LLaVA-Recipe: Visual Instruction Tuning Enhanced Food Recipe VQA

Daoting Chen, Wensong Hu, Tianxu Jiang, Yichen Lu
{daotingc, umhws, tianxuj, nechy}@umich.edu

Abstract—In this study, we present an approach to augmenting the capabilities of Large Language and Vision Assistant (LLaVA) for generating detailed cooking recipes from visual inputs. Leveraging the extensive Recipe1M+ dataset, our methodology involves a two-stage fine-tuning process that integrates visual and textual data, enhancing LLaVA's ability to produce coherent and applicable cooking instructions from image-text pairs. With multi-turn dialogue generation and Ingredients and Instruction Mentioned(IIM), our experiments find that visual instruction tuning on Recipe1M+ enhanced LLaVA's performance in recipe generation. We also explore the effect of IIM and number of training epochs on LLaVA through experiments and draw the conclusion that IIM boosts performance and increased training epochs benefit IIM models.

I. Introduction

Models with multi-modality that including the Large Language Models (LLMs) have increasingly penetrated various fields, since it is capable of generating really reasonable outputs in a lot of scenarios. However, its universal ability sometimes still not fulfilling the direct applications that requires detailed, accurate procedure, notably the culinary domain, where the imperative now lies in developing algorithms capable of synthesizing detailed, executable cooking recipes from visual prompt. Foremost among these large models is the Large Language and Vision Assistant (LLaVA), an extensively trained, multi-modal model that integrates a vision encoder with a linguistic model to facilitate comprehensive visual and linguistic comprehension. Empirical research underscores LLaVA's proficiency in multi-modal communicative tasks, demonstrating a performance parallel to that of GPT-4, albeit with enhanced processing of visual inputs. When specifically tailored for tasks such as Science QA, LLaVA has attained an exemplary accuracy rate of 92.53%, and in the context of biomedical queries, it has transcended the benchmarks set by prior state-of-the-art supervised models. Such achievements underscore the potential of LLaVA to eclipse comparable leading-edge models like GPT-4 in specific realms through the optimization of computational resources.

The essence of our project focusing on augmenting LLaVA's capability to generate detail recipes that consist of both ingredients and cooking instructions from image-text pairs through the fine-tuning of the model on the Recipe1M+ dataset, an extensive compendium comprising over a million culinary recipes and corresponding images. Our objective is to improve LLaVA's ability at crafting coherent and pertinent recipes derived from visual and textual prompts. This work not only exploring the limits of LLaVA in understanding complex multi modal content but also balancing its applicability in decoding realistic contexts, such as culinary guidance.

The integration of LLaVA within this specific subfield compelling a interesting prospect: whether the transformation of a generative multi-modal model into a specialized system would leads to a better performance. This report will introduce our methodological approaches to fine-tune the LLaVA, the background information of previous work, the preliminary outcomes from our experiments, and also the future checkpoints we would like to go through to accomplish our target.

II. Related Work

A. Large Language-and-Vision Assistant (LLaVA)

Recent advancements and progress have highlighted Large multimodal models (LMMs) and their applications in various fields. Among these, Large Language-and-Vision Assistant (LLaVA) [1] , an end-to-end trained LMM, demonstrates impressive chat capabilities in multimodal scenarios. LLaVA connect the visual encoder of CLIP[2] with the language decoder Vicuna[3] to form a powerful LMM that achieves SoTA on Science QA[4] multimodal reasoning dataset.

With its outstanding performance, LLaVA has also achieved remarkable success in applications in specific domains. LLaVA-Med[5] is an LMM assistant for BioMedicine. Aligned to biomedical vocabulary and then trained on a large-scale insturcion-following data generated by GPT-4 based on biomedical image-text pairs from PMC-15M[6] , LLaVA-Med exhibits excellent conversational ability in multimodal scenarios and can assist with inquires about medical images by following open-ended instructions.
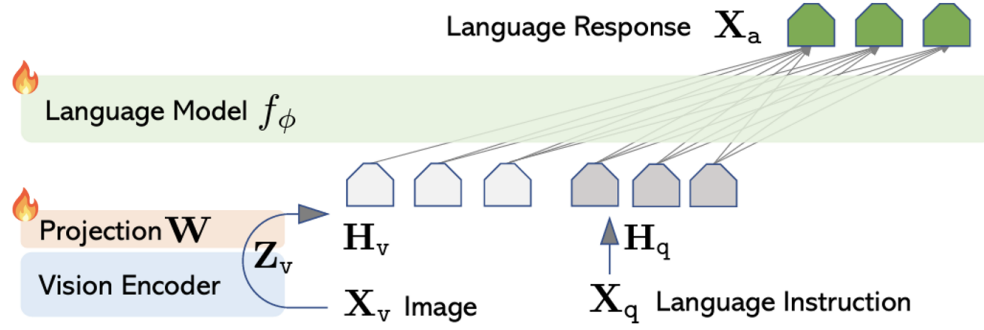
Fig. 1: **LLaVA-Recipe framework**. Model take image (food-related image) and text (question prompt) as input, and generate text response (answer the question based on image instruction). LLaVA-Recipe is specially fine-tuned on ingredient and recipe related visual question answering (VQA) pair to provide not only reliable and technical cooking guidance but also diverse and reasonable food related suggestions. Fire mark means the model is unfreezed during training.

B. Visual Instruction Tuning

The integration of large language model with visual instruction tuning got a significant leap towards in some multimodel AI. For example, Vicuna is a language model developed as part of the broader advancements in natural language processing, which are known for their capabilities in generating human-like text, answering questions, summarizing information, and more. In our approach, LLaVA connects a vision encoder with a language model, specifically integrating the open-set visual encoder CLIP with the language decoder Vicuna, and fine-tuning on generated instructional vision-language data.

C. Recipe Generation

The field of automated recipe generation has expanded significantly with the advances in natural language processing and machine learning. Early attempts, such as the big data approach to culinary recipe design [7], proposing algorithms for novel recipe creation based on flavor compound compatibility. The Chef Watson system [8] exemplifies such an approach, which combines culinary arts and computational creativity to generate innovative recipes.

Cross-modal data processing, particularly the use of both text and image data in recipe analysis, has paved the way for enhanced recipe retrieval and generation systems. The Recipe1M+ model [9] demonstrates how a joint embedding space between images and text can facilitate the retrieval of recipes from food images, which can also be adapted for generating recipes by reversing the retrieval process.

Recent studies have incorporated semantic understanding to improve the relevance and accuracy of generated recipes. For example, Salvador et al. (2017) [10] demonstrated how semantic regularization could enhance recipe and image embeddings, providing a more contextually coherent recipe output. This approach is crucial when fine-tuning models like LLaVA for specific culinary tasks, ensuring that generated recipes are not only novel but also semantically meaningful.

III. METHODOLOGY

A. LLaVA-Recipe Model Structure

The framework of LLaVA and LLaVA-Recipe is shown in Fig. 1. As a multi-modal model, LLaVA and LLaVA-Recipe contains three main module, which is:

- Visual Encoder: Normaly implemented as vision transformer like structure, which take image as input, divided the image into patches processed as image token. Unlike the vision transformer structures that used in classification or detection tasks [11] which provide results with additional token head, the vision encoder here is only used to map the original image patch token into embedding space, the function is known as visual feature learning [12].
- Projection Layer: Introduced to align the feature embedding space that generated by visual encoder with the language model, as not aligned feature would introduces worse generation results [13] [2]. This is achieved by implement few layer of project matrix.
- Language Model: Utilizing the LLMs that can take visual instruction tokens, which is learned from the image of food during training, and text tokens, which is usually the question prompt that queries
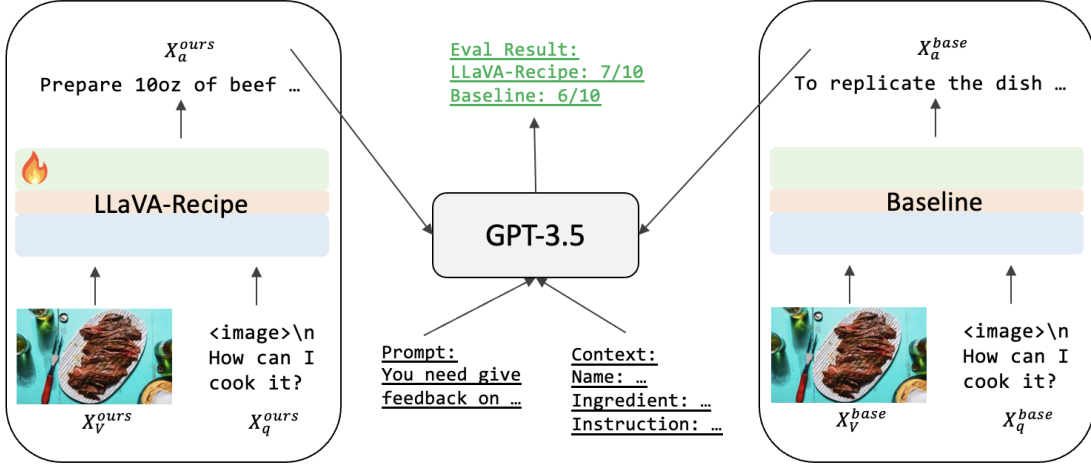
2

Fig. 2: **LLaVA-Recipe evaluation approach**. Both baseline model and fine-tuned model will be tested on same set of image-question pairs. The result will be passed to GPT-3.5 along with the image context, which expressed as dish name, ingredients and instructions, and GPT prompt so that GPT will return the score of answers.

the information about the picture, and generate the text that answers the question prompt.

B. Training and Fine-tune Approach

The proposed method for improving the LLaVA to specialize in recipe generation and food related question-answering from images involves fine-tuning on the Recipe1M+ dataset, a comprehensive collection of over one million cooking recipes and images. Our key idea centers on employing a multimodal training strategy that integrates the visual features from food images with their corresponding textual recipe information. This approach aims to enhance LLaVA's ability to understand and generate complex textual content, such as recipes, based on visual cues.

The training process involves two main stages: pre-training for feature alignment and end-to-end fine-tuning. The image input is the images in Recipe 1M+, and the output supervision is the text in the dataset with corresponding images. Since this dataset didn't provide text instructions, we will simply use some text like: "How to make this dish?" or "Generate a recipe for the dish in this image." and let Chat-gpt to rephrase it for every training sample and add them to the dataset.

The overall training(fine-tune on LLaVa) will be composed by two stages:

- Pre-training projection layer: Initially, with the visual encoder and LLM's weight frozen, pre-training projection layer of LLaVA using a subset of data to align image features with language embeddings, focusing on converting image-text pairs into instruction-following data. This stage helps in aligning the visual features with the LLM's word embeddings.
- End-to-end fine tune: Following that, we would proceed to fine-tune the model end-to-end on the Recipe1M+ dataset, keeping the visual encoder weights frozen while updating both the projection layer and trainable parameters in LLM. This process emphasizes training the model to understand and generate complex textual content based on visual cues from food images, enhancing its capability to generate accurate and relevant recipes.

IV. EXPERIMENTS

A. Datasets

The main dataset Recipe1M+ we suppose to utilize is quite large, this demands large RAM and highs TFLOPs for training or fine-tuning Recipe1M+. Therefore, we started to training with a relative small dataset, which is Food Ingredients and Recipes Dataset with Images, to ensure the pre-train and fine-tune are both working, then we could move to Recipe1M+ to train and fine-tune our model.

Thus, the datasets involved in this paper include:

- Food Ingredients and Recipes Dataset with Images[14]: The small pre-train and fine-tune dataset (with file size around 200MB) initially used for testing. The dataset consist of image-ingredient-instruction pairs which can be proceed and utilized as the ground truth value in training phase.
- Recipe1M+ [15]: The large dataset that consist of 1 million image-ingredient-instruction pairs so that we can randomly select suitable size for better fine-tune quality.

B. Data Processing and Preparation

LLaVA-Recipe accepting specific .json format, so we converted each dataset to this format:

```
1  [
2    {
3      "id": "xxx",
4      "image": "xxx.jpg",
5      "conversations": [
6        {
7          "from": "human",
8          "value": "<image>\n xxx."
9        },
10       {
11         "from": "gpt",
12         "value": "xxx."
13       },
14     ]
15   },
16   ...
17 ]
```

Note that we are using one round conversation for pre-train and multi-round dialogue for fine-tune, we are doing extra processing for different dataset which will is introduced in later sectionss.

1) Small data preparation: In order to fine-tune on the small dataset, we converted the .csv formate file (see Table. I) to acceptable .json file.

#	Title	Ingredients	Instructions	Image	...
...	...	...	...	...	...

TABLE I: Food Ingredients and Recipes Dataset with Images data format in .csv file

2) Recipe1M+ data preparation: Recipe1M+ dataset providing the data in two .json format file, `layer1.json` and `layer2.json`, where `layer1.json` provides annotations information about each recipe, and `layer2.json` providing image locations for corresponding recipe.

`layer1.json` is in this format:

```
1  [
2    {
3      id: String,  // unique 10-digit hex
        string
4      title: String,
5      instructions: [ { text: String } ],
6      ingredients: [ { text: String } ],
7      partition: ('train'|'test'|'val'),
8      url: String
9    }, // each recipe
10   ...
11 ]
```

`layer2.json` is in this format:

```
1  [
2    {
3      id: String,   // refers to an id in layer
        1
4      images: [ {
```

```
5        id: String, // unique 10-digit hex + .
         jpg
6        url: String
7      } ]
8    }, // each recipe
9    ...
10 ]
```

C. Pre-train and Fine-tune on Small dataset

Due to the limited size of the small dataset and the need for pre-training, fine-tuning, and testing, the small dataset will be split into three parts to meet these requirements. According to the observation of LLaVA as well as Llava-Med about implementing pretrain, the data of pretrain is much larger than that of finetune. Therefore, we split the small dataset into 9430,2694,1347 pieces in a 7:2:1 ratio to implement pretrain, finetune, and test. Here, our goal is to use a small dataset to verify that we are on the right track and that pretrain and finetune can improve performance on specific related tasks, so we use a single round of dialogue instead of multiple rounds to simplify the process.

- Pretrain: The original dataset contains a lot of exact numbers in the part about ingredients, and through our experiments, we found that the model generated by using ingredients in pretrain focuses too much on the numbers and cannot generate readable answers. Therefore, in pretrain, we only use the instructions section by randomly selecting the preset ten questions asking how to make the dishes in the figure.
- Finetune: In the dialogue during finetune, we also used only a single round of dialogue. Here, the question is one of a preset set of ten randomly selected questions asking how to make the dishes in the picture. In the answer section, we use ingredients and instructions as answers for fine-tuning.

By comparing with the original LLaVA on small data sets, it can be found that the answers generated by the models obtained by pretrain and finetune for specific related tasks are closer to the answers provided by finetune data. Since the ingredients part of this dataset uses a large number of accurate numbers, which makes the model overemphasize the importance of numbers and generate inaccurate responses, we compare several responses manually instead of directly using GPT3.5 to score.

D. Finetune on Recepe1M+

The implementation of the project as well as the conclusions are mainly based on this data set. Through the previous implementation of pretrain on small data sets and pretrain for single round dialogues, we found that the pretrain step of LLaVA requires a huge amount of data and a lot of computing resources. This makes

Model	Average Score (0-10)
Epoch 1	
1lava-7b	5.497
finetune-IIM-epoch1 (ours)	**6.974**
finetune-noIIM-epoch1 (ours)	<u>6.926</u>
Epoch 3	
1lava-7b	5.636
finetune-IIM-epoch3 (ours)	**7.161**
finetune-noIIM-epoch3 (ours)	<u>7.108</u>

TABLE II: **Compare IIM and noIIM Performance with same training epoch**. **Bold** representing the best score, <u>underline</u> represeting the second best score.

Model	Average Score (0-10)
IIM	
1lava-7b	5.443
finetune-IIM-epoch1 (ours)	<u>6.957</u>
finetune-IIM-epoch3 (ours)	**7.040**
No IIM	
1lava-7b	5.353
finetune-noIIM-epoch1 (ours)	**6.887**
finetune-noIIM-epoch3 (ours)	<u>6.373</u>

TABLE III: **Compare Different Epoches Performance with same IIM Setting**. **Bold** representing the best score, <u>underline</u> represeting the second best score.

it impossible to complete a finetune that pretrains on a large enough dataset and does multiple rounds of conversations in a short period of time. At the same time, since the pretrain step is not necessary for fine-tuning LLaVA, we finally decide to select a sub-dataset of 10K size on the Recipe1M+ dataset, and conduct fine-tuning of multi-round dialogues based on this sub-dataset.

Multi-turn dialogue generation: Since the dataset only contains three pieces of information about the image and the dish, title, ingredients, and instructions, these are common in food-related datasets but are not sufficient to generate effective multi-turn conversation data for finetune. We also wanted the fine-tuned model to be generalizable, i.e., to be able to answer a range of food-related questions rather than just how to make dishes, so GPT-3.5 was used to generate three to five rounds of food-related conversation data for each image. For each image we provide the title of the dish to GPT so that it can generate the associated conversation data.

Ingredients and instructions mentioned (IIM): Since GPT3.5 does not accept images as input, providing only the name of the dish to gpt may be limited and provide incorrect information when generating multiple rounds of dialogue. In addition, experiments on small datasets find that the simple combination of ingredients

and instructions as the answer to related questions will lead to the answer generated by the model is not completely reasonable. To address this potential issue and to compare different dialog generation approaches, we used Ingredients and instructions mentioned (IIM) when generating multiple rounds of dialog using gpt. That is, both the ingredients and instructions of the dish are given to GPT when the conversation is generated. In the recipe1M+ dataset, each dish has multiple images in different states of production, which may appear in other dish related datasets, and we only select one image for each dish. In order to help the fine-tuned LLaVA model be less affected by pictures of different production states and better analyze each dish itself, we expect that IIM is necessary and effective.

E. Evaluation

We evaluate the performance of the baseline model (LLaVA-7b) along with two fine-tuned models on the same test set of image-question pairs, as is shown in Fig. 2. We divided evaluation and compare the performance into 4 groups and only one parameter is changed in each group. The responses generated are then passed to GPT-3.5 together with a carefully designed prompt and the

corresponding information of the test sample, including the dish name, the ingredients, the cooking instructions and the image context. The evaluation of the models are presented in a form of scores on a scale of 0-10.

F. Results

The evaluation results are shown in Table II and III. Our major findings are listed as following.

Finetuned models outperform baseline LLaVA: All finetuned models, regardless of their training epochs and IIM situation, outperform the baseline model of LLaVA (7B) by a large margin. It demonstrates the effect of visual instruction tuning.

IIM improves model performance: We compare the effect of IIM and find out its positive impact on model performance in Table II. On both epoch of 1 and 3, the models with IIM are performing better than the one without IIM. This shows that IIM can boost model performance by helping the models better analyze the dish itself rather than being affected by pictures of dishes at different production states.

More training epochs helps IIM models: We further compare the effect of increasing training epochs in Table III. For IIM model, the performance is boosted by the increased epochs of training. On the other hand, the models without IIM don't benefit from the increased training epochs, resulting in a drop of model performance. The inconsistent result may be explained by the effect of IIM. IIM enhances model performance and this positive effect can be, most likely, further strengthened its effect. By contrast, the model without IIM can't benefit from this strengthened effect, leading to a reduced average score of performance.

G. Discussion

Scaling laws: It describes the relationship between enhancements in system performance, such as accuracy or speed, and increases in resources such as data, computational power, or model size. The performance of our models is significantly constrained by the available computing resources and the size of the datasets, as well as the time of fine tuning.

Data pre-processing In initial tests, the model frequently repeated phrases like "1/2 cup of sugar, and 1/2 cup of butter." This repetition stems from the Vicuna model's algorithm, which tends to select the next token with the highest probability. Such behavior can be traced back to the training datasets, where the ground truth recipes often begin with ingredient lists—typically a sequence dominated by quantities and individual ingredient names. To address this, we pre-processed the data by splitting ingredients and recipe instructions into separate training segments. This adjustment improved the diversity of the model's output, though some outliers persist, indicating a need for further refinement of the dataset.

V. Conclusion

We presented an approach to enhance of the capabilities of LLaVA for the recipe generation tasks. Through visual instruction tuning on the Recipe1M+ dataset, the finetuned LLaVA exhibit excellent visual chat capabilities in generating detailed cooking recipes from image input. We also discovered IIM can boost model performance and more training epochs can further improve LLaVA's capability of recipe generation together with IIM.

VI. Future Work

A. Evaluation

Due to budget constraints as well as traffic constraints imposed by Openai, in this experiment, we can only use GPT-3.5 to evaluate the LLaVA baseline model and its related fine-tuned models. In future work, we plan to use more powerful large language models and large multimodal models for model evaluation, e.g. GPT-4.

The questions used to evaluate in this project are all open-ended questions, for which there exists an objective standard way of calculating. In future work, we plan to add dish related classification problems for better evaluation. For example, please classify the dish in the picture as one of main course, dessert, drink, etc.

B. Pretrain

Due to the lack of time and computing resources, we did not pretrain the recipe1M+ dataset. It would be interesting to compare the performance before and after applying pretrain. In future work, we plan to use large datasets for pretrain such as 600K size. We also noticed that in the pretrain of other Llava-related projects, the dataset of pretrain generalizes more than finetune. Using pretrain, we won't limit ourselves to just cuisine images.

C. Finetune

Due to time constraints and computational resources, we only fine-tuned 4 models. In future work, we plan to vary the parameters to contrast the fine-tuning performance under different conditions. Regarding the choice of model parameter size, we can later compare the gap between 7b and 13b. Llama is also a good model to try when it comes to model selection. In terms of dataset size, 25K is a good way to go.

REFERENCES

[1] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[3] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[4] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," 2022.

[5] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," 2023.

[6] S. Zhang, Y. Xu, N. Usuyama, H. Xu, J. Bagga, R. Tinn, S. Preston, R. Rao, M. Wei, N. Valluri, C. Wong, A. Tupini, Y. Wang, M. Mazzola, S. Shukla, L. Liden, J. Gao, M. P. Lungren, T. Naumann, S. Wang, and H. Poon, "Biomedclip: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs," 2024.

[7] L. R. Varshney, F. Pinel, K. R. Varshney, D. Bhattacharjya, A. Schörgendorfer, and Y.-M. Chee, "A big data approach to computational creativity," *arXiv preprint arXiv:1311.1213*, 2013.

[8] D. Bhattacharjya and Y. Chee, "A big data approach to computational creativity: The curious case of chef watson."

[9] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[10] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning cross-modal embeddings for cooking recipes and food images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3020–3028.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[12] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024.

[13] Y. Zhou, C. Cui, R. Rafailov, C. Finn, and H. Yao, "Aligning modalities in vision large language models via preference finetuning," 2024.

[14] S. Goel and A. Desai, "Food ingredients and recipe dataset with image name mapping," 2024.

[15] J. Marın, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 187–203, 2021.

[16] E. Salin, S. Ayache, and B. Favre, "Towards an exhaustive evaluation of vision-language foundation models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 339–352.

[17] T. Wang, K. Lin, L. Li, C.-C. Lin, Z. Yang, H. Zhang, Z. Liu, and L. Wang, "Equivariant similarity for vision-language foundation models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2023. [Online]. Available: http://dx.doi.org/10.1109/ICCV51070.2023.01102

[18] D. Berrar, *Cross-Validation*, 01 2018.

[19] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," *arXiv preprint arXiv:2310.03744*, 2023.

[20] B. P. Voutharoja, P. Wang, L. Wang, and V. Guan, "Malm: Mask augmentation based local matching for food-recipe retrieval," *arXiv preprint arXiv:2305.11327*, 2023.

[21] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1. Citeseer, 2003, pp. 29–48.

[22] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *Intelligent Data Engineering and Automated Learning–IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013. Proceedings 14*. Springer, 2013, pp. 611–618.

[23] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

CREATE STANDARD FORMAT IN .JSON

```python
def create_json(directory, output_file):
    data = {"training": []}
    subdirectories = [os.path.join(directory, d
        ) for d in os.listdir(directory) if os.path
        .isdir(os.path.join(directory, d))]
    num_directories = len(subdirectories)
    fold_0_limit = int(0.7 * num_directories)
        # 70% of the directories

    for idx, subdir in enumerate(subdirectories
        ):
        fold_number = 0 if idx < fold_0_limit
    else 1
        entry = {
            "fold": fold_number,
            "image": [],
            "label": ""
        }

        for file in os.listdir(subdir):
            filepath = os.path.join(subdir,
    file)
            if file.endswith(".nii.gz"):
                if "seg" in file:
                    entry["label"] = filepath
                else:
                    entry["image"].append(
    filepath)

        if entry["image"] and entry["label"]:
            data["training"].append(entry)

    with open(output_file, 'w') as f:
        json.dump(data, f, indent=4)
```

APPENDIX B
CONVERSATION GENERATION PROMPTS IN GPT-3.5

You are an AI assistant specialized in generating conversations about food related topics.

You are provided with the name of a food image from a online recipe website. The ingredients and the instructions of the recipe will NOT be provided to you. Unfortunately, you don't have access to the actual image.

The name or title of the image is: "title".

Your task is to generate a conversation between a person (User) inquiring about the image and you (Assistant) responding to their questions. The conversation should proceed as though both the User and Assistant are viewing the image, while not referring to the text information.

Below are requirements for generating the questions and answers in the conversation: - Your answer must follow the format of a conversation between a User and an Assistant. - The answers from Assistant should be at least 4 sentences long. And it should be informative and engaging. But do not exceed 2000 tokens. - Try avoiding that the name of the dish is not mentioned in the conversation if possible. - Do not use phrases like "mentioned", "caption", "context" in the conversation.

Instead, refer to the information as being "in the image."
- Ensure that questions are diverse and cover a range of visual aspects of the image. For example, questions can be about the appearance, ingredients, preparation method, or serving suggestions of the dish in the image.
- The conversation should be engaging and informative, providing interesting details about the dish in the image.
- Some questions may involve personal preferences or opinions about the dish, which can be answered based on common knowledge or general food facts. - Some questions may involve comparisons with other dishes or types of cuisine, which can be answered based on general culinary knowledge. - Some questions may ask for recommendations or suggestions related to the dish in the image, which can be answered based on common culinary practices. - Some questions may ask for suitable occasions or settings for the dish in the image, which can be answered based on general food culture. - Some questions may ask for additional information about specific ingredients or cooking techniques used in the dish, which can be answered based on common culinary knowledge. - The conversation should include 3 to 5 turns of questions and answers about the visual aspects of the image.

APPENDIX C
COMPARE OUTPUTS OF THREE ASSISTANTS

Visual input example, Wild-Blackberry-Vanilla-Cupcakes-Sweetopia-small:

Source: http://pic2recipe.csail.mit.edu/
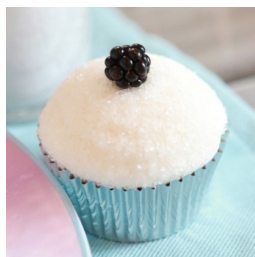
User: Could you guide me on how to prepare the dish displayed in this image?

IIM-e1: Certainly! The image shows a delightful dessert that involves a few simple steps. To prepare it, you would need to preheat the oven to 350 degrees F. Then, you would need to mix the cake mix, sugar, egg, and milk together to form a batter. After that, you would need to spoon the batter into cupcake liners and bake them for about 15 minutes. Once baked, you would need to let them cool before adding a blackberry on top and serving.

User: The cupcakes look so moist and fluffy! How are they made to achieve that texture?

IIM-e1: The moist and fluffy texture of the cupcakes in the image is achieved by using a combination of ingredients like butter, sugar, eggs, and flour. The butter is creamed with sugar to create a light and airy base, while the eggs add richness and moisture. The addition of baking powder helps the cupcakes rise and become fluffy. The use of self-raising flour also contributes to the overall lightness of the cupcakes.

User: Could you guide me on how to prepare the dish displayed in this image?

noIIM-e1: Of course! The dish in the image features a delightful combination of flavors and textures. To prepare it, you'll need to start by making a simple syrup with blackberry juice and sugar. This syrup will be used to poach the blackberries, which will then be used to create a delicious filling for the cupcakes. The cupcakes themselves are made with a light and fluffy batter, giving them a moist and tender texture. To finish, you can top the cupcakes with a dollop of whipped cream and a sprinkle of fresh blackberries for a beautiful presentation.

User: Could you guide me on how to prepare the dish displayed in this image?

LLaVA-7b: To prepare the dish shown in the image, you would need to follow these steps: Bake a cupcake: Choose your favorite cupcake recipe and bake it in a cupcake tin. Cool the cupcake: Allow the baked cupcake to cool completely on a wire rack. Add the topping: Once the cupcake is cooled, you can add the topping. It appears to be a dusting of powdered sugar or a similar sweet, white substance. Place the berry: On top of the sugar-covered cupcake, place a fresh blackberry. Serve: The cupcake is now ready to be served. Enjoy your sweet treat!

TABLE IV: Example prompts reveal distinct model capabilities: "noIIM-e1" primarily describes scenes, while "IIM-e1" follows user instructions accurately and converses more naturally. IIM-e1 matches the detailed and context-aware responses of the fine-tuned "LLaVA-7b," enhancing the interaction with improved conversational engagement.