Independent Study

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

Instructor – Prof. Naveen Kumar

Student – Varun Chowdary Sayapaneni

Introduction –

The advancement of large language models (LLMs) has significantly impacted various natural language processing (NLP) tasks, demonstrating remarkable improvements in language understanding [1, 2], generation [3, 4], and reasoning [5, 6]. Utilizing the advanced capabilities of state-of-the-art LLMs, multimodal large language models (MLLMs) [6, 7, 8], also known as large vision-language models (LVLMs), have emerged with extensive applications and capabilities. MLLMs exhibit strong performance in multimodal tasks such as image captioning [9] and visual question answering [6, 8], among other applications. Notable examples of current vision-language models include MiniGPT-4 [10], LLaVA [11], GPT-4Vo, and LLaVA-1.5 [12]. These MLLMs can be utilized across diverse domains, including medical, defense, and education sectors.

Despite the impressive capabilities achieved by existing LVLMs, a significant challenge remains in providing accurate responses to specific tasks, a phenomenon known as hallucinations. In the context of LVLMs, hallucination refers to the mismatch between the factual content of images and the generated textual content, like the textual hallucination observed in LLMs [13]. Hallucinations in MLLMs can occur during interactions or when generating outputs across various modalities such as audio, images, or videos.

Multimodal Large Language Models (MLLMs) Architecture:

Vision Transformers (ViTs) [14] are a cutting-edge a kind of AI model artificial intelligence model specifically designed for image processing. They represent a significant departure from the traditional Convolutional Neural Networks (CNNs) that have been the standard in image processing for many years. The key aspects of ViTs are outlined below:

1. Transformer Architecture: Initially developed for NLP tasks, transformers excel at capturing relationships and dependencies within data by maintaining a context window. Vision Transformers adapt this architecture for visual data processing.

2. Image Processing Method: Unlike CNNs, which use localized filters (convolutions) to process 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 fed into the transformer model.

3. Attention Mechanism: The transformer uses the attention mechanism [29], which enables the model to focus on parts of the image in a sequence while trying to understand the relations and the contexts between various image patches.

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

5. Applications: Vision Transformers are versatile and can be applied in various areas or fields, such as image classification, object detection, and medical image analysis, including understanding MRI scans, X-rays, and charts.

6. Data-Intensive Nature: One of the challenges with ViTs is their need for large amounts of data to achieve optimal performance. This data intensity can be problematic for tasks with limited available data.

The architecture:

Vision Transformers (ViTs) function by dividing images into patches and generating embeddings for each flattened patch. These embeddings are then processed through a transformer encoder, similar to those used in models like GPT, to produce responses to the given prompts.

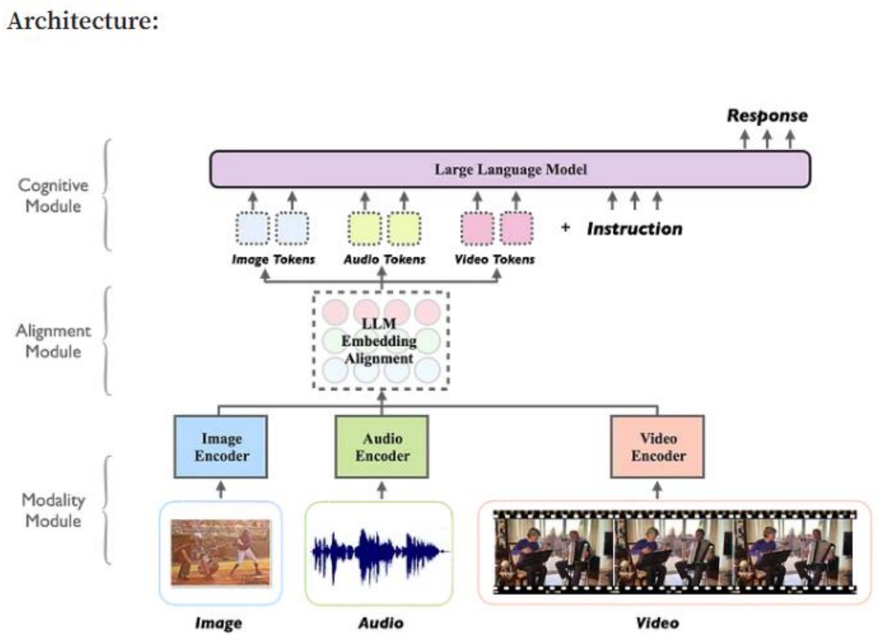


Fig: 1 - Architecture of ViT

LLaVA: Large Language and Vision Assistant

LLaVA is an innovative completely trained large multimodal model that integrates a vision encoder with Vicuna to achieve general-purpose visual and language understanding. This model demonstrates impressive chat capabilities derived from the multimodal GPT-4, setting a new benchmark in accuracy for Science QA. LLaVA utilizes GPT-4 to generate diverse and context rich response sets from images, which not only enhances data quality but also imbues the model with a deeper and conversational agent[26].

The final 13B checkpoint of LLaVA uses only 1.2 million publicly available data points and completes full training in approximately one day on a single 8-A100 node. Central to LLaVA's architecture is the integration of visual features from CLIP with language embeddings from LLMs like GPT-4. This is achieved through a projection matrix that attaches visual data into the language domain, enabling the models to process and comprehend multimodal inputs seamlessly.

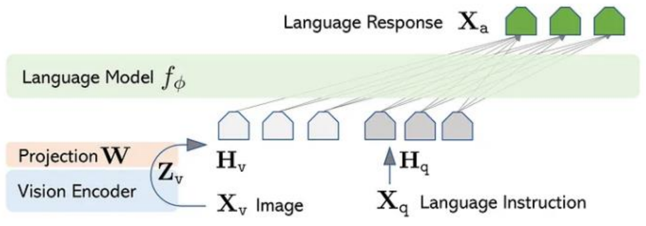


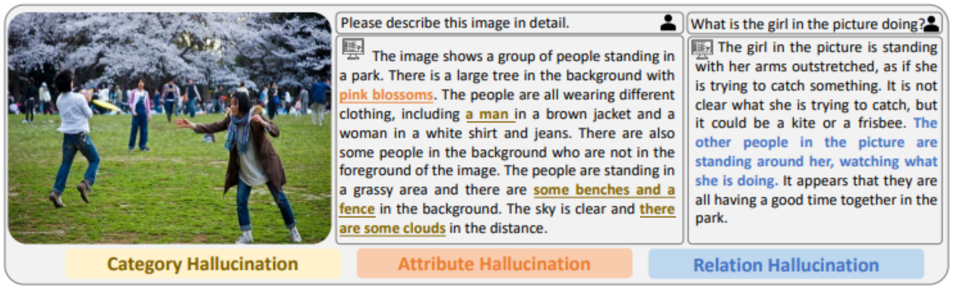
Fig 2: Architecture of LLaVA 1.5

LLaVA's training process involves two stages. Initially, it aligns features between the vision and language models. Following this, the model undergoes fine-tuning for specific applications such as visual chat and science question-answering, utilizing multimodal data to enhance its performance.

We can also fine-tune LLaVA model on specific purposes. {Expand few sentences}

Hallucination types –

In the context of object detection, hallucinations generally manifest in two failure modes: 1) missing objects, and 2) describing nonexistent objects or making incorrect statements about present objects. Empirically, the latter is less acceptable to humans. Object hallucinations can be further categorized into three different types: object category, object attribute, and object relation.


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. Three types of typical hallucination.

From the above image, we will extrapolate different object hallucinations:

- **Category**: MLLMs identify object categories that do not exist or incorrectly identify categories in the given image. For instance, describing "some benches and a fence" or "some clouds" in the text responses when these elements are not present in the image.
- **Attribute**: While the object categories identified by MLLMs are accurate, the descriptions of these objects' attributes (such as color, shape, material, content, counting, action, etc.) are incorrect. For example, describing "pink blossoms" when the actual color is different, demonstrating an attribute hallucination.
- **Relation**: All objects and their attributes are correctly identified, but the relationships among them (such as human-object interactions or relative positions) do not match the actual image content. An example is describing "...standing around her, watching..." when the objects are present but their relative positions or interactions are inaccurately depicted.

Hallucination Causes :

Hallucinations in multimodal large language models (MLLMs) arise from various sources throughout the capability acquisition process. The primary causes can be categorized into four aspects:

- **Data**: Inadequate or biased training data can lead to hallucinations, as the model may learn incorrect associations or fail to generalize properly to new data.

- **Model**: The architecture and design choices of the MLLM itself can contribute to hallucinations. This includes limitations in the model's ability to accurately represent and process multimodal inputs.
- **Training**: Issues during the training process, such as insufficient training duration, inappropriate loss functions, or suboptimal hyperparameters, can result in the model not learning the correct associations between modalities.
- **Inference**: During the inference phase, errors can occur due to limitations in the model's capacity to interpret and generate outputs based on the given inputs, leading to hallucinations.

Hallucination Benchmarks and Tools:

Unlike standard LVLM benchmarks that evaluate general capabilities, LVLM hallucination benchmarks specifically focus on non-hallucinatory generation or hallucination discrimination. Numerous benchmarks have been proposed along with different evaluation methods. The following chart synthesizes representative benchmarks categorized by their evaluation approach: Discriminative (Dis) or Generative (Gen) [16].

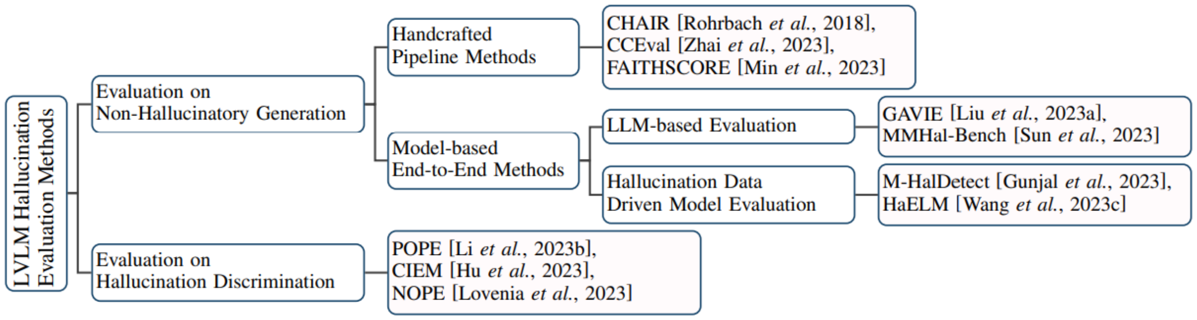

LVLM Hallucination Evaluation Methods
Evaluation on Non-Hallucinatory Generation
Handcrafted Pipeline Methods
CHAIR [Rohrbach et al., 2018], CCEval [Zhai et al., 2023], FAITHSCORE [Min et al., 2023]
Model-based End-to-End Methods
LLM-based Evaluation
GAVIE [Liu et al., 2023a], MMHal-Bench [Sun et al., 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 3: Hallucination Evaluation Methods

Discriminative Benchmarks: Examples include POPE [18], NOPE [19], and CIEM [20]. These benchmarks focus exclusively on object hallucinations and use accuracy as the evaluation metric. This metric is determined by querying the presence of objects in images and comparing model responses with ground-truth answers [16].

Generative Benchmarks: These benchmarks cover a broader range of hallucinations, including those related to attributes and relations [21, 22, 23, 24]. Notably, AMBER [25] is a comprehensive benchmark that incorporates both generative and discriminative tasks, providing a more extensive evaluation of hallucination phenomena.

In this study, we will be using 3 different hallucination tools. One is a derived version of MMHal-bench evaluation which is Model-based, End-to-End Method.

The second tool is Aloha which is a tool for detecting hallucinations in Captioning Models.

The last tool is utilizing GrondingDINO to detect the objects we got from our models and to check whether the objects are present in the given images or if the models are hallucinating.

ALOHA :

ALOHa is a modern open-vocabulary metric designed to leverage 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 from captions and object detections, and utilizes Hungarian matching to generate a final hallucination score [27].

ALOHa enhances the reliability and localization capabilities of CHAIR by applying in-context learning of LLMs and using semantically rich text embeddings for object parsing and matching. For each image caption, ALOHa provides two measures:

- **ALOHao**: A numeric score for each object, indicating the degree to which the object is a hallucination.
- **ALOHa**: An aggregated score evaluating the overall extent of hallucinations within the entire caption.

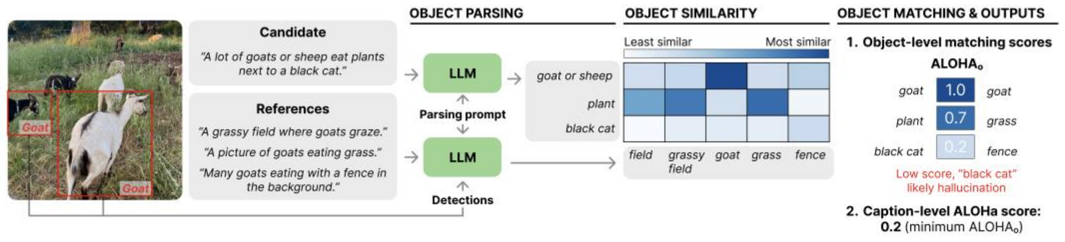

Goat
Goat
Candidate
"A lot of goats or sheep eat plants next to a black cat."
References
"A grassy field where goats graze."
"A picture of goats eating grass."
"Many goats eating with a fence in the background."
OBJECT PARSING
LLM
Parsing prompt
LLM
Detections
goat or sheep
plant
black cat
OBJECT SIMILARITY
Least similar
Most similar
field
grassy field
goat
grass
fence
OBJECT MATCHING & OUTPUTS
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 4: Overview of Aloha

GroundingDINO –

Grounding DINO is an advanced zero-shot object detection tool which integrates the robust DINO architecture with grounded pre-training. Grounding DINO can help in identifying arbitrary objects based on human inputs, such as references or naming or any type of special attributes.

Grounding DINO employs a self-supervised learning algorithm, combining DINO (DETR with Improved deNoising anchOr boxes) with grounded (GLIP) pre-training. DINO, a transformer-based detection method, excels in state-of-the-art object detection and end-to-end optimization, removing the need for handcrafted modules like Non-Maximum Suppression (NMS). GLIP, on the other hand, specializes in phrase grounding, which links phrases or words from a text to their corresponding visual elements in an image or video, effectively connecting textual descriptions to visual representations [28].

Zero-shot object detection models, such as Grounding DINO, which aims to generalize to unseen objects with minimal data, overcoming traditional limitations. 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 corresponding descriptions.
3. **Feature Enhancer**: Fuses image and text features, facilitating cross-modality information exchange.
4. **Language-Guided Query Selection**: Initializes queries using language information.
5. **Cross-Modality Decoder**: This predicts bounding boxes based on the fused features and queries.

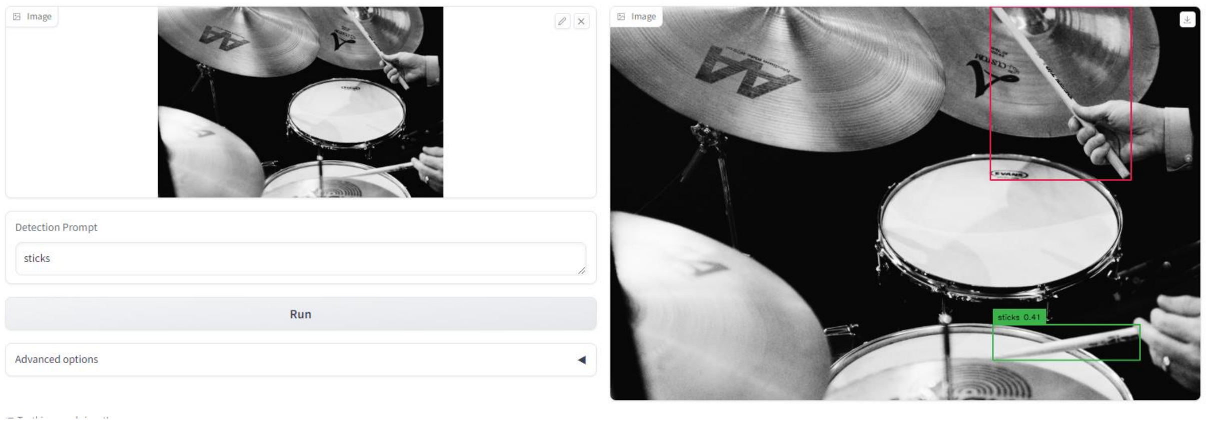


Fig 5: GroundingDINO demo

We will manually take all the objects that the model generates and manually test it out on Huggingface. If the object is present in the image, we will get the object detection with probability score and it not we will not get anything.

MMHAL-Bench :

We have formatted and used the Evaluation part of MMHal-Bench. We have changed the format of the template to evaluate hallucinations in images. We need a truth reference, a response from the LLM, and an instruction for the LLM to evaluate any kind of object hallucination present in the given responses respective to the image.

Objective:

This study aims to fine-tune an open-source Large Vision-Language Model (LVLM) - LLaVA 1.5 on a publicly available dataset that primarily focuses on object detection. Following the fine-tuning process, we will present the results to evaluate the performance of our model against 250 new images. To assess its zero-shot performance on new images, we will compare the results using three custom hallucination detection tools: ALOHa, MMAL-Bench, and GroundingDINO.

Methodology:

The study follows a structured approach, divided into three main steps:

1. **Fine-tuning LLaVA 1.5 Model**:
    - The LLaVA 1.5 model will be fine-tuned using a publicly available dataset that focuses on object detection.
    - Hyperparameter tuning and selecting the right parameters.
2. **Customizing and Using Three Hallucination Detection Tools**:

- Three hallucination detection tools—ALOHa, MMHAL-Bench, and GroundingDINO will be employed.
- MMHAL-Bench will be customized to evaluate hallucinations by comparing the various objects identified in images.

3. **Analyzing the Results**:
   - The performance of the fine-tuned model with 250 images
   - Statistical analysis will be conducted to assess the differences in performance, particularly focusing on zero-shot new images using the hallucination detection tools.

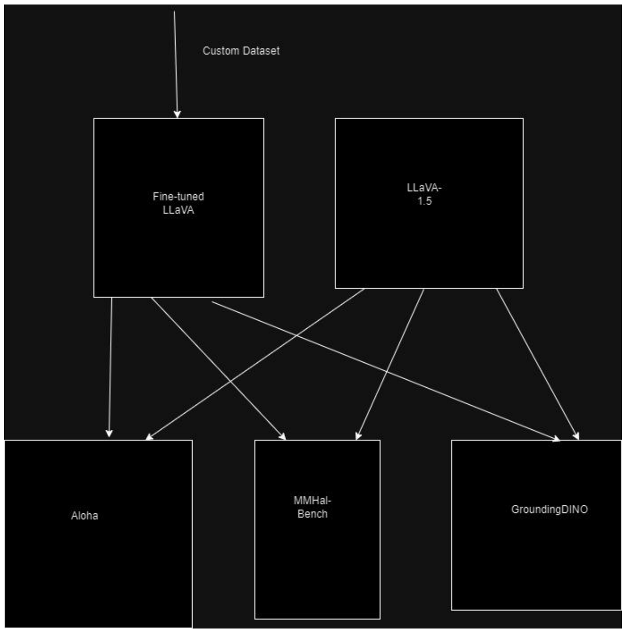


Fig 6: Overview of the Study

1. Fine-tuning LLaVA model –

Objective / Purpose- The primary objective of fine-tuning the LLaVA 1.5 model is to accurately detect or describe only objects present in visual images. 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 a 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. The dataset is divided into training and testing sets:

- **OK-VQA_train**: Contains 9.1k rows.
- **OK-VQA_test**: Contains 5.05k rows.

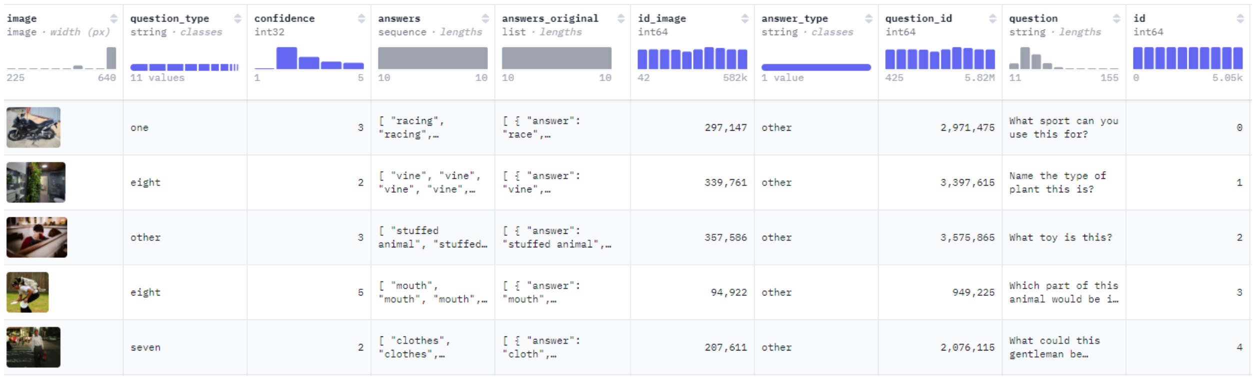


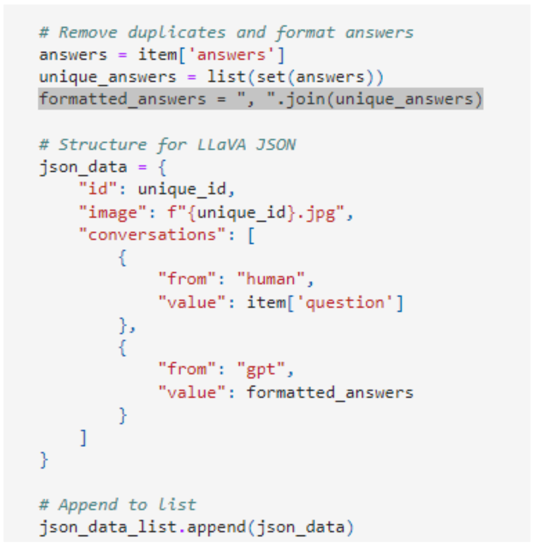
image	question_type	confidence	answers	answers_original	id_image	answer_type	question_id	question	id
image · *width (px)*	string · *classes*	int32	sequence · *lengths*	list · *lengths*	int64	string · *classes*	int64	string · *lengths*	int64
225 640	11 values	1 5	10 10	10 10	42 582k	1 value	425 5.82M	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 train dataset

Columns of the Dataset

- **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:
    - answers: The provided answers.
    - raw_answer: The raw form of the answer.
    - answer_confidence: The confidence level of the answer.
    - 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.

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



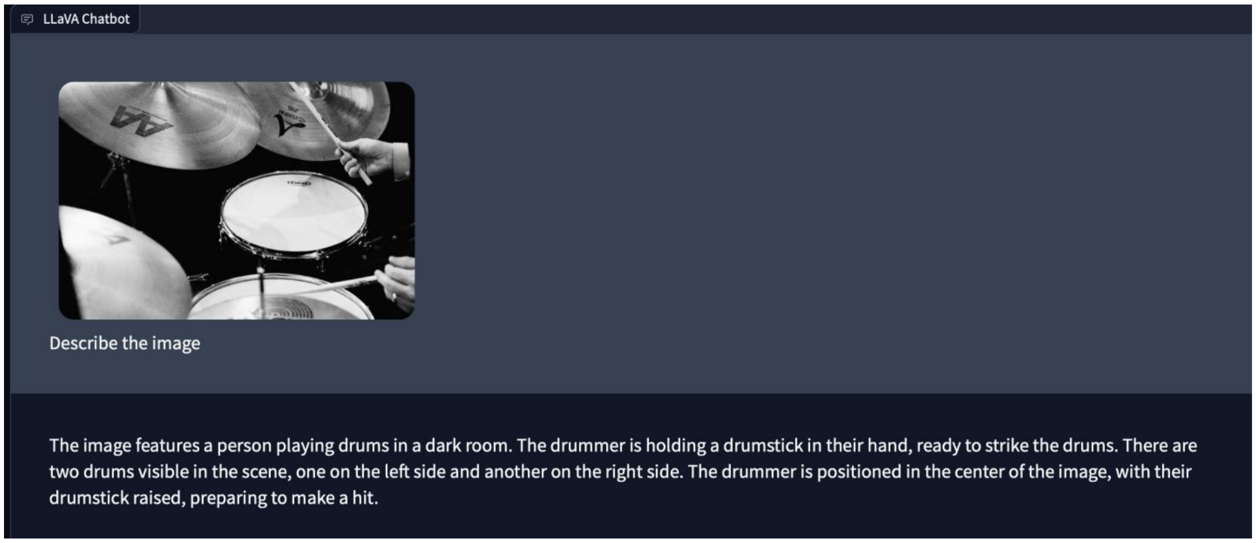
```python
# Remove duplicates and format answers
answers = item['answers']
unique_answers = list(set(answers))
formatted_answers = ", ".join(unique_answers)

# 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 8: 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.

Results:

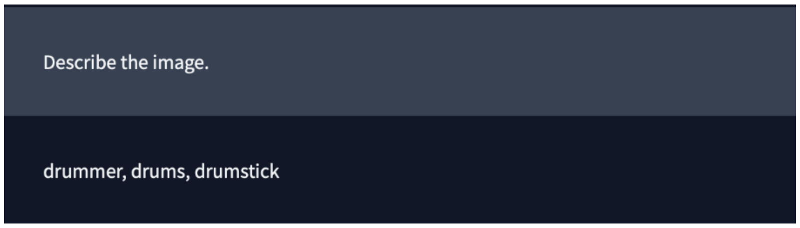
After fine-tuning and running our model on Gradio, we compared the output of the fine-tuned model with the original model.



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 9: Original LLaVA 1.5 model demo


Describe the image.

drummer, drums, drumstick

Fig 10: Fine-tuned LLaVA 1.5 model demo

Our model is able to detect only objects when prompted to describe an image, whereas the original model tends to follow general instructions. Additionally, our model is more specific in listing various types of objects and combinations of objects compared to the original model.

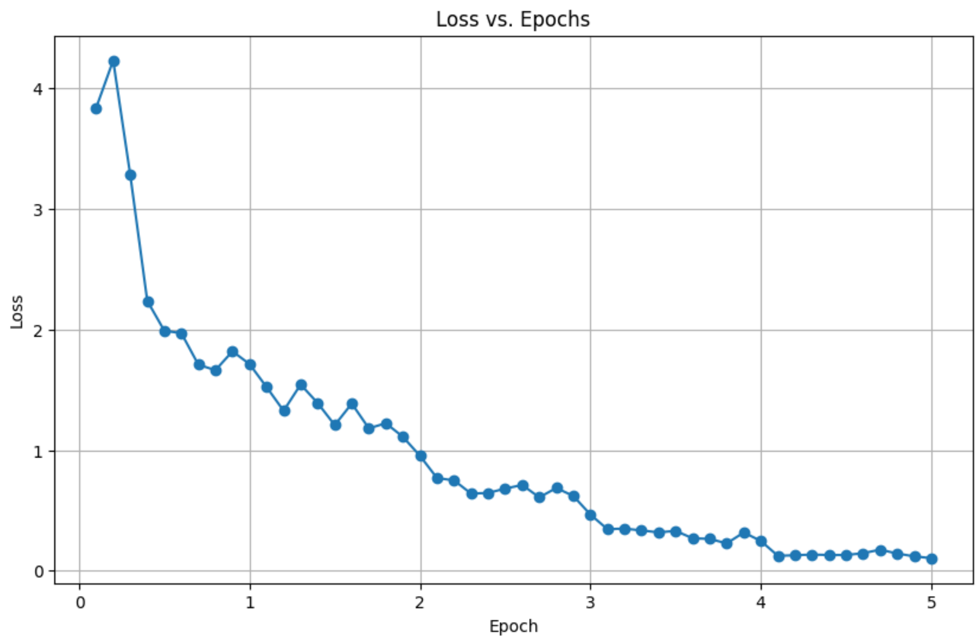
Performance :

```
results = [
    {'loss': 3.8382, 'learning_rate': 0.0001, 'epoch': 0.1},
    {'loss': 4.2314, 'learning_rate': 0.0002, 'epoch': 0.2},
    {'loss': 3.2841, 'learning_rate': 0.00019978589232386035, 'epoch': 0.3},
    {'loss': 2.2376, 'learning_rate': 0.00019914448613738106, 'epoch': 0.4},
    {'loss': 1.99, 'learning_rate': 0.00019807852804032305, 'epoch': 0.5},
    {'loss': 1.9696, 'learning_rate': 0.00019659258262890683, 'epoch': 0.6},
    {'loss': 1.7079, 'learning_rate': 0.0001946930129495106, 'epoch': 0.7},
    {'loss': 1.6628, 'learning_rate': 0.00019238795325112867, 'epoch': 0.8},
    {'loss': 1.8193, 'learning_rate': 0.00018968727415326884, 'epoch': 0.9},
    {'loss': 1.7132, 'learning_rate': 0.00018660254037844388, 'epoch': 1.0},
    {'loss': 1.5226, 'learning_rate': 0.00018314696123025454, 'epoch': 1.1},
    {'loss': 1.3258, 'learning_rate': 0.00017933533402912354, 'epoch': 1.2},
    {'loss': 1.544, 'learning_rate': 0.00017518398074789775, 'epoch': 1.3},
    {'loss': 1.3895, 'learning_rate': 0.00017071067811865476, 'epoch': 1.4},
    {'loss': 1.2073, 'learning_rate': 0.00016593458151000688, 'epoch': 1.5},
    {'loss': 1.3857, 'learning_rate': 0.00016087614290087208, 'epoch': 1.6},
    {'loss': 1.1802, 'learning_rate': 0.00015555702330196023, 'epoch': 1.7},
    {'loss': 1.2227, 'learning_rate': 0.00015000000000000001, 'epoch': 1.8},
    {'loss': 1.1097, 'learning_rate': 0.00014422886902190014, 'epoch': 1.9},
    {'loss': 0.9525, 'learning_rate': 0.000138268343236509, 'epoch': 2.0},
    {'loss': 0.768, 'learning_rate': 0.00013214394653031616, 'epoch': 2.1},
    {'loss': 0.749, 'learning_rate': 0.00012588190451025207, 'epoch': 2.2},
    {'loss': 0.6389, 'learning_rate': 0.00011950903220161285, 'epoch': 2.3},
    {'loss': 0.6445, 'learning_rate': 0.00011305261922200519, 'epoch': 2.4},
    {'loss': 0.6803, 'learning_rate': 0.00010654031292301432, 'epoch': 2.5},
    {'loss': 0.7108, 'learning_rate': 0.0001, 'epoch': 2.6},
    {'loss': 0.6077, 'learning_rate': 9.345968707698569e-05, 'epoch': 2.7},
    {'loss': 0.6856, 'learning_rate': 8.694738077799488e-05, 'epoch': 2.8},
    {'loss': 0.6194, 'learning_rate': 8.049096779838719e-05, 'epoch': 2.9},
    {'loss': 0.4654, 'learning_rate': 7.411809548974792e-05, 'epoch': 3.0},
    {'loss': 0.3437, 'learning_rate': 6.785605346968386e-05, 'epoch': 3.1},
    {'loss': 0.349, 'learning_rate': 6.173165676349103e-05, 'epoch': 3.2},
    {'loss': 0.3334, 'learning_rate': 5.577113097809989e-05, 'epoch': 3.3},
    {'loss': 0.3175, 'learning_rate': 5.000000000000002e-05, 'epoch': 3.4},
    {'loss': 0.33, 'learning_rate': 4.444297669803981e-05, 'epoch': 3.5},
    {'loss': 0.2682, 'learning_rate': 3.9123857099127936e-05, 'epoch': 3.6},
    {'loss': 0.2633, 'learning_rate': 3.406541848999312e-05, 'epoch': 3.7},
    {'loss': 0.226, 'learning_rate': 2.9289321881345254e-05, 'epoch': 3.8},
    {'loss': 0.3165, 'learning_rate': 2.4816019252102273e-05, 'epoch': 3.9},
    {'loss': 0.2463, 'learning_rate': 2.0664665970876496e-05, 'epoch': 4.0},
    {'loss': 0.1233, 'learning_rate': 1.6853038769745467e-05, 'epoch': 4.1},
    {'loss': 0.1276, 'learning_rate': 1.339745962155613e-05, 'epoch': 4.2},
    {'loss': 0.1334, 'learning_rate': 1.0312725846731175e-05, 'epoch': 4.3},
    {'loss': 0.1296, 'learning_rate': 7.612046748871327e-06, 'epoch': 4.4},
    {'loss': 0.1297, 'learning_rate': 5.306987050489442e-06, 'epoch': 4.5},
    {'loss': 0.1447, 'learning_rate': 3.40741737109318e-06, 'epoch': 4.6},
    {'loss': 0.174, 'learning_rate': 1.921471959676957e-06, 'epoch': 4.7},
    {'loss': 0.1419, 'learning_rate': 8.555138626189618e-07, 'epoch': 4.8},
    {'loss': 0.1186, 'learning_rate': 2.141076761396521e-07, 'epoch': 4.9},
    {'loss': 0.1035, 'learning_rate': 0.0, 'epoch': 5.0},
]
```

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 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 approaching 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.



Loss vs Epoch

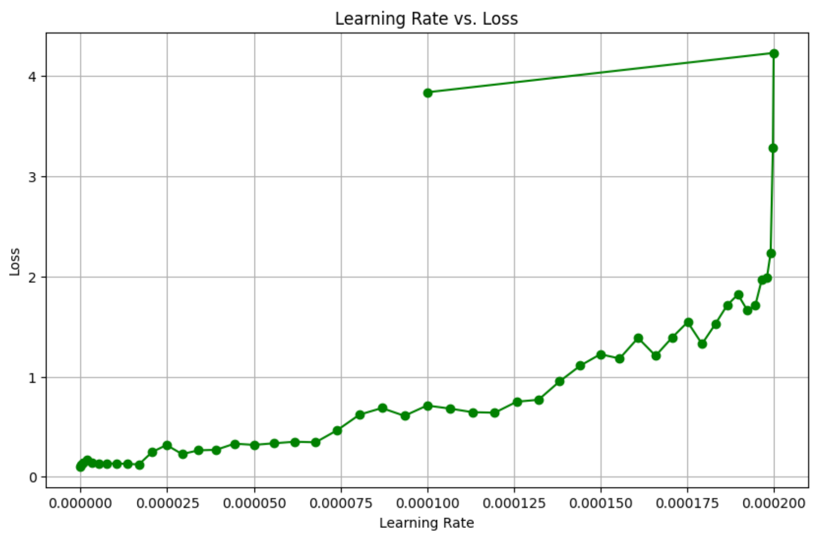


Fig: Learning rate vs Loss

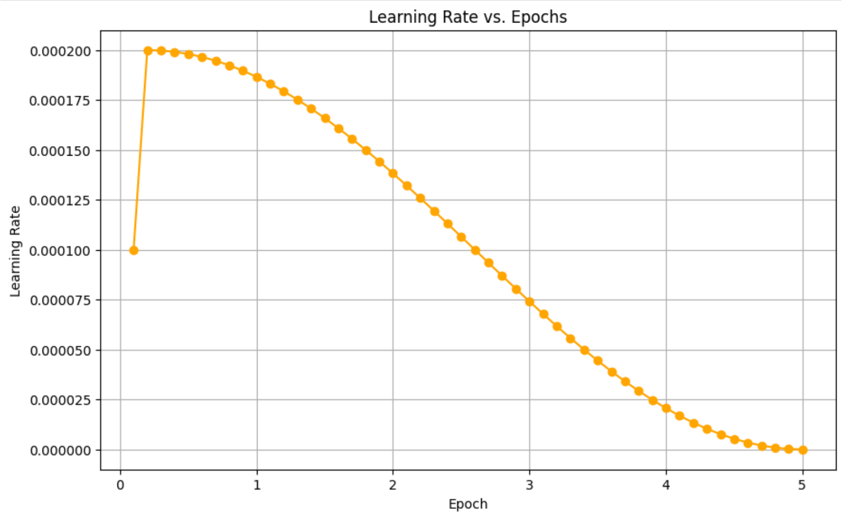


Fig: Learning rate vs Epochs

Confusion Matrix:

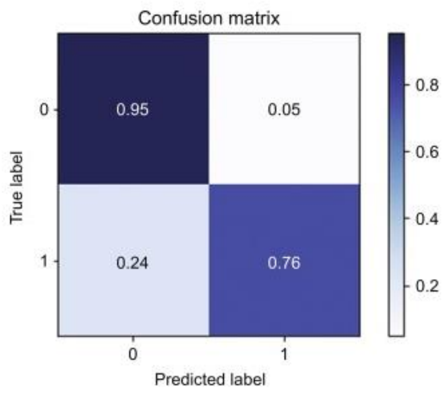


Fig: Confusion Matrix

The provided confusion matrix 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:

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

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

Recall=TPTP+FN=0.76

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

F1 Score=≈0.839

4. **Accuracy**:

Accuracy=0.855

Results:

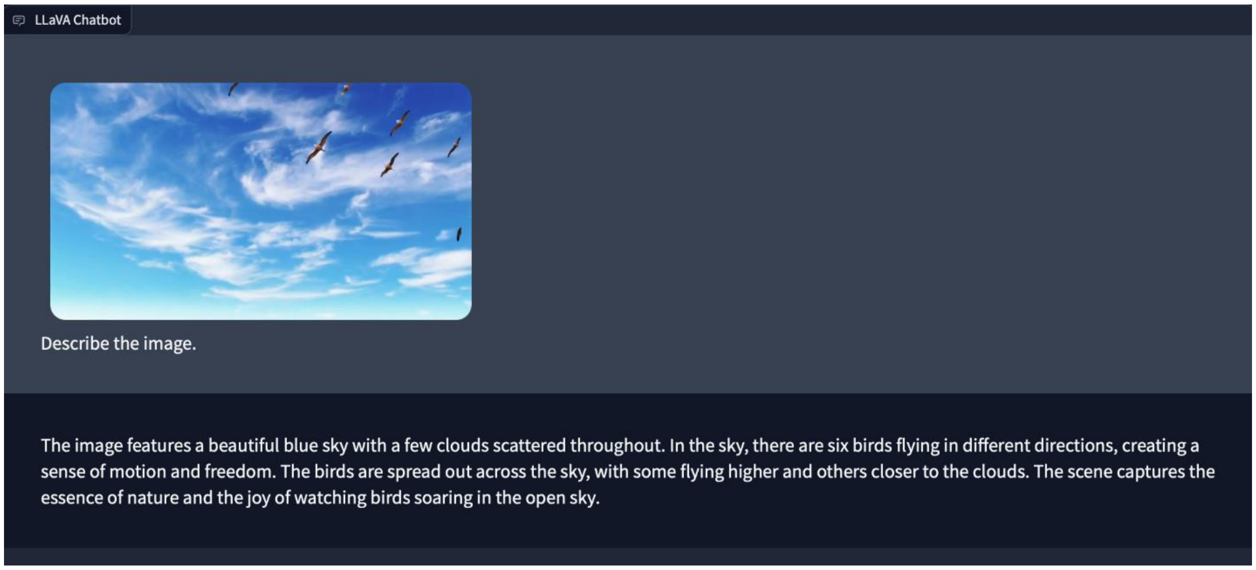
The fine-tuned model was evaluated using Gradio, and its performance was compared to the original LLaVA 1.5 model. Key observations include:

- The fine-tuned model demonstrated improved specificity in listing various types of objects and their combinations within an image.

- The original model tended to follow general instructions, whereas the fine-tuned model focused solely on detecting and describing objects.

The examples of our fine-tuned model vs the original model is given below –

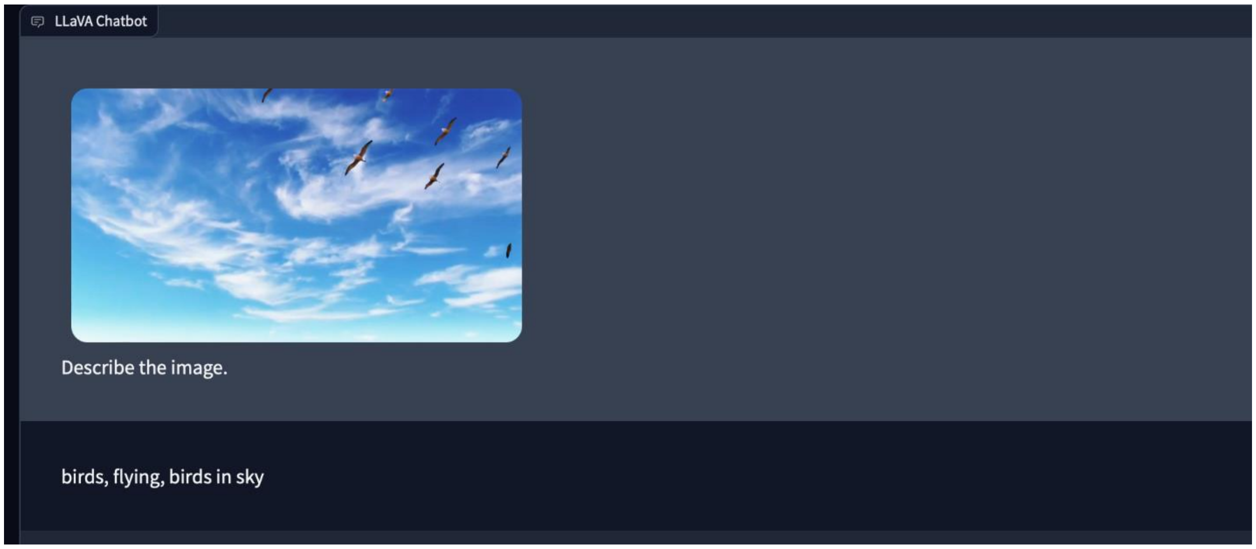
Original model –



Describe the image.

The image features a beautiful blue sky with a few clouds scattered throughout. In the sky, there are six birds flying in different directions, creating a sense of motion and freedom. The birds are spread out across the sky, with some flying higher and others closer to the clouds. The scene captures the essence of nature and the joy of watching birds soaring in the open sky.

Fig 11: Example of LLaVA 1.5 model



Describe the image.

birds, flying, birds in sky

Fig 12: Example of Fine-tuned LLaVA 1.5 model

Conclusion

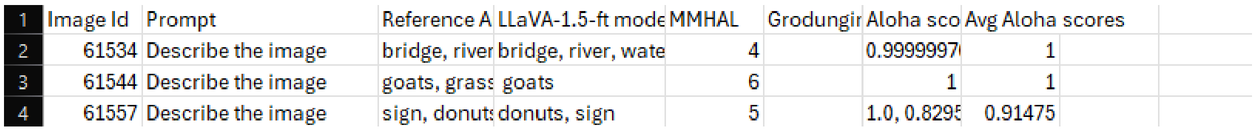
The model has an accuracy of 85.5%, indicating that it correctly predicts the classes in most cases. The precision is high at 93.8%, which means that when the model predicts a positive instance, it is likely to be correct. However, the recall is lower at 76%, indicating that the model misses some positive instances. The F1 score, which balances precision and recall, is 83.9%. Overall, the model performs well and can detect the objects given in a visual image.

For fine-tuning, training for 5 epochs has been found to be optimal, resulting in a log loss of 0.1035. This indicates that the model has effectively learned from the data and achieved a high level of accuracy and precision. After more than 5 epochs, the loss is stabilized so there will not be major change in the performance of the model.

2. Hallucination Detection Tools –
    - Aloha
    - {MMHal-Bench}
    - GroundingDINO

Testing our model on the above-mentioned tools:

We prepare a small dataset of 250 images with questions and reference truths. We manually label the objects in our image dataset. Then we send the questions or prompts along with the images onto the LLaVA 1.5- finetuned model. Once we collect the answers, we will use the mentioned hallucination detection tools to determine how well our fine-tuned model is performing on object detection.



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 13: Overview of the sheet

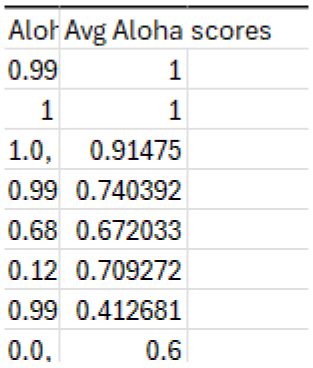
Aloha –

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, we 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 matching 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.

Results:



Alol	Avg Aloha scores	
0.99	1	
1	1	
1.0,	0.91475	
0.99	0.740392	
0.68	0.672033	
0.12	0.709272	
0.99	0.412681	
0.0,	0.6	

Fig 14: LLaVA-1.5 model results

The average ALOHa score across 250 different examples for the the fine-tuned model achieves a score of 0.886077. This indicates that our fine-tuned model is more effective at detecting objects and minimizing hallucinations.

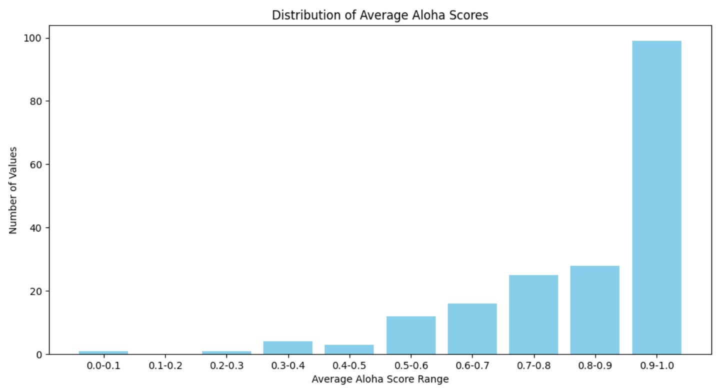
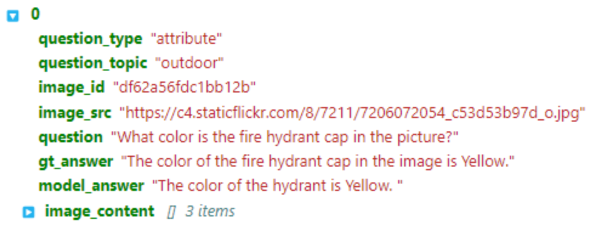


Fig 15: Distribution of Avg Aloha Values

The majority of the scores fall within the 0.9 to 1.0 range, indicating that the model performs exceptionally well for most instances. There are smaller, but notable, counts in the ranges of 0.7 to 0.8 and 0.8 to 0.9, further demonstrating strong performance. Lower score ranges have significantly fewer values, suggesting that the model very few performs poorly. Overall, this distribution reflects a high-performing model with a tendency towards achieving near-perfect object detection by the fine-tuned model.

MMHal-Bench –

The Data is given in the following format.



```
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 16: Format of MMHal-bench

After modifying the template and some evaluation methods of MMHal-Bench, we collected hallucination scores across 10 different images for both our fine-tuned model and the original model. The results show a slight decrease in the hallucination rate for our fine-tuned model compared to the original model. Our data has been formatted accordingly for LLM analysis.

0 – Not informative, with hallucinations
…
6 – Very informative and no hallucination

Results

The average hallucination score for our fine-tuned model achieves a slightly higher score of 4.428. This indicates that our model is marginally better at detecting objects accurately, thereby reducing hallucinations. 4.428 falls in between 4 (somewhat informative to the references, no hallucination) and 5(very informative, no hallucination).

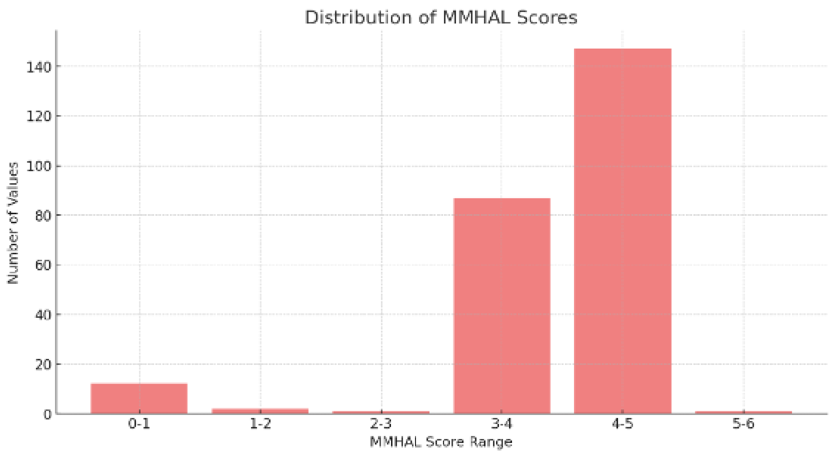


Fig 17: Distribution of MMHAL Scores

GroundingDINO:

We utilized the GroundingDINO object detection tool to identify the given object labels from both our fine-tuned model and the original model responses. The responses from the original model were converted into a list of objects using OpenAI ChatGPT 3.5. We manually detected all the objects for both models and conducted an analysis to determine how well our model predicted the objects.

Results

We were able to find only a few object labels that weren't present in their respective visual images. 10 object labels were wrongly identified by the fine-tuned model which aren't present in the visual image. Only 4% of the responses had one or few object-label hallucinations.

3. Conclusion:

The evaluation of our fine-tuned LLaVA 1.5 model using three distinct tools—ALOHa, MMHal-Bench, and GroundingDINO—demonstrates its superior performance in object detection and minimizing hallucinations.

- **ALOHa**: The fine-tuned model achieved an average score of 0.886, indicating better object detection and reduced hallucination rates.
- **MMHal-Bench**: Our model scored an average of 4.428 shows an improvement in object detection capabilities.
- **GroundingDINO**: Analysis showed that our fine-tuned model detected 10 hallucinated object label responses across 250 examples, highlighting its enhanced reliability in object identification.

Overall, the fine-tuned model's consistent performance across these tools confirms its effectiveness in reducing hallucinations and improving object detection accuracy. This enhancement is crucial for applications requiring precise visual and language understanding, as it ensures reliable and accurate outputs. By fine-tuning a general purposed model to specific tasks can be a major improvement and can have better results.

Future Scope:

To further enhance the performance of multimodal large language models (MLLMs) and effectively address the issue of hallucinations, the development of more robust frameworks and diverse evaluation methods is essential. 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.

References –

1. Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 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
2. Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. ArXiv preprint abs/2305.08322 (2023). https://arxiv.org/abs/2305.08322
3. Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. ArXiv preprint abs/2301.13848 (2023). https: //arxiv.org/abs/2301.13848
4. Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. ArXiv preprint abs/2304.04675 (2023). https://arxiv.org/abs/2304.04675
5. Fei Yu, Hongbo Zhang, and Benyou Wang. 2023. Nature language reasoning, a survey. ArXiv preprint abs/2303.14725 (2023). https://arxiv.org/abs/2303.14725
6. Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023).
7. Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. arXiv preprint arXiv:2304.14178 (2023).
8. Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arXiv:2305.06500
9. Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023).
10. Analyzing and mitigating object hallucination in large vision-language models. In NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following, 2023
11. Haotian Liu, Chunyuan Li, Qingyang Wu, et al. Visual instruction tuning. In NeurIPS, 2023
12. Haotian Liu, Chunyuan Li, Yuheng Li, et al. Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744, 2023.
13. Lei Huang, Weijiang Yu, Weitao Ma, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint arXiv:2311.05232, 2023.
14. https://medium.com/@cout.shubham/exploring-multimodal-large-language-models-a-step-forward-in-ai-626918c6a3ec
15. Hallucination of Multimodal Large Language Models: A Survey
16. A Survey on Hallucination in Large Vision Language Models
17. Anna Rohrbach, Makarand Tapaswi, Atousa Torabi, Tegan Maharaj, Marcus Rohrbach, Sanja Fidler Christopher Pal, and Bernt Schiele. 2017. The Joint Video and Language

Understanding Workshop: MovieQA and The Large Scale Movie Description Challenge (LSMDC).

18. Yifan Li, Yifan Du, Kun Zhou, et al. Evaluating object hallucination in large vision-language models. In EMNLP, 2023.
19. Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, et al. Negative object presence evaluation (nope) to measure object hallucination in vision-language models. arXiv preprint arXiv:2310.05338, 2023.
20. Lei Huang, Weijiang Yu, Weitao Ma, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint arXiv:2311.05232, 2023.
21. Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. arXiv preprint arXiv:2308.06394, 2023.
22. Fuxiao Liu, Kevin Lin, Linjie Li, et al. Mitigating hallucination in large multi-modal models via robust instruction tuning. arXiv preprint arXiv:2306.14565, 2023.
23. Ziwei Ji, Nayeon Lee, Rita Frieske, et al. Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), 2023.
24. Junyang Wang, Yiyang Zhou, Guohai Xu, et al. Evaluation and analysis of hallucination in large vision-language models. arXiv preprint arXiv:2308.15126, 2023.
25. Junyang Wang, Yuhang Wang, Guohai Xu, et al. An llm-free multi-dimensional benchmark for mllms hallucination evaluation. arXiv preprint arXiv:2311.07397, 2023.
26. https://medium.com/ai-insights-cobet/introducing-llava-the-fusion-of-visual-and-linguistic-intelligence-in-ai-with-code-ceb97c684d5a
27. ALOHa: A New Measure for Hallucination in Captioning Models - https://arxiv.org/pdf/2404.02904
28. https://medium.com/@tauseefahmad12/zero-shot-object-detection-with-grounding-dino-aefe99b5a67d
29. Attention is all you need - https://arxiv.org/abs/1706.03762