9999997615814209, 0.9999998807907104, 1.0000001192092896	0.999999921
1	1
1.0, 0.8295004367828369	0.914750218
0.9999997615814209, 0.22117644548416138, 0.9999999403953552	0.740392049
0.6881298422813416, 1.0000001192092896, 1.0000001192092896,	0.67203252
0.1278163492679596, 1.0, 0.9999998211860657	0.709272057
0.9999997615814209, 0.23804447054862976, 0.0	0.412681411
0.0, 1.0000001192092896, 0.0, 0.9999998807907104, 1.0	0.6
1.0000001192092896, 0.7081226706504822, 1.0, 0.86074078083038	0.892215893
0.9999999403953552, 0.9999997615814209, 0.9999998807907104,	0.999999925
0.9999997615814209, 0.9999999403953552, 0.705170750617981	0.901723484

Fig 16: Aloha scores

The fine-tuned LLaVA 1.5 model achieved an average ALOHa score of 0.886 across 250 different examples. This high score reflects the model's enhanced effectiveness in accurately detecting objects while significantly minimizing hallucinations. The results indicate that the fine-tuning process has successfully improved the model's ability to deliver precise and reliable outputs in visual question-answering tasks. Some of the Aloha results are shown in Figure 16.

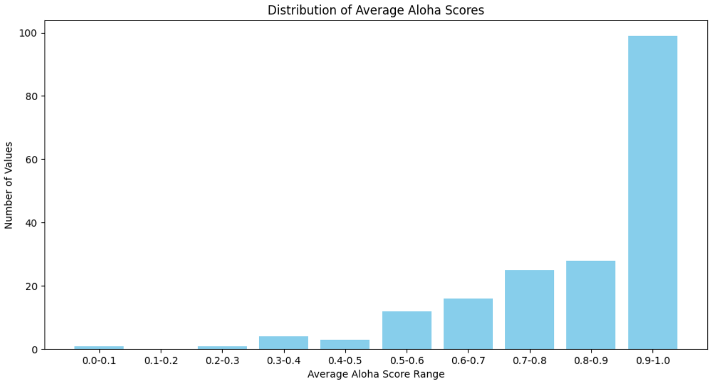


Fig 17: Distribution of Avg Aloha Values

Most of the ALOHa scores fall within the 0.9 to 1.0 range, highlighting the model's exceptional performance in many cases as shown in Figure 17. There are also smaller, yet notable, counts in the 0.7 to 0.8 and 0.8 to 0.9 ranges, which further demonstrate the model’s strong overall performance. The lower score ranges are sparsely populated, indicating that the model rarely performs poorly. Overall, this distribution reflects a high-performing model with a consistent tendency toward achieving near-perfect object detection after fine-tuning.

Fine-tuned model on MMHal-Bench

The fine-tuned model achieved an average hallucination score of 4.428, indicating a marginal improvement in its ability to accurately detect objects and reduce hallucinations compared to the original model. A score of 4.428 falls between 4, which signifies responses that are somewhat informative with no hallucinations, and 5, which denotes very informative with no hallucinations. This result suggests that fine-tuning has enhanced the model’s reliability in producing accurate and informative outputs. Most of the results are in 3-4 and 4-5 categories as shown in Figure 18.

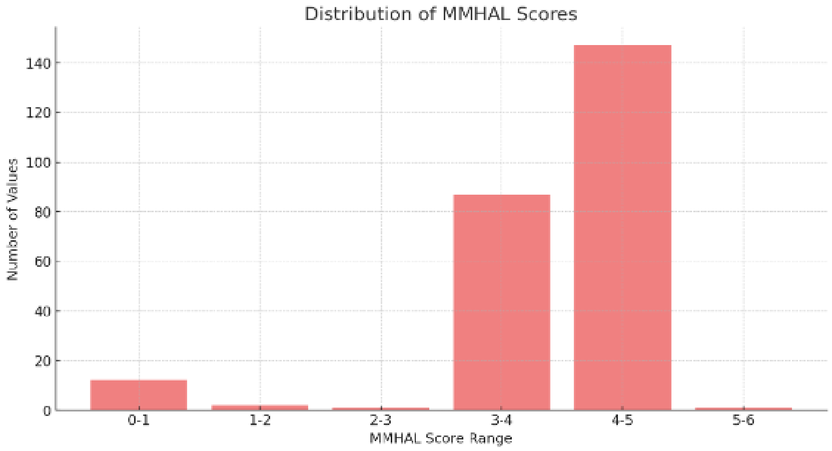


Fig 18: Distribution of MMHAL Scores

Fine-tuned model on GroundingDINO

The analysis revealed that only a few object labels generated by the fine-tuned model were not present in their respective visual images. Specifically, the fine-tuned model incorrectly identified 10 object labels that were not present in the images. Overall, only 4% of the responses contained one or more object-label hallucinations, indicating a relatively low rate of error in the fine-tuned model's predictions.

7. Conclusion

The evaluation of our fine-tuned LLaVA 1.5 model, utilizing three distinct tools—ALOHa, MMHal-Bench, and GroundingDINO—demonstrates the model's superior performance in object detection and its ability to minimize hallucinations effectively.

ALOHa: The fine-tuned model achieved an impressive average score of 0.886 across 250 examples, indicating its strong capability to detect objects accurately while significantly reducing the rate of hallucinations. The majority of the scores fell within the 0.9 to 1.0 range, reflecting the model’s consistent and high-level performance in most cases.

MMHal-Bench: The model scored an average of 4.428, which represents a marked improvement in object detection capabilities compared to the original model. This score suggests that the fine-tuned model is better at distinguishing between real and hallucinated objects, providing more reliable outputs. The customization of MMHal-Bench allowed for a more refined evaluation, further confirming the model's enhanced accuracy.

GroundingDINO: Using the GroundingDINO tool, the analysis identified only a small number of hallucinated object labels—specifically, 10 labels out of 250 examples that were not present in the visual images. This low error rate, with only 4% of responses containing one or more hallucinated object labels, underscores the model's improved reliability in object identification and its ability to generalize well to new data.

Overall, the fine-tuned LLaVA 1.5 model demonstrated consistent performance across all three evaluation tools, confirming its effectiveness in reducing hallucinations and improving object detection accuracy. These enhancements are particularly crucial for applications requiring precise visual and language understanding, ensuring that the model delivers reliable and accurate outputs in real-world scenarios. The results underscore the value of fine-tuning a general-purpose model for specific tasks, highlighting the potential for significant improvements in performance through targeted training. By refining the model to address specific challenges, such as hallucinations, we can achieve better results that align with the demands of various applied fields.

8. Future Scope

The development of more robust frameworks and diverse evaluation methods is essential to further enhance the performance of multimodal large language models (MLLMs) and effectively address the issue of hallucinations. Current tools and benchmarks, while effective, have limitations that need to be addressed to capture the full spectrum of hallucination types. Advancements in evaluation methodologies can lead to more comprehensive assessments of MLLMs, enabling the identification of subtle and complex hallucination patterns that existing tools might miss.

In addition to refining evaluation techniques, there is a need for more sophisticated frameworks that can integrate various modalities more seamlessly. These frameworks should be capable of dynamically adjusting to different types of data and tasks, reducing the likelihood of hallucinations by better aligning the model's outputs with the contextual information provided. Incorporating advanced machine learning techniques, such as self-supervised learning and reinforcement learning, could further improve the model's ability to generate accurate and contextually relevant responses, thereby enhancing its overall reliability and applicability in real-world scenarios.

9. References

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021). Measuring massive multitask language understanding. In *9th International Conference on Learning Representations (ICLR 2021), Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=d7KBjmI3GmQ

Huang, Y., Bai, Y., Zhu, Z., Zhang, J., Zhang, J., Su, T., Liu, J., Lv, C., Zhang, Y., Lei, J., & others. (2023). C-eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. *ArXiv Preprint*, abs/2305.08322. https://arxiv.org/abs/2305.08322

Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., & Hashimoto, T. B. (2023). Benchmarking large language models for news summarization. *ArXiv Preprint*, abs/2301.13848. https://arxiv.org/abs/2301.13848

Zhu, W., Liu, H., Dong, Q., Xu, J., Kong, L., Chen, J., Li, L., & Huang, S. (2023). Multilingual machine translation with large language models: Empirical results and analysis. *ArXiv Preprint*, abs/2304.04675. https://arxiv.org/abs/2304.04675

Yu, F., Zhang, H., & Wang, B. (2023). Nature language reasoning, a survey. *ArXiv Preprint*, abs/2303.14725. https://arxiv.org/abs/2303.14725

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. *ArXiv Preprint*, arXiv:2304.08485.

Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., & others. (2023). mplug-owl: Modularization empowers large language models with multimodality. *ArXiv Preprint*, arXiv:2304.14178.

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., & Hoi, S. (2023). InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *ArXiv Preprint*, arXiv:2305.06500.

Li, J., Li, D., Savarese, S., & Hoi, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv Preprint*, arXiv:2301.12597.

Analyzing and mitigating object hallucination in large vision-language models. (2023). *In NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Liu, H., Li, C., Wu, Q., & others. (2023). Visual instruction tuning. *In NeurIPS 2023*.

Liu, H., Li, C., Li, Y., & others. (2023). Improved baselines with visual instruction tuning. *ArXiv Preprint*, arXiv:2310.03744.

Huang, L., Yu, W., Ma, W., & others. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ArXiv Preprint*, arXiv:2311.05232.

Cout, S. (n.d.). Exploring multimodal large language models: A step forward in AI. *Medium*. https://medium.com/@cout.shubham/exploring-multimodal-large-language-models-a-step-forward-in-ai-626918c6a3ec

Bai, Z., Wang, P., Xiao, T., He, T., Han, Z., Zhang, Z., & Shou, M. Z. (n.d.). Hallucination of multimodal large language models: A survey. *ArXiv Preprint*. https://arxiv.org/abs/2404.18930

Liu, H., Xue, W., Chen, Y., Chen, D., Zhao, X., Wang, K., Hou, L., Li, R., & Peng, W. (n.d.). A survey on hallucination in large vision-language models. *ArXiv Preprint*, arXiv:2402.00253. https://arxiv.org/abs/2402.00253

Rohrbach, A., Tapaswi, M., Torabi, A., Maharaj, T., Rohrbach, M., Fidler, S., Pal, C., & Schiele, B. (2017). The joint video and language understanding workshop: MovieQA and the large-scale movie description challenge (LSMDC).

Li, Y., Du, Y., Zhou, K., & others. (2023). Evaluating object hallucination in large vision-language models. *In EMNLP 2023*.

Lovenia, H., Dai, W., Cahyawijaya, S., & others. (2023). Negative object presence evaluation (NOPE) to measure object hallucination in vision-language models. *ArXiv Preprint*, arXiv:2310.05338.

Huang, L., Yu, W., Ma, W., & others. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ArXiv Preprint*, arXiv:2311.05232.

Gunjal, A., Yin, J., & Bas, E. (2023). Detecting and preventing hallucinations in large vision-language models. *ArXiv Preprint*, arXiv:2308.06394.

Liu, F., Lin, K., Li, L., & others. (2023). Mitigating hallucination in large multi-modal models via robust instruction tuning. *ArXiv Preprint*, arXiv:2306.14565.

Ji, Z., Lee, N., Frieske, R., & others. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12).

Wang, J., Zhou, Y., Xu, G., & others. (2023). Evaluation and analysis of hallucination in large vision-language models. *ArXiv Preprint*, arXiv:2308.15126.

Wang, J., Wang, Y., Xu, G., & others. (2023). An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation. *ArXiv Preprint*, arXiv:2311.07397.

Ahmad, T. (n.d.). Zero-shot object detection with GroundingDINO. *Medium*. https://medium.com/@tauseefahmad12/zero-shot-object-detection-with-grounding-dino-aefe99b5a67d

ALOHa: A new measure for hallucination in captioning models. (2024). *ArXiv Preprint*, arXiv:2404.02904. https://arxiv.org/pdf/2404.02904

Ahmad, T. (n.d.). Zero-shot object detection with GroundingDINO. *Medium*. https://medium.com/@tauseefahmad12/zero-shot-object-detection-with-grounding-dino-aefe99b5a67d

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *ArXiv Preprint*, arXiv:1706.03762.

Liu, H., et al. (2023). GroundingDINO: Marrying DINO with grounded pre-training for open-set object detection. *ArXiv Preprint*, arXiv:2303.05499.

Alignment for multimodal models with factually augmented RLHF. (2023). *ArXiv Preprint*, arXiv:2309.14525.

10. Self-Assessment

Instructor: Dr. Naveen Kumar

Credit Hours: 3

For this project, my primary learning objectives were:

1. To fine-tune an advanced Large Vision-Language Model (LVLM) for enhanced object detection.
2. To gain hands-on experience with state-of-the-art hallucination detection tools.
3. To deepen my understanding of multimodal large language models (MLLMs) and their applications.
4. To enhance my technical skills in model evaluation and statistical analysis.

I am pleased to report that I successfully met these objectives, significantly improving my technical proficiency and deepening my understanding of LVLMs and their real-world applications.

One of my key goals was to fine-tune the LLaVA 1.5 model, focusing on improving its object detection capabilities while minimizing hallucinations. Through this process, I gained valuable experience in hyperparameter tuning, dataset management, and the practical challenges of model fine-tuning. This task not only improved my skills in working with complex AI models but also provided insights into the intricacies of balancing accuracy and reliability in AI outputs.

Another critical objective was to utilize and customize three specific hallucination detection tools: ALOHa, MMHal-Bench, and GroundingDINO. Engaging with these tools allowed me to develop a more nuanced understanding of how to assess and mitigate hallucinations in LVLMs. The process of customizing MMHal-Bench to better evaluate hallucinations was particularly enlightening, as it required both technical knowledge and creative problem-solving to effectively adapt the tool to the project’s needs.

My project also allowed me to delve deeper into the architecture and functioning of multimodal large language models. By working with LLaVA 1.5, I explored how vision transformers and language models can be integrated to enhance multimodal understanding. This experience enriched my knowledge of AI model architecture, fine-tuning a model for a specific purpise and the challenges associated with processing multimodal data.

Applying advanced AI techniques to a real-world task was highly rewarding. The practical application of these techniques, particularly in improving model accuracy and reliability, reinforced the importance of thorough evaluation and iterative refinement in AI development. This hands-on experience solidified my understanding of how theoretical knowledge translates into practical results.

Lastly, this project provided significant opportunities to enhance my technical and analytical skills. Each phase of the project, from model fine-tuning to detailed statistical analysis, presented unique challenges that required strategic thinking and technical expertise. Overcoming these challenges has been a crucial part of my professional development, equipping me with the skills and confidence to tackle complex AI projects in the future.

In conclusion, this project has been an invaluable learning experience that has substantially contributed to my growth as a data science and AI professional. It has prepared me to address the challenges of fine-tuning advanced models and applying AI to solve real-world problems, making me better equipped to contribute to the field of AI and its ethical applications.