# Privacy-Preserving Smart Refrigerator Using CNN for Ingredient Detection and SLM for Recipe Recommendation in IoT Edge Environments

Joon Soo Yoo, So Hee Choi, Geon Woo Jeon

***Abstract*—This study focuses on the development of a system for smart refrigerators that integrates food ingredient recognition with interactive recipe recommendations. The system combines a Convolutional Neural Network (CNN) model, which classifies stored ingredients using images captured by an internal camera, and a Small Language Model (SLM), which provides personalized recipe suggestions based on the detected items. The CNN model identifies ingredients such as cucumbers and oysters, while the SLM suggests possible recipes, enabling user interaction through natural language. For example, the system can prompt, ”With these ingredients, you can make spaghetti. Would you like the recipe?” Additionally, when a user requests to make a dish, such as ”I want to make spaghetti,” the system provides guidance on any additional ingredients needed. All data processing is performed locally within the refrigerator, ensuring privacy by preventing external data transmission. This paper proposes an innovative, privacy-centric smart home appliance solution that offers real-time, customized recipe recommendations.**

***Index Terms*—Smart Refrigerator, Convolutional Neural Network, Small Language Model, Recipe Recommendation**

## I. Introduction

In recent years, the integration of Artificial Intelligence (AI) and Internet of Things (IoT) devices has provided innovative solutions for daily life. One prominent example is the application of advanced AI models in **smart refrigerators** to recognize food items, suggest recipes, and offer an interactive user experience. However, concerns about data privacy and the complexity of real-time AI processing have limited the widespread adoption of such systems.

This project aims to develop a **privacy-preserving, real-time recipe recommendation system**. The system leverages a **Convolutional Neural Network (CNN)** to detect ingredients within the refrigerator and employs a **Small Language Model (SLM)** to provide interactive recipe recommendations. The core features of the system are as follows:

- A lightweight CNN model detects the ingredients inside the refrigerator.
- The detected ingredients serve as input for the SLM, which interacts with the user and suggests recipes based on the available ingredients.
- All processing is conducted locally on the device, ensuring user privacy by preventing data transmission to external servers.

The CNN model is trained to recognize individual food items from images captured inside the refrigerator and outputs a vector representing the detected ingredients. This vector is passed to the SLM, which suggests recipes to the user through natural language interactions. For instance, after detecting ”kimchi” and ”salt,” the system could propose making kimchi stew.

By processing the entire AI pipeline on the device, the system not only enhances data privacy but also provides intelligent recommendation functionality without relying on external servers or cloud services. Therefore, this system offers a secure, privacy-preserving solution for modern households.

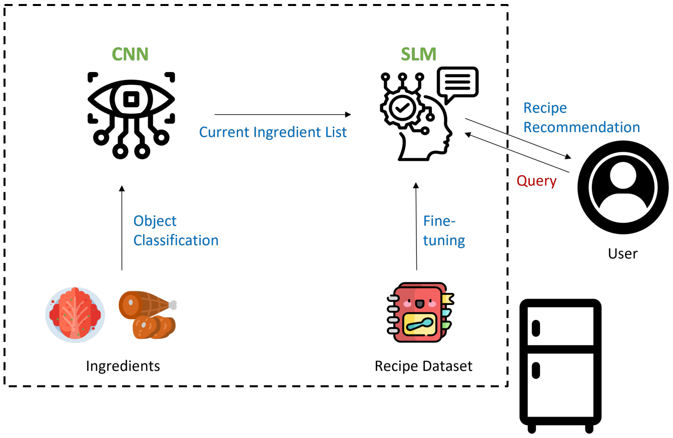


Fig. 1. Overview of Our Work

### A. Related Works

In [1], a study was conducted using a variant of ResNet50 to detect food ingredients. This study did not specifically target IoT device-level implementation, but focused on ingredient detection and recipe recommendation. The Food101 [2], Fruit 360 [3], and UECFOOD256 [4] datasets were used to train the model, with data augmentation techniques applied to improve performance. Specifically, image rotation (45 degrees), horizontal and vertical translation (20%), scaling (20%), horizontal flipping, and 20% shearing were performed. The Adam optimizer was used for training, achieving 99.71% training accuracy and 92.6% validation accuracy. Of the total 9,856

Manuscript created October, 2020; This work was developed by the IEEE Publication Technology Department. This work is distributed under the LATEX Project Public License (LPPL) ( http://www.latex-project.org/ ) version 1.3. A copy of the LPPL, version 1.3, is included in the base LATEX documentation of all distributions of LATEX released 2003/12/01 or later. The opinions expressed here are entirely that of the author. No warranty is expressed or implied. User assumes all risk.

images, 70% were used for training, 20% for testing, and 10% for validation.

## II. Our Model

The proposed system is centered around a **real-time interactive recipe recommendation system**, which operates using two pre-trained models: a **Convolutional Neural Network (CNN)** and a **Small Language Model (SLM)**.

The models employed in this system undergo the following preprocessing steps:

### A. Preprocessing Phase

- **CNN Model**: The pre-trained CNN model is used to detect ingredients within the refrigerator. It has been trained using a food detection dataset.
- **SLM Model**: This small language model has been pre-trained to recommend recipes based on the detected ingredients.

### B. Real-Time Scenario

In a real-world setting, the system operates as follows:

- **CNN Model**: Positioned inside the refrigerator, the CNN model scans and classifies the stored ingredients in real time.
- **SLM Model**: The detected ingredients are passed to the SLM in vector format, which then recommends recipes based on the ingredients.
- **User Interaction**: The user can interact with the SLM and receive real-time recipe information. For example, if ”kimchi” and ”salt” are detected, the system can suggest making kimchi stew.

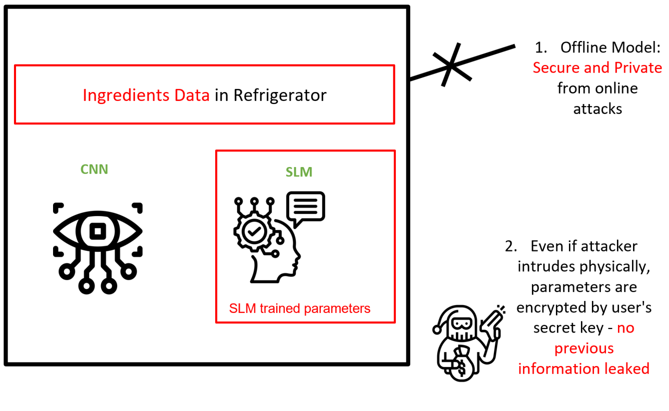

Ingredients Data in Refrigerator
CNN
SLM
SLM trained parameters
1. Offline Model: Secure and Private from online attacks
2. Even if attacker intrudes physically, parameters are encrypted by user's secret key - no previous information leaked

Fig. 2. Security and Privacy Aspect of Our Work

### C. Offline Processing and Security

One notable feature of this system is that all data processing occurs **offline**. The trained models are stored locally on the device, without any connection to the internet, and no data is transmitted externally. Therefore, unless physically compromised, the user's personal data and preferences remain securely protected.

## III. Convolutional Neural Network (CNN)

Our approach utilizes a pre-trained CNN model, specifically **ResNet-34**, which has been trained on the **ImageNet** dataset. For fine-tuning, we employed the **Grocery Store Dataset** to adapt the model to the specific task of food item classification. The fine-tuning process was completed in approximately **1 minute**, demonstrating the efficiency of the model in this context. Figure 3 illustrates the overall workflow of our image classification system, detailing both the pre-training and fine-tuning stages using the CNN.

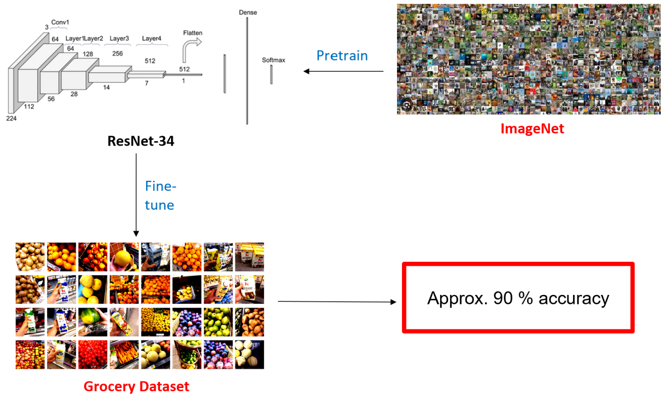


Fig. 3. Framework for CNN for Image Classification

### A. Dataset Selection

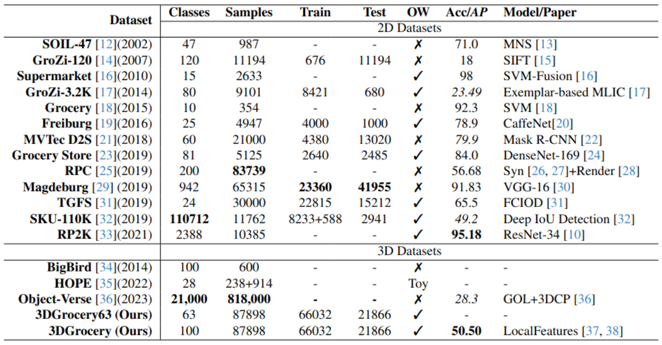
In this study, we utilized the **Grocery Store Dataset**, provided by Klasson et al. (2019) [5]. This dataset consists of 5,125 natural images taken with a smartphone camera in various grocery stores (see Fig. 4). The dataset includes 81 detailed classes, such as fruit, vegetables, juice, milk, and yogurt, which are further grouped into 42 broader categories. For example, detailed classes like ’Royal Gala’ and ’Granny Smith’ belong to the broader ’Apple’ category. The dataset is well-suited for fine-grained image classification tasks as it provides both detailed and generalized labels. The Grocery Store Dataset was first introduced in the paper *”A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels”* at WACV 2019.

Fig. 4. Grocery Store Dataset food item images

The reasons for choosing the Grocery Store Dataset are twofold. First, the dataset contains a wide variety of food items, aligning with the objective of this project, which is to develop a **food ingredient recognition system**. Second, the dataset consists of real-world images taken in a grocery store setting, providing diversity and realism.

### B. Model Selection

We referred to the work of Sheshappanavar et al. (2020) [6] to select a suitable model and dataset for food image classification. The ResNet-34 model achieved an accuracy of 95.18% on the **RP2K dataset**, as shown in Table 5. While the RP2K dataset is a large-scale retail product dataset, it contains images with Chinese labels, which makes it unsuitable for our study. Consequently, we selected the **Grocery Store Dataset**, where the DenseNet-169 model achieved 84% accuracy, making it a sufficiently robust dataset for food classification tasks.



| Dataset | Classes | Samples | Train | Test | OW | Acc/*AP* | Model/Paper |
|---|---|---|---|---|---|---|---|
| | | | | 2D Datasets | | | |
| **SOIL-47** [12](2002) | 47 | 987 | - | - | ✗ | 71.0 | MNS [13] |
| **GroZi-120** [14](2007) | 120 | 11194 | 676 | 11194 | ✗ | 18 | SIFT [15] |
| **Supermarket** [16](2010) | 15 | 2633 | - | - | ✓ | 98 | SVM-Fusion [16] |
| **GroZi-3.2K** [17](2014) | 80 | 9101 | 8421 | 680 | ✓ | *23.49* | Exemplar-based MLIC [17] |
| **Grocery** [18](2015) | 10 | 354 | - | - | ✗ | 92.3 | SVM [18] |
| **Freiburg** [19](2016) | 25 | 4947 | 4000 | 1000 | ✓ | 78.9 | CaffeNet[20] |
| **MVTec D2S** [21](2018) | 60 | 21000 | 4380 | 13020 | ✗ | *79.9* | Mask R-CNN [22] |
| **Grocery Store** [23](2019) | 81 | 5125 | 2640 | 2485 | ✓ | 84.0 | DenseNet-169 [24] |
| **RPC** [25](2019) | 200 | **83739** | - | - | ✗ | 56.68 | Syn [26, 27]+Render [28] |
| **Magdeburg** [29] (2019) | 942 | 65315 | **23360** | **41955** | ✗ | 91.83 | VGG-16 [30] |
| **TGFS** [31](2019) | 24 | 30000 | 22815 | 15212 | ✓ | 65.5 | FCIOD [31] |
| **SKU-110K** [32](2019) | **110712** | 11762 | 8233+588 | 2941 | ✓ | *49.2* | Deep IoU Detection [32] |
| **RP2K** [33](2021) | 2388 | 10385 | - | - | ✓ | **95.18** | ResNet-34 [10] |
| | | | | 3D Datasets | | | |
| **BigBird** [34](2014) | 100 | 600 | - | - | ✗ | - | - |
| **HOPE** [35](2022) | 28 | 238+914 | - | - | Toy | - | - |
| **Object-Verse** [36](2023) | **21,000** | **818,000** | **-** | **-** | ✗ | *28.3* | GOL+3DCP [36] |
| **3DGrocery63 (Ours)** | 63 | 87898 | 66032 | 21866 | ✓ | - | - |
| **3DGrocery (Ours)** | 100 | 87898 | 66032 | 21866 | ✓ | **50.50** | LocalFeatures [37, 38] |

Fig. 5. Comparison of food-related datasets and models ([6])

### C. ResNet-34 Architecture

The **ResNet-34** is a convolutional neural network (CNN) architecture composed of 34 layers. It is designed to address the *vanishing gradient problem* in deep networks by introducing *residual blocks*, which allow for stable learning even in very deep networks.

- **Convolutional Layers**: ResNet-34 begins with a 7x7 convolutional layer, followed by several 3x3 convolutional layers within residual blocks.
- **Residual Blocks**: These blocks consist of two or three 3x3 convolutional layers with batch normalization and ReLU activations. The input is passed directly to the output through an *identity mapping*, enabling the network to propagate information without loss even in deep networks.
- **Pooling and Classification Layers**: The final layers consist of max-pooling and fully connected layers for class prediction.

*1) Model Suitability:* The **ResNet-34** is a deep model, but thanks to *residual learning*, it can be trained efficiently with relatively fewer computational resources. This makes it well-suited for the **IoT devices** targeted in this study. Additionally, the ResNet-34 performs well in the classification tasks of the Grocery Store Dataset.

### D. Parameter Considerations and IoT Device Constraints

The ResNet-34 model contains approximately **21.8 million parameters**, each stored as a 4-byte floating point number, occupying approximately **87.2 MB of memory**. This memory requirement is manageable for mid-range IoT devices with **6GB of RAM and low-power CPU/GPU** capabilities.

Compared to deeper networks like ResNet-50 (23.6 million parameters) or ResNet-101 (44.5 million parameters), ResNet-34 offers a good balance between computational efficiency and model performance, making it suitable for deployment on memory- and resource-constrained IoT devices.

*1) Hardware Capacity and Model Compatibility:* Given the parameter size and memory footprint, the target IoT device with **6GB of RAM** is sufficient to handle ResNet-34 for inference and image processing without latency or resource bottlenecks. Optimization techniques such as *quantization* or *pruning* can further reduce the model size if necessary.

In summary, ResNet-34 strikes an ideal balance between *depth, accuracy, and efficiency*, making it a strong choice for building a real-time food classification system on edge devices. By leveraging pre-trained models and focusing on fine-tuning for specific food items, we can maintain high classification accuracy while adhering to the hardware constraints of IoT devices.

## IV. Small Language Model

TABLE I

IoT Device Classification Based on Hardware Constraints

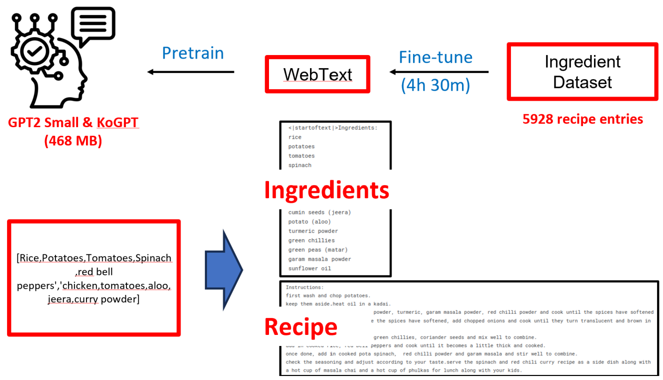
| **Category** | **Small** | **Medium** | **Large** |
|---|---|---|---|
| **Description** | Low-power, minimal processing | Moderate processing power, capable of handling localized tasks | High-performance devices with significant computational resources |
| **RAM** | Kilobytes to a few megabytes of RAM, low power | 1–8 GB RAM, edge computing capabilities | 8+ GB RAM, capable of complex tasks |
| **Examples** | Sensors, environmental monitors | NVIDIA Jetson, Raspberry Pi | Smart TV, laptop, edge servers |

Based on the IoT device classification outlined in Table I, we selected the appropriate Small Language Model (SLM) models for our system. The procedure for model selection, pre-training, and fine-tuning is illustrated in Figure 6. This process ensures that the models are well-suited for the computational and memory constraints of the IoT devices, while also providing efficient performance for real-time interactions.

The procedure is as follows: we initially selected the GPT-2 model, which was pre-trained on a large corpus of web text. Next, we fine-tuned the model using a recipe dataset containing 5,928 ingredient-recipe pairs. During testing, we provided vectors representing the detected ingredients and evaluated the model's ability to generate meaningful and relevant recipe outputs based on these inputs.

## V. Evaluation of CNN and SLM Models

**Hardware.** The experiments were conducted on a system equipped with a 13th Gen Intel Core i9-13900K processor



```
<|startoftext|>Ingredients:
rice
potatoes
tomatoes
spinach

cumin seeds (jeera)
potato (aloo)
turmeric powder
green chillies
green peas (matar)
garam masala powder
sunflower oil
```

```
Instructions:
first wash and chop potatoes.
keep them aside.heat oil in a kadai.
                         powder, turmeric, garam masala powder, red chilli powder and cook until the spices have softened
                        e the spices have softened, add chopped onions and cook until they turn translucent and brown in

                         green chillies, coriander seeds and mix well to combine.
                        peppers and cook until it becomes a little thick and cooked.
once done, add in cooked pota spinach,  red chilli powder and garam masala and stir well to combine.
check the seasoning and adjust according to your taste.serve the spinach and red chili curry recipe as a side dish along with
a hot cup of masala chai and a hot cup of phulkas for lunch along with your kids.
```

Fig. 6. Framework for SLM

(24 cores, 32 threads, up to 5.8 GHz max frequency) running Ubuntu 24.04 LTS. The system utilized the TFHE library version 1.1. For GPU-accelerated parallel processing, the system was equipped with an NVIDIA GeForce RTX 4060 Ti GPU (16 GB GDDR6 memory) running CUDA version 12.4.

## A. CNN

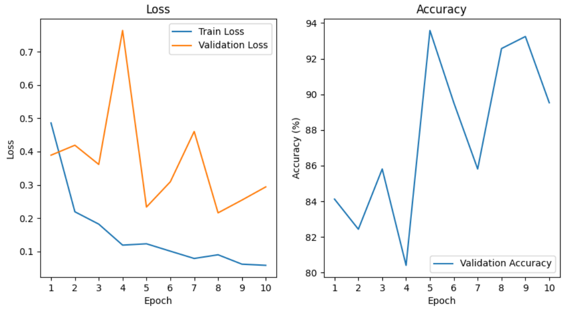


Fig. 7. Training and validation loss and accuracy trends.

The **ResNet-34** model was trained on the **Grocery Store Dataset** for a total of **10 epochs**. The **Adam optimizer** was used with a learning rate set to 0.001. As training progressed, the training loss steadily decreased, indicating successful learning from the data. However, the validation loss, after an initial decrease, fluctuated, as did the validation accuracy.

- **Training Loss**: The training loss consistently decreased from 0.485 in the first epoch to 0.058 by the final epoch, indicating that the model was well-fitted to the training data.
- **Validation Loss**: Validation loss decreased until the third epoch but began to fluctuate thereafter, spiking to 0.763 in the fourth epoch. This suggests signs of overfitting.
- **Validation Accuracy**: Validation accuracy increased from 84.12% in the first epoch to a peak of 93.58% by the fifth epoch, after which it fluctuated and declined in subsequent epochs.

As a result, it was determined that overfitting began after the fifth epoch. The fluctuation in validation loss and accuracy, coupled with the continuous improvement in training accuracy, indicated that the model was overfitting the training data and generalizing poorly to unseen data. To mitigate this issue, training was stopped after the fifth epoch, when the validation accuracy reached its peak (93.58%), and the model was saved for evaluation on the test dataset.

In conclusion, monitoring the variation in validation loss and accuracy throughout training proved crucial in preventing overfitting and ensuring optimal model performance. The model was saved after the fifth epoch, and further evaluations were conducted on the test dataset.

# VI. Discussion and Future Works

For effective object detection, the availability of a larger and more diverse dataset is crucial. Future work should focus on expanding the dataset by including a broader range of labels and corresponding images. This will enhance the model's ability to generalize across various food items and improve classification accuracy.

Additionally, exploring alternative models for the CNN component is necessary. For instance, employing models such as **YOLO (You Only Look Once)**, which is lightweight and optimized for real-time image classification, could significantly improve the system's performance in detecting ingredients efficiently in real-time.

Furthermore, for the SLM component, it is essential to evaluate the model's performance using a variety of metrics to better understand how well it performs in the task of ingredient-based recipe recommendation. This would ensure the model's suitability and effectiveness in generating meaningful and accurate recommendations.

# VII. Conclusion

In this project, we developed a privacy-preserving smart refrigerator system that integrates a CNN model for real-time ingredient detection and an SLM for personalized recipe recommendations. The system processes data locally, ensuring user privacy without relying on external servers. While the current implementation demonstrates promising results, future work will focus on expanding the dataset, exploring alternative models like YOLO for real-time classification, and refining the SLM's performance metrics to improve the system's overall efficiency and accuracy.

# References

[1] M. K. Morol, M. S. J. Rokon, I. B. Hasan, A. M. Saif, R. H. Khan, and S. S. Das, "Food recipe recommendation based on ingredients detection using deep learning," in *Proceedings of the 2nd International Conference on Computing Advancements*, 2022, pp. 191–198.

[2] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining Discriminative Components with Random Forests," in *European Conference on Computer Vision*, 2014.

[3] H. Mureșan and M. Oltean, "Fruits 360 Dataset," 2018. Available: https://www.kaggle.com/datasets/moltean/fruits

[4] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13*, Springer, 2015, pp. 3–17.

[5] M. Klasson, C. Zhang, and H. Kjellström, “A hierarchical grocery store image dataset with visual and semantic labels,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 491–500.

[6] S. V. Sheshappanavar, T. Anvekar, S. Kundargi, Y. Wang, and C. Kambhamettu, “A Benchmark Grocery Dataset of Realworld Point Clouds From Single View,” in *2024 International Conference on 3D Vision (3DV)*, 2024, pp. 516–527.

[7] Meta AI, “LLaMA: Open and Efficient Foundation Language Models,” 2023.

[8] Meta AI, “LLaMA 3.1: Architecture and Innovations,” 2023.

[9] Meta AI, “LLaMA Performance Benchmarks,” 2023.

[10] Meta AI, “Optimizing LLaMA for IoT Devices,” 2023.