MASID: Mobile Artificial Intelligence System for Image-Based Dish Generation: A Computer Vision and Prompt Engineering Application for Filipino Recipe Generation

Joackim Pether R. Agno[1], Vince Craig S. Buenaventura[1] , Andrei Vincent V. Collado[1], Robert Patrick D. Francisco[1], and Melvin K. Cabatuan[1]
[1] Gokongwei College of Engineering, De La Salle University
**Corresponding Author: email address@dlsu.edu.ph*

Abstract: MASID (Mobile Artificial-Intelligence System for Image-Based Dish Generation) is an Android application designed to generate unique Filipino recipes from images of available ingredients. The study addresses common challenges faced by local eateries and households, such as limited menu variety and food waste, through an AI-powered solution. The system combines a small object detection model, YOLOv12s, to detect ingredients and a fine-tuned small language model, Qwen 2.5 7B, to produce step-by-step recipes. In performance tests, MASID registered a mean Average Precision (mAP) of 96.7%, 94.2% precision, and 93.8% recall in 34 Filipino food classes. Recipes produced by MASID complied with carinderia constraints, presenting simplified, economical, and ingredient-efficient dishes. Recipe generation produced coherent, context-aware outputs aligned with carinderia constraints, such as simple steps and minimal ingredients. The application maintained an end-to-end average runtime of 20–30 seconds, enabling real-time interaction. A Flask-based backend was used for communication and data handling, while the user interface was developed in Flutter to ensure ease of use. These results demonstrate MASID's potential to create unique Filipino recipes tailored to carinderias or households with the help of AI.

Key Words: Small Object Detection Model, Small Language Model, Carinderia, Computer Vision, Prompt Engineering

1. INTRODUCTION

Filipino cuisine, with staples like adobo, sinigang, and pancit, showcases the culture's creativity and local resourcefulness. However, many households and small businesses struggle with meal planning and recipe innovation due to time constraints, limited resources, and technological access (World Wide Fund for Nature [WWF], 2024). This often results in food wastage, fast food dependence, and missed culinary opportunities, as regional contexts and ingredient limitations are rarely addressed by online platforms.

Local carinderias and home kitchens face deeper challenges, such as low access to digital tools and lack of nutritional awareness, hindering culinary creativity and leading to repetitive meals. MASID (Mobile Artificial-Intelligence System for Image-Based Dish Generation) tackles these issues by utilizing advancements in computer vision and language modeling to provide a personalized and culturally relevant cooking experience.

With features like image-based ingredient detection and AI-generated Filipino recipes, MASID allows users to create new dishes from limited ingredients. Designed for small eateries and households, it employs a lightweight object detection model and a fine-tuned language model, ensuring efficiency even on low-power devices. The app emphasizes accessible technology and user feedback, promoting community engagement, cultural preservation, and sustainable cooking practices.

2. PRIOR WORKS

2.1 Ingredient Detection and Recognition Systems

AI systems have significantly improved in recognizing ingredients using object detection models. NutrifyAI by Han et al. (2024) combined YOLOv8 and Edamam API to recognize food items and provide nutritional information. However, it is limited by class imbalance issues and prediction uncertainty due to visually similar food categories and low confidence scores. These studies highlight the potential of AI models but also reveal limitations for more complex and diverse Filipino dishes.

Other works, such as Chef Dalle by Hannon et al. (2024) and Smart Cuisine by Kansaksiri et al. (2023), have explored multimodal inputs like image, voice, and text for ingredient recognition and meal suggestions. While these systems offer high interactivity and flexibility, they rely heavily on pretrained models like GPT-Vision, which could not adapt well to regional cuisines like Filipino dishes due to cultural limitations in their training.

These existing solutions show crucial gaps in regional coverage, dataset variety, and model generalization. This research aims to solve this problem with a lightweight, high-accuracy ingredient detection model catered to Filipino ingredients.

2.2 Recipe Generation Models

Recent models like Ratatouille by Goel et al. (2022) and FIRE by Chhikara et al. (2024) leverage deep learning to generate cooking instructions, titles, and ingredients using architectures like GPT-2, Vision Transformers, and T5. While FIRE enabled structured and customizable outputs, it struggled with coherence when generating novel recipes outside of its training data. Likewise, Ratatouille lacked details on the ingredient quantity and occasionally produced incoherent steps.

Leveraging language models has its challenges. Vij et al. (2025) conducted a benchmark study on the performance of small language models (SLMs) in recipe generation. Using the Food.com dataset of over 180,000 recipes, they fine-tuned models like FPT-2 small, T5-small, SmolLM-135M, and larger variants like SmoLM-1.7B and Phi-2 through QLoRA adapters to manage memory constraints. Interestingly, some large models showed degradation in specific metrics post-fine-tuning. This suggests that size does not guarantee improved domain performance.

3. METHODOLOGY

3.1 Data Gathering

The ingredient dataset includes several Filipino ingredients acquired through webscraping several online Filipino recipe websites, and open platforms like Roboflow, or self-taken images. Table 1 shows the summarized sources for the datasets used.

Ingredients	**Recipes**
• Roboflow • Self-taken Photos	• Allrecipes.com • KawalingPinoy.com • KusinaSecrets.com • PanlasangPinoy.com • CookPad.com • Other online sources of Filipino recipes

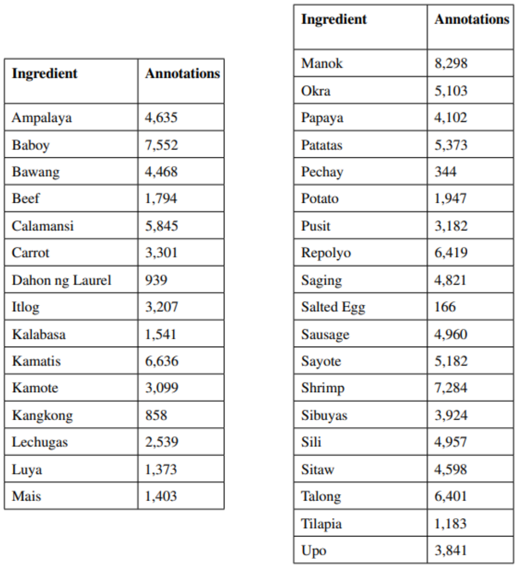
Table 1: Dataset Sources

Below are the recipe data used for SLM training:
(1) Recipe name, (2) Preparation Time, (3) Cook Time, (4) Total Time, (5) Servings, (6) Ingredients, (7) Instructions, (8) Description. The sample recipe data is shown in Figure 1.

Recipe Name	Prep Time	Cook Time	Total Time	Servings	Ingredient Names	Full Ingredients	Directions	Description
string	string	string	string	float64	string	string	string	string
Suam na Mais	10 minutes	25 minutes	35 minutes	4	1. white corn kernel…	1. 30 ounces white corn…	1. Heat oil in a cooking pot.…	null
Chicharon Bulaklak…	10 minutes	2 hours 10…	2 hours 20…	4	1. pork masentery 2…	1. 2 lbs. pork…	1. Prepare the chicharon…	null

Fig. 1: Recipe Data

The total number of Filipino Recipes that were used for this study is 100 recipes, while the ingredients included in the dataset comprise 42,664 images and 34 Filipino Food classes that is shown in Table 2.



Ingredient	Annotations
Ampalaya	4,635
Baboy	7,552
Bawang	4,468
Beef	1,794
Calamansi	5,845
Carrot	3,301
Dahon ng Laurel	939
Itlog	3,207
Kalabasa	1,541
Kamatis	6,636
Kamote	3,099
Kangkong	858
Lechugas	2,539
Luya	1,373
Mais	1,403
Manok	8,298
Okra	5,103
Papaya	4,102
Patatas	5,373
Pechay	344
Potato	1,947
Pusit	3,182
Repolyo	6,419
Saging	4,821
Salted Egg	166
Sausage	4,960
Sayote	5,182
Shrimp	7,284
Sibuyas	3,924
Sili	4,957
Sitaw	4,598
Talong	6,401
Tilapia	1,183
Upo	3,841

Table 2: Ingredient Classes

3.2 Data Cleaning/Filtering

Over 60,000 raw images of Filipino ingredients were collected from Roboflow and personal photos. These images underwent manual annotation using bounding boxes in Roboflow Annotate, with strict adherence to guidelines to prevent mislabeling. A verification step followed to correct inconsistencies. The dataset was formatted into YOLO-compatible labels with normalized coordinates and divided into training (70%), validation (20%), and test (10%) sets, resulting in a balanced final dataset of 42,664 images.

Additionally, a survey of 21 carinderia owners established criteria for filtering a dataset of 2,000 Filipino recipes from various websites. These criteria defined requirements related to preparation and cooking times, as well as ingredient types, ensuring that the dataset complies with carinderia standards and enabling the fine-tuned model to generate appropriate recipes.

3.3 System Architecture

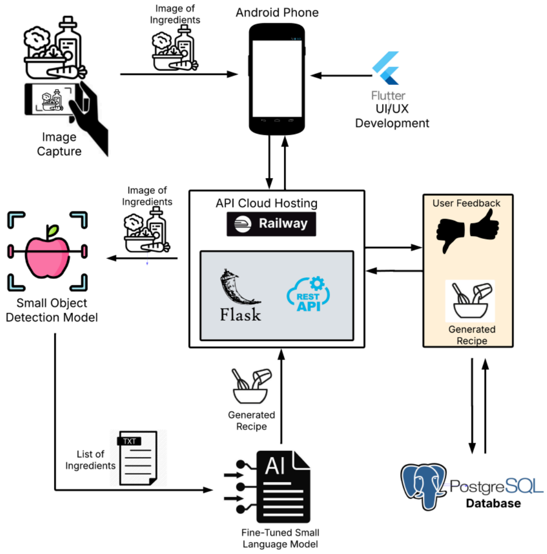


Fig. 2: System Architecture

The system is designed for ingredient recognition and recipe generation using AI and web technologies. Users upload an image of their ingredients via a Flutter mobile app. The image is processed with the SODM algorithm for classification, and the identified ingredients are sent to SLM for recipe generation. User feedback is managed by a Flask backend hosted on Railway, with RESTful API facilitating communication between the Android app and the server.

3.4 Small Object Detection Model

The figure below shows the *Small Object Detection Life Cycle* used in the development of the SODM.

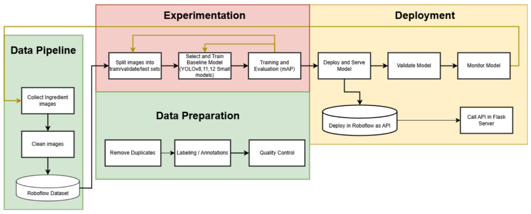


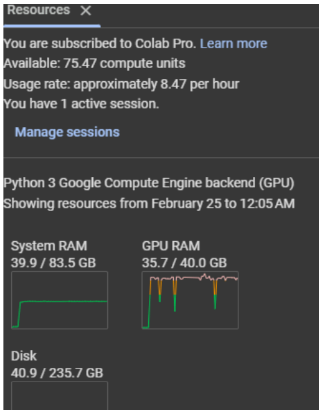
Fig. 3: Small Object Detection Life Cycle

This research utilizes small object detection models including Roboflow Object Detection (fast), Roboflow YOLOv11/12 (fast), and manually trained YOLOv8, 11, and 12 models, based on a CNN architecture with a custom dataset of Filipino food ingredients filtered from carinderia criteria. Due to Google Colab's constraints, smaller variants (YOLOv8s, YOLOv11s, YOLOv12s) were considered, with YOLOv12s selected for optimal performance, using 39.9/83.5GB of System RAM, 35.7/40.0GB of GPU RAM, and 40.9/235.7GB of DISK.

The model is trained on 29,000 images (with 4,630 for validation), resizing each to 640×640 pixels. Training lasts for 50 epochs with a batch size of 128 and the AdamW optimizer, incorporating mixed precision training for efficiency, along with in-memory caching to reduce lag. Once trained, the YOLOv12 model outputs bounding boxes with class labels (e.g., 'ampalaya', 'chicken') and confidence scores for integration into an Android application via Roboflow's API.

Table 3: Comparison of Trained SODM

Model Size	Batch Size	Training Time (Hours)	Compute Units Used	Cost Per Single Train (USD)	Cost Per Single Train (PHP)
Small	128	6.67	60	$6	₱348
Medium	65	16	144	$14.40	₱835
Large	32	30	270	$270	₱1,566



You are subscribed to Colab Pro. Learn more
Available: 75.47 compute units
Usage rate: approximately 8.47 per hour
You have 1 active session.

Python 3 Google Compute Engine backend (GPU)
Showing resources from February 25 to 12:05 AM

System RAM 39.9 / 83.5 GB
GPU RAM 35.7 / 40.0 GB
Disk 40.9 / 235.7 GB

Fig. 4 Hardware Usage Rate of Training a (small-model) using Nvidia A100 GPU

Figure 4 shows typical RAM and GPU usage Training a small YOLOv12s model on a single NVIDIA A100 GPU in Google Colab Pro is more practical than medium or large models due to lower computational needs and costs. A small model takes about 6.7 hours and costs $6 or ₱348, while medium models take around 16 hours and cost $14.40 or ₱835, and large models require about 30 hours and cost $27 or ₱1,566. Smaller models also allow for quicker iterations and reduce the risk of session crashes, making them ideal for MASID's mobile recipe detection application.

3.5 Small Language Model

The fine-tuning process started with a corpus of around 60,000 recipes from Food.com, cleaned into a CSV format by removing duplicates and incomplete rows. Key features for the model included recipe name, prep/cook time, servings, ingredient list, and instructions. This cleaned dataset was uploaded to Hugging Face for easy access.

Each recipe was reformatted into the Alpaca style with "Instruction," "Input," and "Response" to help the model transform ingredient lists into full recipes. Researchers utilized the Qwen-2.5 7B model, applying Low-Rank Adaptation (LoRA) with rank and scaling factors of 16, allowing less than 0.5% of the network's parameters to be trainable. The primary model's weights stayed frozen while the LoRA matrices learned recipe specifics.

After a general pre-training phase, which included recipes from approximately 2,000 Filipino dishes, the model was further refined. Filipino recipe data was gathered from diverse sources, filtering for main ingredient names and excluding basic pantry items. Utilizing this dataset, the model was retrained in the Alpaca format to generate comprehensive recipes based solely on the primary ingredients.

```
SYSTEM_INSTRUCTION = (
    "You are a Filipino chef specialized in unique Filipino karinderya-style "
    "recipes. Follow these rules:\n"
```

```
   "1. Do NOT require the following tools: blender, food processor, mixer, "
   "pressure cooker, air fryer, microwave, sous vide, torch.\n"
   "2. Effort level must be simple: Prep time 10–20 min, Cook time 10–20 min, "
   "Total active time  must be at most 1 hour (longer passive simmering
allowed).\n"
   "3. Return the fields in this order:\n"
   "Recipe name, Prep time, Cook time, Total time, Servings, "
   "Full Ingredients (numbered list), Instructions (numbered list), "
   "Description (1–2 sentences)."
)

alpaca_prompt = (
   "Below is an instruction that describes a task, paired with an input that "
   "provides further context. Write a response that appropriately completes the
request.\n\n"
   "### Instruction:\n{}\n\n### Input:\n{}\n\n### Response:\n{}"
)

user_input = (
   "Generate a unique Filipino karinderya recipe using: "
   "pork belly, onion, garlic, shrimp paste, thai chili peppers, coconut cream "
   "Format ingredients and instructions as numbered lists."
)
```

Listing 1: Dataset Format

The instructions incorporate the karinderya setting, requiring the model to generate only recipes that adhere to these rules: they must be prepared within an hour and should not require uncommon appliances that are typically unavailable in Filipino households.

3.6 Food Safety and Recipe Standardization

The application will guarantee that every time it generates a recipe with the same ingredients, it provides a standardized result, meaning users can expect a reliable, consistent, and user-friendly recipe when using the application. The recipe standardization of the United States Department of Agriculture (USDA) is composed of three (3) phases: **recipe verification** - identifying the recipe, sourcing the ingredients, and writing the recipe in detail. **Recipe evaluation (tasting)** - informal evaluation, formal evaluation; and **Quantity adjustment -** ensure that even if the ingredient quantity is adjusted, the recipe remains proportionate and delivers the same results.

In addition, quality control will be implemented after the recipe standardization to ensure the safe consumption of the recipes generated. Safety measures are based on the Hazard Analysis & Critical Control Points (HACCP) that are established by the Food and Drug Administration (FDA). This is done through fine-tuning and prompt engineering.

3.7 Mobile Application Integration

Mobile Frontend Architecture

The mobile application, developed using Flutter, takes advantage of its cross-platform capabilities and material design guidelines. Each prominent feature is accessed through separate screens: the Home screen features a split design, while the Image Capture and Camera screens allow image capture or selection via the ImagePicker package. The Inventory screen lets users manage ingredients, and the Conversations screen presents recipes in a chat-like interface, enabling users to favorite, share (with optional dish photos), and rate recipes. Each UI module is contained in its own Dart file (e.g., home.dart, camera.dart) for modularity and maintainability.

Backend Integration

The mobile application interacts with a Flask-based RESTful API that serves as a data hub. When users capture or upload a photo, it's encoded in Base64 and sent via an HTTP POST request. The server processes the image using a YOLO-based small object detection model, returning the image with bounding boxes and a list of detected ingredients. These ingredients are then sent to another API endpoint, where a fine-tuned small language model generates a personalized Filipino recipe. User interactions like favoriting, sharing, and rating recipes are tracked and stored in a PostgreSQL database through additional Flask endpoints. The backend runs on Railway Cloud, ensuring fast performance and reliable data storage with effective error handling and caching for real-time responsiveness.

Standards and Overall Integration

The entire process of integration is in accordance with best practices laid out in ISO/IEC 25010 and ISO 9241-210, and it focuses on usability, efficiency in performance, and user experience. The user interface of the mobile app is consistent and simple, making it easy for even a new user to navigate through the system.

4. RESULTS AND DISCUSSION

The ingredient detection component of MASID achieved significant performance improvements through progressive model refinement. The initial chosen model to be deployed in MASID is the Roboflow YOLOv12-based object detection model, which attained a mean Average Precision (mAP) of 96.7%, with a precision of 94.2% and recall of 93.8%, surpassing the target thresholds of 75% mAP, 78% precision, and 73% recall.

Table 4: Comparison of Trained SODM

Model	mAP	Precision	Recall
COLAB YOLOv11s	96.7%	94.2%	93.8%
RF-YOLOv11s	88.3%	88.5%	85.8%
RF-YOLOv12s	96.7%	88.5%	90.4%
Roboflow 3.0	88.3%	83.9%	86.3%
COLAB YOLOv8s	88.4%	85.4%	83.7%
RF-DETR	90.6%	96.0%	89.0%

Comparative evaluation with prior systems reveals MASID's advantage. Previous studies achieved an average mAP of 78.5% and recall of 72.8% using only 13 ingredient classes, whereas MASID was trained on a more diverse set of 34 ingredient classes and achieved higher performance scores .

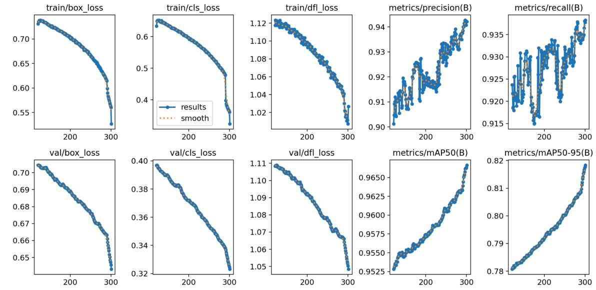


Fig. 5: SODM Training Loss Graph

The end-to-end object detection process, including upload, prediction, and bounding box generation, consistently maintained an average runtime of 20 to 30 seconds, aligning with the system's objective of real-time responsiveness for recipe generation.

For the finetuned small language model, shown below are the BLEU and ROGUE results:

Evaluation using General Recipe Dataset		
Model	**BLEU-4**	**ROGUE-L**
Qwen 2.5 7B	0.1520	0.3786
Qwen 2.5 7B (finetuned)	0.3208	0.4367

Table 5: Baseline Model and Finetuned Model comparison using BLEU and ROGUE

In recipe generation, where both correct ingredient wording and inclusion of all necessary steps matter, the two metrics, BLEU and ROGUE, provide concise proxies for usefulness without requiring manual judgement. Comparing the baseline model to the fine-tuned model, BLEU-4 rises from 0.152 to 0.321 and ROGUE-L improves slightly from 0.379 to 0.437. The large gain in BLEU dictates that the fine-tuned model reproduces many more exact n-gram fragments found in the recipes. This suggests that the fine-tuned model generated better ingredient names, quantities, and conventional culinary phrases. The ROGUE-L dictates that the overall sequence and coverage of key information were already reasonable in the baseline and have been refined. These results show that the fine-tuning process polished lexical accuracy and phrasing while delivering a smaller improvement in holistic recipe completeness and order. The comparison of the model evaluated to the test set of the Filipino recipe dataset is shown in the table below:

Evaluation using Filipino Recipe dataset		
Model	**BLEU-4**	**ROGUE-L**
Qwen 2.5 7B (finetuned)	0.0798	.2235
MASID Model	0.1545	0.3147

Table 6: Finetuned model and MASID model

The MASID model outperformed the fine-tuned Qwen 2.5 7B on the Filipino recipe dataset, achieving better BLEU-4 and ROUGE-L scores. This superior performance is due to MASID's training with LoRA adapters on 2,000 Filipino recipes, allowing it to effectively capture the unique linguistic and procedural patterns of Filipino cuisine. In contrast, the general fine-tuned model struggled outside its training domain, highlighting the need for culturally accurate fine-tuning in localized recipe generation.Shown in the listing below is a sample Filipino recipe generated using the MASID model:

```
Recipe name: Ginataang Baboy (Pork in Coconut Milk)
Prep time:  10 minutes
Cook time:  1 hour
Total time: 1 hour 10 minutes
Servings:   4.0
Full Ingredients:
1. 2 pounds pork belly cut into 1-inch cubes
2. 1 large onion chopped
3. 4 cloves garlic crushed
4. 1 tablespoon shrimp paste
5. 4 Thai chili peppers chopped
6. 2 cups coconut cream
7. 1/2 cup water
8. 1 tablespoon fish sauce patis
9. Salt and pepper to taste
Instructions:
1. In a large pot over medium heat, combine pork, onion, garlic, shrimp paste,
and chili peppers. Cook, stirring occasionally, until the pork is lightly browned
and the onions are soft.
2. Add coconut cream and water. Bring to a boil, skimming any foam that floats
on top.
3. Lower heat, cover, and simmer for about 40-45 minutes or until the pork is
tender and the sauce is reduced.
4. Season with fish sauce, salt, and pepper to taste. Serve hot with steamed rice.
Description: Ginataang Baboy is a classic Filipino dish where pork belly is
simmered in coconut milk until tender, creating a rich, creamy stew that's
perfect with steamed rice.
```

Listing 2: Sample MASID model output

The prototype Android mobile application features a user-friendly home page with three main buttons, each with distinct functions. The box icon provides access to the digital inventory system, where users can view and manage previously scanned ingredients. The camera icon allows users to capture images of ingredients using the Small Object Detection Model (SODM), while the upload icon enables users to select and upload ingredient photos from their devices.

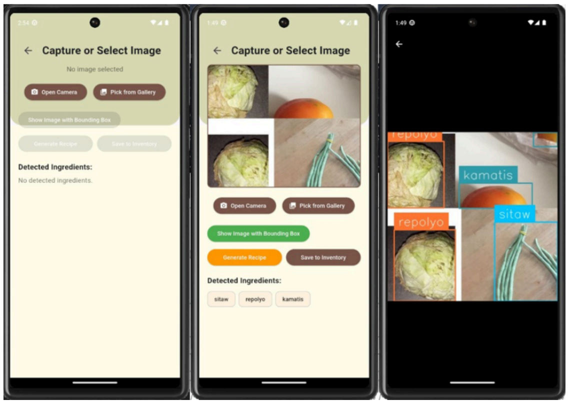
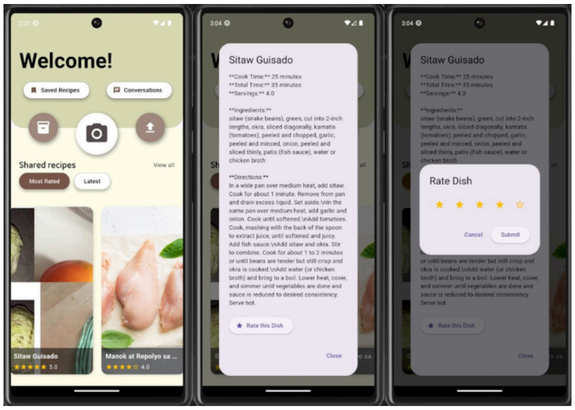


Fig. 7: Actual Ingredient Detection UI



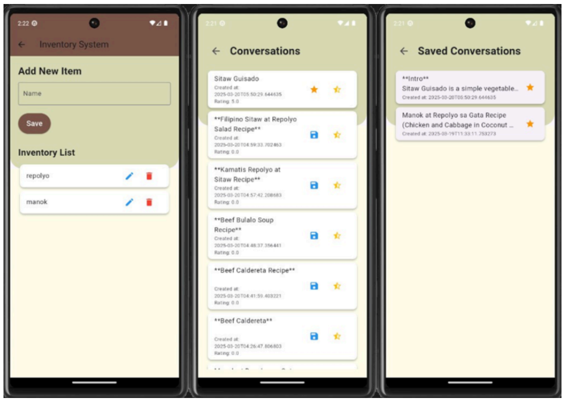
Sitaw Guisado

Cook Time: 25 minutes
Total Time: 35 minutes
Servings: 4.0

Ingredients:
sitaw (snake beans), green, cut into 2-inch lengths, okra, sliced diagonally, kamatis (tomatoes), peeled and chopped, garlic, peeled and minced, onion, peeled and sliced thinly, patis (fish sauce), water or chicken broth

Directions:
In a wide pan over medium heat, add sitaw. Cook for about 1 minute. Remove from pan and drain excess liquid. Set aside.\nIn the same pan over medium heat, add garlic and onion. Cook until softened.\nAdd tomatoes. Cook, mashing with the back of the spoon to extract juice, until softened and juicy. Add fish sauce.\nAdd sitaw and okra. Stir to combine. Cook for about 1 to 2 minutes or until beans are tender but still crisp and okra is cooked.\nAdd water (or chicken broth) and bring to a boil. Lower heat, cover, and simmer until vegetables are done and sauce is reduced to desired consistency. Serve hot.

Fig. 8: Actual Recipe Generation Page UI



Sitaw Guisado
Created at: 2025-03-20T05:50:29.644635
Rating: 5.0

Filipino Sitaw at Repolyo Salad Recipe
Created at: 2025-03-20T04:59:33.702463
Rating: 0.0

Kamatis Repolyo at Sitaw Recipe
Created at: 2025-03-20T04:57:42.208683
Rating: 0.0

Beef Bulalo Soup Recipe
Created at: 2025-03-20T04:48:37.356441
Rating: 0.0

Beef Caldereta Recipe
Created at: 2025-03-20T04:41:59.403221
Rating: 0.0

Beef Caldereta
Created at: 2025-03-20T04:26:47.806803
Rating: 0.0

Intro
Sitaw Guisado is a simple vegetable...
Created at: 2025-03-20T05:50:29.644635

Manok at Repolyo sa Gata Recipe (Chicken and Cabbage in Coconut ...
Created at: 2025-03-19T11:33:11.753273

Fig. 9: Actual ConversationsPage UI

5. CONCLUSIONS

MASID (Mobile Artificial-Intelligence System for Image-Based Dish Generation) demonstrated the integration of artificial intelligence in supporting Filipino culinary practices, particularly among small food businesses and home cooks. By using a lightweight and accurate object detection model (YOLOv12) and a small language model (Qwen 2.5 7B), the system was able to achieve a 96.7% mAP, 94.2% precision and 93.8% recall and achieved an efficient end-to-end processing time of under 30 seconds, which meets and even surpasses expectations. Moreover, the BLEU and ROGUE comparisons showed how training SLMs can improve on domain specific tasks like Filipino recipe generation. While user satisfaction metrics are yet to be fully surveyed, MASID has already established a solid foundation for its overall purpose.

6. ACKNOWLEDGMENTS

The researchers, in profound gratefulness, extend this most solemn gratitude to our parents for their unfailing support and encouragement. We also sincerely thank Aeron Aguilar for contributing to and helping shape this paper. Thank you to all who have provided helping hands.

7. REFERENCES

Berkowitz, R. T., Wadden, T. A., Tershakovec, A. M., & Cronquist, J. L. (2003). Behavior therapy and sibutramine for the treatment of adolescent obesity [Electronic version]. Journal of the American Medical Association, 289, 1805-1812.

Carmona, R. H. (2004, March 2). The growing epidemic of childhood obesity. Testimony before the Subcommittee on Competition, Foreign Commerce, and Infrastructure of the U.S. Senate Committee on Commerce, Science, and Transportation. Retrieved October 10, 2004, from http://www.hhs.gov/asl/testify/t040302.html

Chhikara, P., Chaurasia, D., Jiang, Y., Masur, O., & Ilievski, F. (2024). FIRE: Food image to recipe generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 2183–2192). https://openaccess.thecvf.com/content/WACV2024/papers/Chhikara_FIRE_Food_Image_to_REcipe_Generation_WACV_2024_paper.pdf

Crtiser, G. (2003). Fat land: How Americans became the fattest people in the world. Boston: Hougton Mifflin.

Değerli, A. H., & Tatlısu, N. B. (2023). Cooking with ChatGPT and Bard: A study on competencies of AI tools on recipe correction, adaptation, time management, and presentation. *Journal of Tourism and Gastronomy Studies, 11*(4), 2659–2673. https://doi.org/10.21325/jotags.2023.1312

Demition, A. D. R., Narciso, Z. A. L., & Paglinawan, C. C. (2024). Filipino meal recognition scale with food nutrition calculation and smart application. *Engineering Proceedings, 74*(1), 54. https://doi.org/10.3390/engproc2024074054

Duenwald, M. (2004, January 6). Slim pickingsL Looking beyond ephedra. The New York Times, p. F1. Retrieved October 12, 2004, from LexisNexis.

Goel, M., Chakraborty, P., Ponnaganti, V., Khan, M., Tatipamala, S., Saini, A., & Bagler, G. (2022). Ratatouille: A tool for novel recipe generation. arXiv preprint arXiv:2206.08267. https://doi.org/10.48550/arXiv.2206.08267

Han, M., Chen, J., & Zhou, Z. (2024). NutrifyAI: An AI-powered system for real-time food detection, nutritional analysis, and personalized meal recommendations. arXiv preprint arXiv:2408.10532. https://doi.org/10.48550/arXiv.2408.10532

Hannon, B., Kumar, Y., Li, J. J., & Morreale, P. (2024). Chef Dalle: Transforming cooking with multi-model multimodal AI. Computers, 13(7), 156. https://doi.org/10.3390/computers13070156

Irkaal. (2021, January 5). *Food.com – Recipes and Reviews* DatasetData setDataset. Kaggle. Retrieved June 25, 2025, from https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews

Kansaksiri, K., & Wongsuphasawat, K. (2023). Smart Cuisine: Generative recipe & ChatGPT powered nutrition assistant. Procedia Computer Science, 227, 1526–1533. https://doi.org/10.1016/j.procs.2023.10.193

Vij, A., Liu, C., Nair, R. A., Ho, T. E., Shi, E., & Bhowmick, A. (2025). Fine-tuning language models for recipe generation: A comparative analysis and benchmark study. arXiv preprint arXiv:2502.02028. https://doi.org/10.48550/arXiv.2502.02028

World Wide Fund for Nature. (2024). Solving the great food puzzle: Philippines – Food system type 2. https://wwfint.awsassets.panda.org/downloads/solving-the-great-food-puzzle-wwf-2024.pdf