Finish Your Peas! Utilizing Multi-Label ImageClassification to Identify Food Items and Ingredients for Recipe Suggestions and Reducing Food Waste

Stanford CS224N {Custom} Project

Arianna Damiani
Department of Computer Science
Stanford University
adamiani@stanford.edu

Prashaant Ranganathan
Hasso Plattner Institute of Design
Stanford University
prashran@stanford.edu

Abstract

Consumers and retailers waste over a 100 billion pounds of food a year in the United States. Our goal is to develop a neural model that identifies ingredients from images and suggests recipes to minimize consumer food waste and improve nutrition. This project extends C-Tran, a multi-label classification convolutional neural network combined with transformer layers. We trained and fine-tuned a C-Tran model to the task of identifying 100 classes of raw ingredients from images. Analysis showed high mean average precision scores of 89.2 with comparable overall F1, recall, and precision scores. We then took labels generated from images and fine-tuned a GPT-2 model to write novel recipes. We evaluated with ROUGE scoring. The model displayed high performance when integrating common and widely used ingredients but struggled when attempting to incorporate or accurately describe preparation of specialty ingredients.

1 Key Information to include

- TA mentor: Shijia
- External collaborators: No
- External mentor: No
- Sharing project: No
- Arianna's contributions: Pre-proccessing of food images to generate accurate labels for recipes. Modification of C-Tran code, training, and evaluation of computer vision model. Writing and editing of final report.
- Prashaant's contributions: Building the recipe generating model from food ingredients to actual recipes with corresponding evaluation and analysis. Writing and editing of final report. Making the poster for the final presentation.

2 Introduction

Food waste is a pressing global issue, with approximately one-third of all food produced for human consumption being wasted. This leads to significant environmental, social, and economic repercussions. Included are the wastage of resources like water, land, and energy which contributes substantially to the generation of greenhouse gases emission. Addressing food waster requires innovative solutions that can effectively utilize leftover ingredients to create nutritious meals.

One promising approach is utilizing technologies to help individuals and households make better use of available food resources. Most techniques for recipe generation, however, focus on making recipes from images of already completed dishes. These approaches only take into consideration the preferences of the user rather than prioritize factors such as ingredient availability. Here we implement a method of generating recipes from images of raw ingredients.

Our approach involves two main steps. First a computer vision model, combining convolutional neural networks and transformer layers, identifies the ingredients within an image via multi-label classification. Next a fine-tuned GPT-2 model is used to generate recipes from the provided list of available resources. This method aims to bridge the gap between finding ingredients to recipe generation for users, enabling them to more sustainably utilize food resources.

3 Related Work

Our project builds upon existing work in image recognition, multi-label classification, and recipe generation using advanced neural network models.

3.1 Multi-Label Image Classification

Jack Lanchantin et al. Jack Lanchantin (2020) developed an improved multi-label image classification model using transformers, known as C-Tran. This model enhances the capability of computer vision systems to process images with multiple features, making it suitable for identifying various ingredients within a single image.

3.2 Recipe Generation from Images

Several studies have explored the generation of recipes directly from food images:

- **Inverse Cooking:** A. Salvador et al. A. Salvador (2019) demonstrated how deep learning models can generate recipes directly from food images using a dual network architecture.
- **Monte Carlo Tree Search with GPT-2:** Karan Taneja et al. Karan Taneja (2023) integrated Monte Carlo Tree Search with GPT-2 to optimize recipe generation, resulting in higher quality and more diverse recipe suggestions.
- **RecipeGPT:** Helena H. Lee et al. Helena H. Lee (2020) presented RecipeGPT, a system that uses a fine-tuned GPT-2 model to generate and evaluate recipes based on given titles, ingredients, and instructions.

4 Approach

To restrict the number of ingredients for multi-label classification, we began with identifying the top 100 used ingredients in "Food Ingredients and Recipes Dataset with Images" dataset available through Kaggle (foo). Recipe ingredients were tokenized to contain only the ingredient name with quantity and preparation instructions stripped. This leads to a loss of detail in the size and special formatting of ingredients needed. Further it assumes that whatever quantity is identified in the image should be sufficient to cook a variety of recipes. Additional specifiers such as "grounded", "granulated", and "minced" were also removed to simplify the classes the model would need to predict across. For instance, "red onion", "yellow onion", and "green onion" were all simplified to "onion" as they are commonly made substitutions for each other.

The first aspect of our model regards how we can generate multiple labels for images containing an unknown number of ingredients. We re-implemented the architecture already developed by Lanchantin et al., Classification Transformer (C-Tran)(Jack Lanchantin, 2020). Their method involves using ResNet-101 trained on ImageNet to generate image embeddings. Next a transformer encoder is utilized to infer on the image feature embeddings and modified label embeddings when present. This process is repeated for three layers until a final set of embeddings is generated. The decoder then makes predictions across a finite set of predefined labels. Losses are calculated via cross-entropy loss.

We began with the same combination of ResNet-101 to calculate image embeddings and use the same three transformer layers but trained on VOC20 dataset. This is done on a different size set of labels than our desired label set size. We therefore subsequently alter the decoder and fine-tune with our ingredients dataset, labelling the 100 most frequently used in ingredients. Our baseline of comparison is a ResNet-101 model pretrained on ImageNet and then fine-tuned to our dataset (res).

Given the complexity and importance of grammatical structure and order in recipes, we utilized GPT-2, a pre-trained language model, to handle the language generation tasks. Training a model from scratch for such tasks would be computationally intensive and time-consuming. We fine-tuned GPT-2 on the RecipeNLG dataset to specialize it for recipe generation. The fine-tuning process adjusts the model's parameters to better align with the specific characteristics and requirements of recipe data.

5 Experiments

5.1 Data

To train the image classifier we used the "Food Ingredients and Recipes Dataset with Images" dataset composed of images, ingredient lists, and recipes (foo). Pre-processing steps involved removal of common adjectives such as "fresh", size descriptors such as "large", preparation descriptors such as "diced", and quantity descriptors such as "1/2 cup" from all ingredients listed. The top 100 most frequent ingredients were then extracted from all recipes in the dataset. Images were then labelled in binary fashion based on absence/presence of a top 100 ingredient. In total there are 13,501 labelled images with a $70:30$ split into training and validation sets. Image pre-processing included commonly applied transformations such as cropping, sharpening, and brightening.

We used the RecipeNLG dataset for our evaluations of recipe quality, which is available on Kaggle (Mooney, 2020). This dataset provides a rich source of recipes, including both ingredient lists and instructions. We used this dataset To thoroughly test and benchmark our model's performance in generating recipes from identified ingredients. In total there are over 2 million recipe in the dataset, however, only 1 million were randomly selected for training and validation in a $90:10$ split.

5.2 Evaluation method

To evaluate the generation of ingredients by models, we computed standard F1 scores, recall, precision, mean average precision (mAP), and accuracy between the set of generated ingredients and the ground-truth ingredients. These measures should evaluate how well ingredients are recognized as well as give insights into whether the model is biased in over or under predicted the presence of certain ingredients.

For evaluating the generated instructions, we used a n-gram overlap-based metrics: ROUGE. Additionally, we compute the normalized Cosine Similarity (as a gauge of uniqueness of the recipe), tree edit distance (NTED) to measure the overall quality of the generated instruction texts compared to the ground-truth instructions. These metrics provide a comprehensive assessment of how well the generated instructions align with existing recipes.

5.3 Experimental details

For our baseline, ResNet101 pre-trained model on ImageNet 1k was provided through HuggingFace. Further training was conducted on GCP with 2 T4 GPUs and a learning rate of 1E-4, ADAM optimizer, batch size of 16, and a max epoch of 100. Label classification was done across top 100 recipe ingredients with cross entropy loss. Pre-training of C-Tran was done with VOC20 dataset on GCP with 2 T4 GPUs at a learning rate of 1E-4, ADAM optimizer, batch size of 16, and a max epoch of 150. Classification task was across 20 labels with cross entropy loss used for updates. Fine-tuning was done freezing all layers before the decoder, dubbed "Frozen C-Food", and without freezing, dubbed "Pre-C-Food", at a learning rate of 1E-4, ADAM optimizer, batch size of 16, max epoch of 150, and cross entropy loss for classification across 100 labels.

The finetuning of the GPT-2 model involved setting parameters to optimize the training process with evaluations occurring every 5000 steps. Logging was set to occur every 2500 steps. The learning rate was set at 3×10^{-5}, and both training and evaluation batch sizes were set to 32. The model was trained for a total of 3 epochs. Checkpoints were saved every 5000 steps, with a maximum of 2 checkpoints retained. Additionally, mixed precision training (fp16) was enabled to improve memory efficiency and computational speed.

5.4 Results

The evaluation metrics for the image classifier are detailed in Table 1. Of note the baseline ResNet101 model performed worse in all metrics compared to C-Tran based variants. Pre-training and fine tuning over all layers led to a very good performance with a mAP of 89.2 and high F1 score of 85.9.

Ingredient Identification Task					
Model	Loss	mAP (%)	F1 (%)	Precision (%)	Recall (%)
ResNet101	20.592	20.6	39.4	62.1	28.8
C-Food	19.664	26.0	44.5	63.8	34.2
Frozen C-Food	11.8	60.7	75.3	46.1	57.2
Pre-C-Food	1.158	89.2	85.9	88.5	83.5

Table 1: Metric evaluation of performance of models. Baseline ResNet101 model in row 1. C-Tran model with no pre-training dubbed "C-Food" in row 2. C-Tran model with pretraining on VOC20 dataset and then fine-tuning only at the decoder level in row 3. C-Tran model with pretraining on VOC20 dataset and then fine-tuning on all levels in row 4.

The average ROUGE-1 F1 score of 0.32, ROUGE-2 F1 score of 0.12, and ROUGE-L F1 score of 0.30 indicate a moderate overlap with expected recipes, with better performance on simpler recipes. The average NTED score of 285 reflects good structural similarity, and an average Cosine Similarity score of 0.35 suggests reasonable semantic alignment.

6 Analysis

The pretrained and fine-tuned C-Tran model, Pre-C-Food, expectedly performed better than its counterparts. It is a more expressive form of the baseline model ResNet101 owing to its additional transformer layers. An ablation study would describe better how each layer improves the performance but the stark increase in performance speaks for itself. This model had slightly higher precision than recall suggesting that when ingredients are identified they are likely to be correctly identified but that not all items will be identified and therefore incorporated in the recipe. The high precision is relevant here as the method should focus on suggesting users recipes primarily based on the ingredients they have available rather than promote purchasing of more ingredients. The lower recall may also be a symptom caused by low prevalence in the dataset. Some items such as "salt" appear in 66.7% of recipes while others like "pistachio" are only in 18.5%. This presents an issue when considering that specialty ingredients make a dish unique and should still be applicable to this method. This effect is further amplified when considering that the GPT-2 model excels at generating recipes with simple and common ingredients.

The GPT-2 model's proficiency with simple and common ingredients is similarly likely to be due to the prevalence of such ingredients in the training dataset. High prevalence in this dataset allows the model to learn their usage patterns more effectively. When recipes involve fewer and less complex ingredients, the generated instructions tend to closely match the expected recipes. This indicates that the model can generate accurate and coherent instructions when the ingredient list is short and commonly used items. However, the performance drops with more complex recipes that require detailed instructions and involve less common ingredients. This suggests that the model's ability to handle intricate cooking processes and specialized ingredients is limited. Enhancing the dataset with a wider variety of complex recipes and ingredients could improve the model's ability to generate accurate and detailed instructions for more complex dishes.

7 Conclusion

Our project addresses the issue of food waste by generating recipes from images of raw ingredients. The dual-step approach combines a computer vision model, utilizing convolutional neural networks and transformer layers for multi-label ingredient classification, with a fine-tuned GPT-2 model for recipe generation. The usage of pre-training and fine-tuning of C-Tran model led to good performance in the task of raw ingredient labelling with mAP of 89.2. This marks significant improvement over baseline convolutional networks such as ResNet101 and demonstrates the effectiveness of transformer models in computer vision tasks. Evaluation of the GPT-2 model showed promising results for generating recipes with simple and common ingredients, as indicated by moderate ROUGE scores (average ROUGE-1 F1 score of 0.32), structural similarity (average NTED score of 285), and reasonable semantic alignment (average Cosine Similarity score of 0.35). However, the model struggles with complex recipes that require detailed instructions and multiple steps.

Limitations of the implementation evaluated include data and computing. Although there are many established datasets for completed recipe dishes there are few available for raw ingredients. Additionally, transformer and GPT-2 models require extensive computational resources in order to train due to their high expressivity and complex structures. Ablation studies would aid in understanding which layers are most relevant for transformers in computer vision tasks. To date there are also more and more convolutional neural networks being trained to generate image embeddings which may perform better than ResNet101.

Future work will focus on enhancing the model's performance by incorporating user preferences and dietary restrictions, expanding the training dataset with more diverse and complex recipes, and integrating user feedback. These improvements aim to create a more robust and versatile tool for reducing food waste and promoting sustainable consumption practices. By bridging the gap between ingredient identification and recipe generation, this project contributes to more efficient and environmentally friendly use of food resources.

8 Ethics Statement

Our recipe generation model to address food waste presents several ethical challenges and possible societal risks. Here are the key issues and our mitigation strategies:

8.1 Privacy and Data Security

One of the primary ethical challenges is ensuring the privacy and security of user data. Users upload images of ingredients, which may inadvertently include sensitive information. The risk of unauthorized access and data breaches poses significant threats to user privacy. To mitigate this, we anonymize all user data and store it securely, complying with international data protection regulations such as GDPR. We implement robust security measures to prevent unauthorized access and data breaches, ensuring that user information remains confidential.

8.2 Bias and Fairness

There is a risk of the model reflecting biases present in the training data, which could lead to unfair or culturally insensitive recipe suggestions. This can alienate users from diverse backgrounds and perpetuate existing inequalities. To mitigate this risk, we train our model on diverse datasets that encompass a wide range of culinary traditions and dietary preferences. This approach helps minimize cultural and dietary biases, ensuring that our tool is accessible and fair to all users. Continuous monitoring and evaluation help identify and address any potential biases that may arise.

8.3 Sustainability

While our project aims to reduce food waste, there is a risk that the model's suggestions may not always align with sustainable practices. For example, the recipes generated could inadvertently encourage the use of ingredients that have a high environmental impact. To mitigate this risk, we plan to incorporate sustainability metrics into our model, prioritizing recipes that use locally sourced and low-impact ingredients. By promoting the efficient use of food resources and reducing waste, our project contributes to broader environmental sustainability goals. We also aim to raise awareness about the impact of food waste on the environment and encourage sustainable consumption practices.

8.4 Health and Nutritional Balance

Another ethical challenge is ensuring that the recipes generated by the model are nutritionally balanced and promote healthy eating habits. There is a risk that some recipes may lack essential nutrients or encourage unhealthy eating. To address this, we incorporate nutritional information into our model, allowing it to generate recipes that are not only tasty but also nutritionally balanced. Users can customize their dietary preferences and restrictions, ensuring that the generated recipes meet their specific health needs. This approach promotes better health outcomes and supports users in making healthier food choices.

References

Food ingredients and recipe dataset with images. https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images. Accessed: 2024-05.

Hugging face, resnet-50. https://huggingface.co/microsoft/resnet-50. Accessed: 2024-05.

Y. Aytar J. Marin F. Ofli I. Weber A. Torralba A. Salvador, N. Hynes. 2019. Inverse cooking: Recipe generation from food images. arXiv:1812.06164.

Palakorn Achananuparp Philips Kokoh Prasetyo Yue Liu Ee-Peng Lim Lav R. Varshney Helena H. Lee, Ke Shu. 2020. Recipegpt: Generative pre-training based cooking recipe generation and evaluation system. https://arxiv.org/abs/2003.02498.

Vicente Ordonez Yanjun Qi Jack Lanchantin, Tianlu Wang. 2020. General multi-label image classification with transformers. arXiv:2011.14027.

Richard Goodwin Karan Taneja, Richard Segal. 2023. Monte carlo tree search for recipe generation using gpt-2. arXiv:2401.05199v1.

Paul Timothy Mooney. 2020. Recipenlg: A large-scale, high-quality dataset for recipe generation. Accessed: 2024-05-23.