Team CRIMSON NOIR

Jithendra Katta - G34424060
Nikhil Reddy Kodumuru - G27606239

Neural Networks and Deep Learning

Design Report for Project: **InterACT**

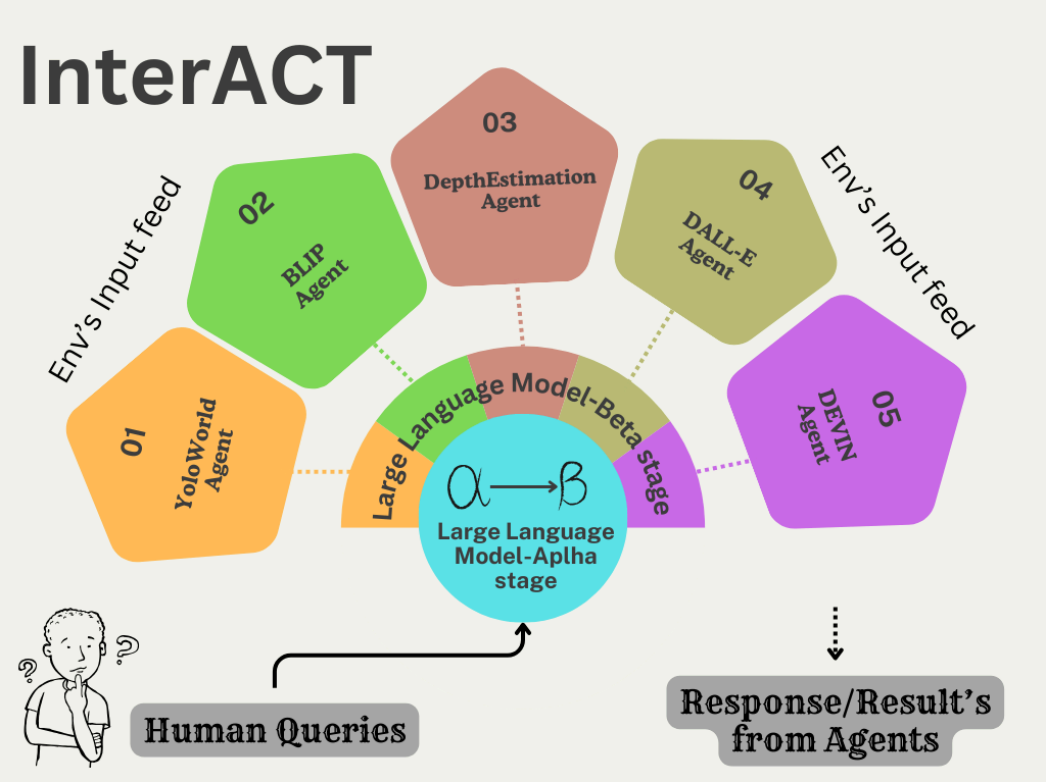

Fig 1: Project InterACT Design

THE BIG PICTURE:

The above figure displays a blueprint of the project InterACT. The main Idea of this project is to build a Multi-Modal AI system where a Human can Interact with this system and can get intended responses/results from the local Environment.

The Project InterACT is mainly dependent on the large language model. which has two stages: Alpha stage and Beta stage. Based on the query given by the user, the alpha stage functionality is to direct the processed query to the beta stage of a corresponding Agent. And at the beta stage the LLM is responsible for reading the processed query and tries to generate texts/responses which will be sent to the corresponding Agent. With the Environment Input feed on, the corresponding agent will execute the response with the Environment input feed and generate results(relevant responses) which are interpreted to users.

Each and every Agent used in this project will be fine-tuned or retrained with Environments feed or Custom datasets.

Note: As mentioned in the Project Proposal, due to time and computation constraints, we are planning to **implement only the phase-1 of the project with a single Agent and an LLM**: an Object detection model and a Fine-tuned Large Language Model.

Project Architecture:

The Project architecture is based on two sections: An Object Detection Model and An LLM. When a Query/Expression given by a user will be processed by an LLM to extract specific classes(objects) which will be used by the Object Detection Model to detect them in the Environment Feed(Snap of the Environment).

Input/output:

Input: 1. An Image of the local environment.
2.A query or any text input from the user.

Output: the same input Image with bounding boxes to corresponding objects.

Example:

Input: 1. Image of a living room with a person watching television.
2.query: I want change the tv channel

Output: The Image with some bounding box around TV remote control and Television if they exist in the image frame.

Design of Experiment:

1. **Regression/Classification Task:**
   The task we intend to do is object detection on specific objects in videos when a user asks any query to the system. Here we are performing classification task(Object detection)

2. **Uncertain Quantity of Interest**
   The presence, set of classes(response generated by LLM) and position of the object in the frame will be our uncertain quantity of Interest.

3. **Utility Function**
   The combination of Object detection accuracy and localization precision can be considered as utility functions. Here, we can use the weighted sum of following metrics:
   - Object Detection Accuracy: The accuracy of correctly classifying the presence and class of objects in the frame, based on the user's query.
   - Large Language Model Response: The precision of the responses for the user's query/expression, measured by the ROGUE (Recall oriented understudy for gisting evaluation).

4. **Proposed Model Architectures**
   We use a Multi Model Architecture that combines a large language model and a single agent(Object detection). Based on the query given by the user the query will be processed by an LLM which is responsible for reading the query and generates texts/responses which will be sent to the Object Detection model. With the Environment Input feed on, the Object Detection model will execute the response with the Environment input feed and generate results(bounding boxes around specific objects) which are interpreted to users.

- Large Language Model (LLM): We are planning to use transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) or FLAN-T5 (Fine-tuned Language Net - Text-To-Text Transfer Transformer), Llama-2-chat models for language understanding and query interpretation.
- Agent (Object detection): We are planning to use CNN based models such as YOLOv8 or YoloWorld which are SOTA for Object detection.

Both the models will be attached, where output of LLM will be processed as input to the Object Detection model and the Image from Environment also will be an input factor which will be processed using computer vision.

5. **Evaluating Model Architectures**
   We are planning to use the following strategies for evaluating our model architectures:

   - Hyperparameter Tuning: We will perform systematic hyperparameter tuning for both the LLM and object detection models, including learning rates, batch sizes, optimizers, and regularization techniques, to find the optimal configurations.
   - Cross-Validation: For each and every model we assess the generalization performance individually and will try to mitigate any kinds of overfitting or underfitting.

6. **Measuring Model Success**
   The success of the every individual model will be measured separately using the following metrics:
   - Object Detection Accuracy: Precision, recall, confidence and F1-score for object detection, considering the user's query.
   - Combined Metric: A weighted combination of object detection accuracy and localization precision, as defined in the utility function.

7. **Comparison to Prior Art**
   While there has been prior work in object detection and large language models separately combinedly, the integration of these two domains for interactive object detection in videos is a relatively new area. We will compare the performance of every individual model to any existing benchmarks or related work in this specific domain.

Data

1. **Data Sources**
   As we are planning to Use **YoloWorld**: YOLO-World revitalizes the YOLOv8 framework with open-vocabulary detection capabilities, employing vision-language modeling and pre-training on expansive datasets to excel at identifying a broad array of objects in zero-shot scenarios with unmatched efficiency, where YOLOv8 trained on COCO Image dataset. Here, we use some custom or popular datasets to finetune our SOTA models. By considering these, for this project, We are planning to utilize the following data sources:
   Flickr30k: The Flickr30k dataset is a popular benchmark for sentence-based picture portrayal. The dataset comprises 31,783 images that capture people engaged in everyday activities and events. Each image has a descriptive caption. Flickr30k is used for understanding the visual media (image) that corresponds to a linguistic expression (description of the image). This dataset is commonly used as a standard benchmark for sentence-based image descriptions.

   Custom built Corpus dataset: A dataset of prompts and responses, which can be used for Fine-tuning the LLM on interactive environments related to visual scenes and objects.

   Environment Input feed: We give a snap of the environment feed which will be processed using CV.

2. **Data Format and Tagging**
   - The Flickr30k dataset provides annotations in the form of JSON file, with bounding box positive and negative tokens and class labels for each object instance and an images folder where all the 30k images exist. The Custom built Corpus dataset consists of text files with movie transcripts and metadata.

3. **Data Suitability**

The Flickr30k dataset is widely used for object detection tasks and provides a diverse range of object classes and visual scenarios, making it suitable for training and fine-tuning the object detection model.
The Custom built Corpus data contains natural language related to visual scenes and objects, which can be useful for fine tuning the LLM to understand queries in the context of object detection.

Progress of the Project

LLM:

- **Fine Tuning Llama-2-7B-chat_hf LLM Model:**
  We have finetuned **Llama-2-7B-chat_hf** model on **Custom built Corpus dataset** and pushed both dataset and the fine tuned model into the Hugging Face. As the Llama-2-7B-chat_hf model is very large we got our fine tuned model a large one and unable to load in the Inference API to try it and also taking a lot of computational power.
  Here I am attaching the dataset and Fine Tuned Model repo's:
  Custom Dataset for Llama model: https://huggingface.co/datasets/Jithendra-k/InterACT
  Fine Tuned Llama model: https://huggingface.co/Jithendra-k/interACT_LLM

- **Fine Tuning Flan-T5 LLM Model:**
  Due to the computation and size issue we have tried fine tuning the **Flan-T5_small** model which requires very less computation when compared to the Llama model. We have **formatted the dataset** as required by the Flan-T5 input format and pushed it into Hugging Face. After Fine Tuning we were able to load it in the Inference API but the results were not that good. So, We are proceeding with a fine tuned Llama-2-7B-chat model.
  Here I am attaching the dataset and Fine Tuned Model repo's:
  Custom Dataset Flan-T5 model:
  https://huggingface.co/datasets/Jithendra-k/Flan-T5_interACT
  Fine Tuned Flan-T5 model: https://huggingface.co/Jithendra-k/Flan_T5_InterACT

Object Detection:

- **Fine Tuning YOLO-World:**
  YOLO-World, an innovative approach that enhances YOLO with open-vocabulary detection capabilities through vision-language modeling and pre-training on large-scale datasets. It is trained on the COCO dataset.
  We have some issues with Fine-tuning YOLO-World, as this is a very new model which was released just in February month 2024. Fine-tuning YOLO-World requires **mmyolo** Library which is a third party library which has version conflicts with other required Libraries.

So we are trying to Implement Training YOLOWorld from scratch using the Flickr 30K dataset. But it requires very High computation and many hours of time. So, we are trying to reduce the dataset size and train the YOLO-World from scratch.

After all, If the issue with **mmyolo** resolves we will try to fine-tune it rather than training it from scratch.

References

1. **YOLO-World:** https://arxiv.org/html/2401.17270v3
2. **Llama-2-7B-chat**: https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
3. **Flan-T5_small:** https://huggingface.co/google/flan-t5-small
4. **Flickr-30K**: https://paperswithcode.com/dataset/flickr30k