# SmartChef: A Virtual Chef that substitutes menu components in images

Members: Rui Zhou, Jingmin Xu, Yuhang Chen, Kuanmin Wang
Github Repo: https://github.com/Bettyyy666/nutrivision/tree/main

## Introduction

To help small restaurants visualize customized food, we propose a deep learning-based system that enables automatic image editing. Customers can select different ingredients to include in their dishes and receive realistic images to help them visualize their decisions. In our design, the categories of dishes include burgers, sandwiches, tacos, and sides accompanying main dishes.

We decided to use Mask R-CNN to segment and classify dish components precisely. Then, we applied diffusion-based inpainting to change the ingredients in the image at high resolution, realistically. This solution enhances customized food visualization from the customer's side and reduces photography costs for restaurants. Our method would offer an efficient and scalable way to food visualization and ultimately empower small-scale restaurant businesses.

## Methodology

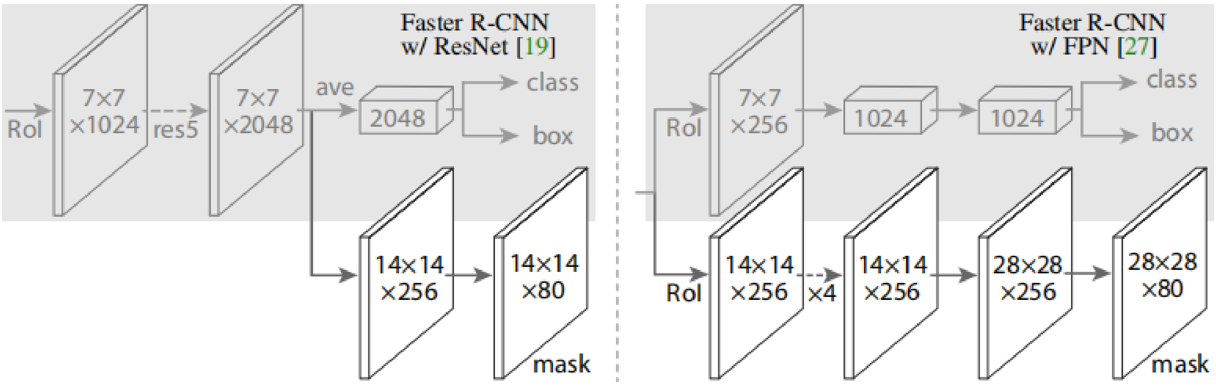
Our system uses a two-step approach to automatically detect and edit food images. First, we use Mask R-CNN to segment individual food ingredients from each image. Mask R-CNN is fine-tuned based on the FoodSeg103 dataset that provides the segmentation of each food ingredient in the image. This training helps the model accurately identify and separate different food components within complex images.

Mask R-CNN: Mask R-CNN is the model that backboned on the Faster R-CNN. The whole development process from R-CNN is below.

| | From R-CNN to Fast R-CNN | From Fast R-CNN to Faster R-CNN | From Faster R-CNN to Mask R-CNN |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Improvement | Instead of searching on each region in the image, Fast R-CNN runs a CNN model to generate the feature map for the whole image. The later block in Fast R-CNN includes ROI pooling to help with multi-object classification. | Faster R-CNN introduces a Region Proposal Network to solve the computing speed problem of Selective Search in Fast R-CNN. Compared to Fast R-CNN, Faster R-CNN enables much faster computing speed, which allows use in real-time monitoring. | Add-on to Faster R-CNN, Mask R-CNN adds a Fully Convolutional Network (FCN) parallel to the classification and box regression in Faster R-CNN. FCN allows pixel-to-pixel detection that Faster R-CNN can't do due to ROI pooling being more high-scale looking. |

**Architecture of Mask R-CNN**

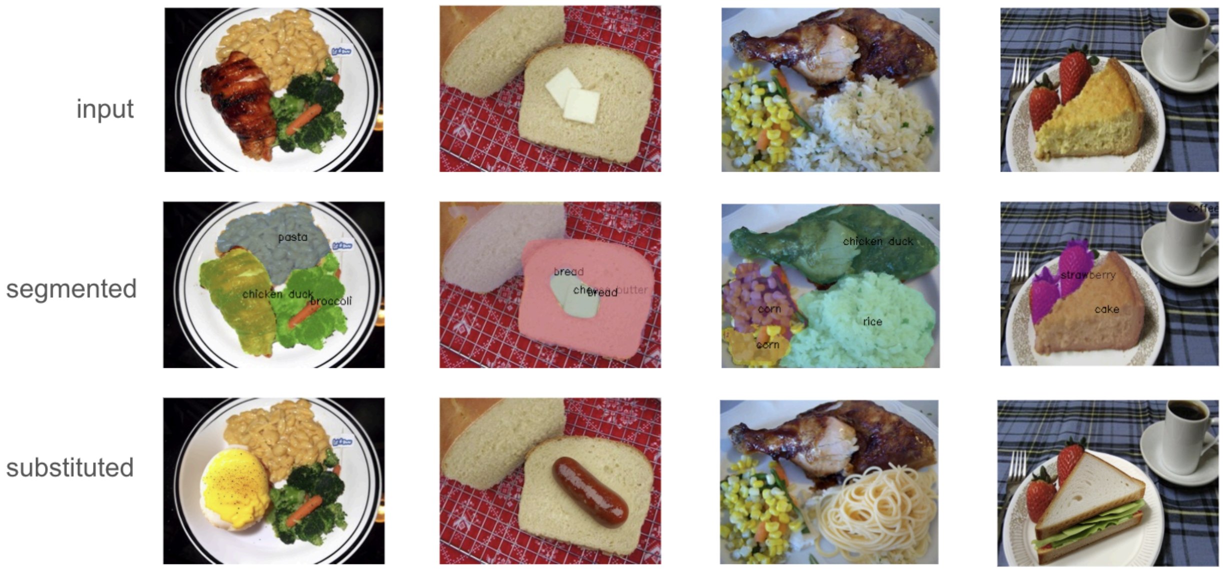


In the second step, we apply a diffusion-based inpainting model (DreamShaper v8) to realistically replace specific food items within the image. Once Mask R-CNN identifies and masks out the target ingredient, the diffusion model generates a new item and replaces the target based on user-defined prompts, such as "replace hamburger with salad."

For training, the segmentation module (Mask R-CNN) was fine-tuned specifically on the FoodSeg103 dataset to achieve accurate identification of food components. For the replacement step, we did not retrain the diffusion model itself. Instead, we focused on careful and iterative prompt engineering to ensure effective and natural-looking replacements.

Due to the nature of our system, we utilize Mask R-CNN with ResNet. Compared to FPN, which is designed for detecting small objects or objects with varied sizes in images, ResNet is more suitable for our case, since our dataset is relatively stable in object size. During the fine-tuning process, we adjusted batch size, learning rate, anchor size and ratio, and the learning rate scheduler to optimize our model's performance.

## Results



Chicken→scrambled egg    Cheese butter → sausage    Rice → pasta    cake → sandwich

***Mask R-CNN:***

| | IoU | mAP50 | mAP70 |
|---|---|---|---|
| MRCNN | 0.0455 | 0.0000 | 0.0000 |
| Our model | 0.5374 | 0.4105 | 0.2826 |

For our Mask R-CNN model, we use three metrics(IoU, mAP50, and mAP70)to evaluate its performance. In terms of all three metrics, our model demonstrates better performance than the original Mask R-CNN model. The poor performance of the original model supports our assumption that food segmentation is a more challenging task for general-purpose models due to the variability in shape, color, texture, and other visual characteristics.

***Diffusion Inpainting Model:***

We evaluate our diffusion inpainting model through manual inspection. We use a binary method to assess whether the generated output is plausible. Our evaluation criteria include content accuracy, edge consistency, color matching, texture realism, and the completeness of the replaced item. Based on this assessment, 70.67% of images passed our human plausibility test.

**Challenges:**

***Feasibility of the Topic and Pivoting***

The first challenge was selecting the topic for our project. Although we explored several inspiring ideas, many of them were determined as not viable due to the complexity of the models or the high computational power. Our original idea was to build an AI-powered virtual try-on system, but we quickly found out that it would exceed our available computing resources. As a result, we decided to pivot to our current project, which offered a more manageable balance between technical feasibility and project scope.

***Limited Computational Resources and Time***

After prompting our topic, we continued to face difficulties during the fine-tuning process. The most significant challenge was to handle different hyperparameters with limited computational resources. While Mask R-CNN is relatively efficient compared to other models, it still requires considerable time and memory to train. To address this, we used both local machines and cloud computing resources in parallel to accelerate the finetuning process.

***Dataset Selection***

Our initial dataset was the UNIMIB2016 Food Database, which contains 1,027 tray images of multiple food items. However, all the images share a similar layout. We believe it limited the model's ability to generalize. In fact, we observed that models that are fine-tuned based on this dataset are struggling to segment food if the images have different style and structure. Consequently, we decided to switch to FoodSeg103, a more complex and varied dataset. Finding a high-quality food segmentation dataset was a challenge. Segmentation dataset is relatively small since the label task requires an extensive manual. The food segmentation dataset further narrows down the available dataset. We spent significant time finding a dataset that met our requirements.

***Determining Hyperparameter Combinations***

Another key challenge was selecting the right combination of hyperparameters for fine-tuning. To speed up the whole process, we initially limited training to 3 epochs to filter out bad solutions. We understand this could increase our false negative errors, but we believe it is an understandable compromise. While the trail testing helped reduce iteration time, we found that many combinations produced similar results. It makes it hard to figure which combination could perform better.. We had to document each value for comparing results later. In hindsight, having a stronger understanding of the underlying logic behind each parameter's impact could have made this process more efficient.

## Reflection

Overall, we’re proud of how our project turned out. Our system successfully demonstrated how food images can be automatically edited using a combination of segmentation and generative methods. We met our base and target goals: we were able to fine-tune Mask R-CNN to detect food components from real images and apply a pre-trained diffusion model to make meaningful and visually realistic changes. While the results weren’t perfect, they showed the potential of this two-step pipeline for real-world applications.

Our model worked largely as expected, especially the segmentation part after fine-tuning on the FoodSeg103 dataset. Mask R-CNN did a good job in identifying most ingredients, although it sometimes struggled with overlapping items or small, similar-looking components. The inpainting model was less predictable — it occasionally generated replacements that didn’t fully match the surrounding image or were visually off. Still, with careful prompt tuning, we were able to guide it toward producing more realistic results.

As the project progressed, our approach evolved. Originally, we considered training an end-to-end system from scratch, but we quickly realized that fine-tuning pre-trained models and focusing on integration would be more practical. We also shifted datasets from UNIMIB2016 to FoodSeg103 for better annotation quality and broader category coverage. If we could start over, we might spend more time on choosing a stronger segmentation model up front and exploring additional guided editing tools like ControlNet earlier in the process.

Given more time, we would work on improving the consistency and realism of the inpainted images. This could include refining segmentation masks further, improving prompt engineering, and exploring more advanced models that allow greater control over the output. We'd also want to make the system faster and more user-friendly, potentially for real-time or mobile applications.

The biggest takeaway from this project was learning how to combine different deep learning components into a single workflow. We got hands-on experience fine-tuning vision models for a niche domain and working with generative models that depend heavily on well-phrased prompts. We also learned how small changes in model parameters, prompt wording, or input formatting could lead to very different results. Beyond technical skills, we gained a better understanding of the challenges involved in building systems that need to be both flexible and user-friendly.

## Reference

1.He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. arXiv preprint arXiv:1703.06870. https://arxiv.org/abs/1703.06870 ￼

2.Wu, X., Fu, X., Liu, Y., Lim, E.-P., Hoi, S. C. H., & Sun, Q. (2021). A Large-Scale Benchmark for Food Image Segmentation. arXiv preprint arXiv:2105.05409. https://arxiv.org/abs/2105.05409 ￼

3.Lykon. (n.d.). Dreamshaper 8 Inpainting [Computer software]. Hugging Face. https://huggingface.co/Lykon/dreamshaper-8-inpainting