Food Recipe Retrieval: Using CLIP Prompts to Query Food Recipes

Christopher Le, Matheus Rempel
Department of Computer Science, Rice University, Houston, TX 77005
[ctl7, mmr13]@rice.edu

Abstract

Determining the recipes of foods given an image has been a common deep learning task in the past decade. With freely accessible datasets such as Food-101[1] and Recipe1M+[4] being released for public use, several attempts have been made to build a deep learning model to predict the recipes of food items given a picture of the food. However, these models vary significantly in accuracy in terms of ingredients and cooking steps, and the results are not verifiable. That is, even if the steps given can produce a food that matches an image, the taste of the food is unpredictable because they are generated through a model. In our project, we explore the capabilities of CLIP[5] for highly-accurate food classification over common image classifier models such as ResNet and InceptionV3. We then propose a recipe retrieval pipeline using CLIP and data mining techniques to output relevant recipes for food images directly from verified recipe websites. This way, every recipe returned by our model is verifiable and guaranteed to produce good-tasting food.

1. Introduction

Image classification is the process of categorizing images based on the pixels by which they are made of. This process, for humans is often rather trivial; however, for a computer this can be a challenging task. The concept of image classification has been growing significantly over the past decade along with its prospective field of AI and Deep Learning. Fundamentally, the process by which one can program a computer is relatively trivial, although the implementations are often complex in practice.

Many implementations of image classification have focused on detecting objects which show minimal to no variation between them. For example, by the autonomous car industry, to classify road markings, brake lights, or traffic signs while cars drive without human input. The task of image classification increases significantly in difficulty when the object attempting to be recognized is not constant in shape, color, and detail when compared to others within the same class. One category which is notorious for the aforementioned problem is food. Not only is it extremely challenging for humans to replicate the visual appearance of a dish from a recipe, but as is often the case, humans like to create their own rendition of the food in question-inevitably leading to significant visual representations.

Often when people go out to eat, many choose to take pictures of their food before eating it. This practice has become so popular in fact, that the social media industry has coined the phrase "the camera eats first" as a manner to emphasize the practices' importance to many people. While these pictures are generally taken with the intended use of uploading to social media in order to provide updates or gain "likes", they have the ulterior benefit of acting a memory and snapshot of the pre-consumed meal. Hence, if one were to scroll through their camera roll, and see a picture of a previously eaten food, they may want to eat said dish again but not recall the name of the dish or restaurant.

In this paper, we will go over methods for taking an image of food as an input and returning its potential recipe. We will explain the pros and cons of such methods and finally propose our own method which utilizes the well-known and effective CLIP neural network model in conjunction with info retrieval to produce recipes.

1.1. Dataset

To achieve this goal, the first step is to obtain a dataset. For this application, an ideal dataset would have multiple images representing various categories of dishes/meals. Fortunately, there is an appropriately named public-use dataset which has exactly this – Food 101[1]. This dataset contains 101 different food items (i.e., Apple Pie, Pizza, etc.), each with 1,000 images. These images have large variance in quality, lighting, and background and contents. Figure 1 contains sample images from the dataset, but the variances can be clearly seen with the human eye.

[1] https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/

Figure 1. Here we include sample images from the first 4 classes in the Food101 dataset: apple pie, baby back ribs, baklava, and beef carpaccio

2. Related Work

Neither Food Classification nor Recipe Retrieval are new concepts, and many attempts have been made to improve the accuracy of previous models. Many of these implementations, including ours, make use of pre-trained neural networks which are either directly applicable to the necessary implementation or are fine-tuned accordingly.

2.1. Food Classification

One approach [7] was attempted in 2016 using the GoogLeNet CNN, a 22 layer neural network built in a manner akin to that of the famous ImageNet using the Place365 dataset. From there, the paper uses the datasets called food-5k and food-11 to train the model. This paper will follow a similar approach to that proposed in the GoogLeNet publication; however, in that approach, there was the added element of distinguishing food from non-food items. Then, only if the image was classified as "food", would it attempt to categorize the food into its correct label. One of the biggest differences in this model is the sheer scale of the dataset. When the two aforementioned datasets are combined, the final dataset will contain 2500 pictures of food and 2500 of non-food items. Moreover, it has 16,000 labeled images of foods split into 11 different categories. While GoogLeNet improved the accuracy from the AlexNet, VCG, and ResNet; the overall 83.6% accuracy seems to have room for improvement.

Cornell University has also attempted to improve this result with a paper [3] published in 2019 with a custom-built CNN. The training dataset used was the same as that used for ImageNet. In this paper, the researchers implement an SGD algorithm (stochastic gradient descent). Moreover, the neural network consists of 3 convolutional layers, along with a ReLU activation function and SoftMax loss function. This CNN allowed for an improved accuracy of 87%, with the reported issue being the size of the dataset being not being large enough. This claim is backed up with data representing only 10 food categories each with anywhere from 40 to 350 images.

2.2. Recipe Retrieval

Possibly the most leading work [6] in the topic of recipe retrieval is im2recipe, which is a multi-modal neural network trained on a collection of images and associated recipes called Recipe1M[2]. This model learns both image and text embeddings to predict entire recipes (ingredients and steps to make the food). While this model is trained on an extensive dataset, we opted not to use this model as a basis for our experimentation. This is due to our goal of returning rigidly defined recipes to the user instead of attempting to predict the entire recipe given an image. We figure that the best way to accomplish this task is through retrieving predefined recipes through the web.

3. Model

Once the dataset is obtained; the next step is to setup the images/text for use in Machine Learning algorithms. This step involves encoding the images and text- a process which effectively converts the images and captions into various arrays such that every pixel or word is represented numerically. A neural network would then need to be built to train a model. Image classification usually involves either CNNs (Convolutional Neural Networks) or RNNs (Recurrent Neural Networks) due to their general high accuracy combined with efficiency in training a model. While it would be possible to train a model from scratch, due to the size, complexity, and extreme variance of the dataset, the option of adapting and tweaking a pre-existing model is more typically more favorable. However, even finetuning pre-existing models can be time and resource intensive. This paper will leverage the zero-shot capabilities of the neural network proposed by OpenAI: CLIP[5] (Contrastive Language–Image Pre-training) to perform accurate food classification without needing to spend hours finetuning a model.

CLIP is a neural network that has been pre-trained on a variety of image-label pairs Ultimately, CLIP is an image encoder and a text encoder to predict which images were paired with which labels in OpenAI's dataset; a behavior turning CLIP into a zero-shot classifier[3]. Figure 2 shows a visual representation of the CLIP encoder mechanism. We will use the CLIP model first for food classification. That is, when supplied an image of a food item and possible labels, the model will predict the associated label.

[2]http://pic2recipe.csail.mit.edu/
[3]https://openai.com/blog/clip/

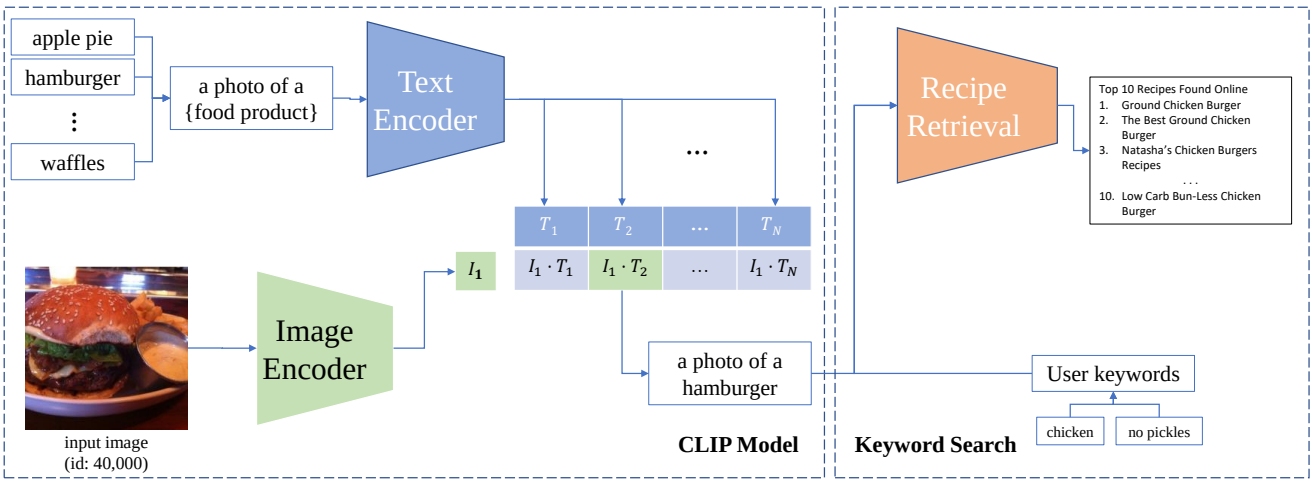

Figure 2. Recipe Retrieval Pipeline using CLIP

3.1. Recipe Retrieval Algorithm

Previous works in recipe retrieval make use of datasets which contain images and their associated images [4] and train models to directly produce recipes using image captioning techniques for ingredients and cooking instructions. We propose a more direct addition to the recipe retrieval pipeline (also shown in Figure 2) which incorporates an info retrieval system that will take the output of the CLIP model and a number of keywords (i.e. vegan, organic, with cheese, etc.) from the user. The system will use a data mining algorithm to scrape websites using the search query to return several recipes related to the food item. For instance, if the user supplies a picture of a plate of unknown ravioli and keywords "pesto" and "beef," the CLIP model will likely classify this food item as "ravioli." Then the recipe retrieval system will query the search term "beef ravioli with pesto" or a syntactically similar query and return the top search results.

For the purposes of this implementation, we obtain our results using allrecipes [4] because of the simplicity and consistency of its search bar results.

Model	Test Acc. @ 1	Test Acc. @ 5
RF [1]	50.76%	XX%
InceptionV3 [8]	87.31	XX
ResNet-50 [2]	89.09	98.11
CLIP (ViT B/32)	82.05	96.87
CLIP (ViT L/14)	92.04	99.19

Table 1. Preliminary experimental results (Top 1 and Top 5 accuracy) on food type classification on Food101 dataset.

[4]https://www.allrecipes.com/

4. Experiments and Results

To measure the efficacy of the classification model (CLIP), we partitioned the dataset into training and testing sets with 75%-25% split respectively - 750/250 per category. While CLIP does not require any fine-tuning because it is already trained as a zero-shot classifier, we must keep the partitions consistent with the baseline models. The Food-101 dataset contains preset train/test partitions, so every trainable model was trained on the same data. Likewise, each model is tested on the same $250 * 101$ data points.

4.1. Baselines

We judge CLIP's effectiveness for food classification by comparing its overall accuracy with similar classification models that are trained to perform the same task. The first baseline is presented in the paper [1] that introduces the Food-101 dataset. This model (RFDC) is trained using random forest classification and serves as a preliminary work to test the Food-101 dataset.

The two additional baselines [8][2] shown in Table 1 are food classification models built using popular pretrained models (InceptionV3 and ResNet-50 respectively) and fine-tuned using the Food-101 dataset. Both of these models were initially trained on the ImageNet dataset and require ample amounts of transfer learning time[5] to be used with the Food-101 dataset and implemented several forms of data augmentation techniques such as Mixup[9], rotations, flips, perspective warping, etc, to improve accuracy.

4.2. Classification Results and Discussion

As shown in Table 1, the first available CLIP model (ViT B/32) which is trained with Vision Transformer architecture

[5]According to the github repos [2] [8], these models took upwards of an hour to train

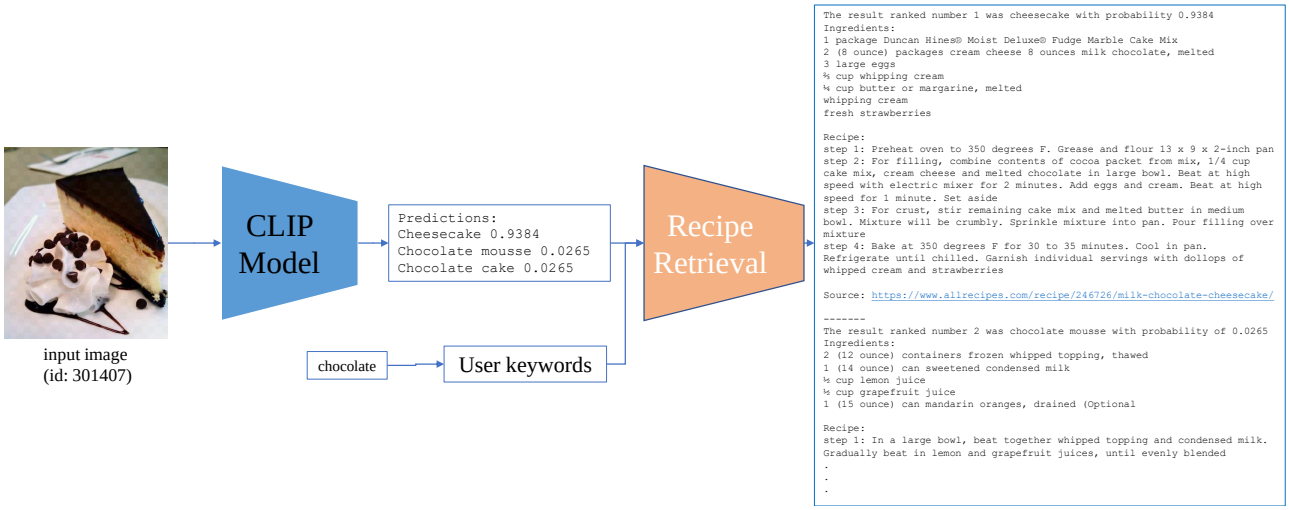

Figure 3. Example of our algorithm on a picture of a cheesecake obtained from the Food-101 dataset.

reaches around 82% top 1 accuracy, which is approximately 5-7% worse than the transfer learning models. However, the CLIP ViT L/14 model, which is trained using higher resolution images over an extra epoch[6] performs about 3% better than the ResNet-50 model.

With these results, we find that CLIP outperforms many other pretrained CNN models for food classification and requires zero time for transfer learning. This model proves to be excellent and undoubtedly scalable for future work and implementations of food classification tasks, as it performs better than the best free-use image classification models and does not require finetuning whenever the dataset is updated with new food items.

4.3. Recipe Retrieval Results and Discussion

The recipe retrieval algorithm properly leverages the search function of allrecipes to return the ingredients and cooking steps for an input image based on different user keyword inputs. In the example provided in Figure 3, the model first classifies a picture of a cheesecake and provides the top 3 predictions to the recipe retrieval algorithm. The user is prompted to supply keywords to better refine the search; in this case, our test user supplied the term ”chocolate.” Finally, the retrieval process will output the top search results for each of the top 3 predictions.

Evaluating this retrieval proves to be difficult, as there is no verifiable baseline to judge our accuracy off of. While similar projects utilize the Recipe1M+ dataset[4] which has predetermined recipes to train models on, the drawback of these models is that they can output inaccurate or incorrect recipes that are not verified by reliable sources. Our project returns ingredients and recipes that are already verified by allrecipes, which allows the model to have security in returning recipes that always make sense and taste great.

Thus, we propose an evaluation process that involves the feedback of actual people. We would construct an interface/demo where users can input their own food images then provide their feedback or rating out of 10 for the relevance of the returned recipe. This way, we can determine the satisfaction that our algorithm will bring to the general public and be able to make adjustments to our algorithm based on feedback.

5. Improvements and Future Work

The most immediate next-step to take is implementing the feedback/rating interface to evaluate our recipe retrieval algorithm. Given more time to gather feedback from real users, we would be able to give quantifiable metrics to base our retrieval model on. Additionally, we would find more varied results if we include more recipe sources in our algorithm,

Apart from these immediate changes, we can foresee several areas where we can improve our algorithm. Firstly, since our algorithm searches the allrecipes website for each query, we can significantly decrease running time by building a local search index based on all the recipes from the website. We can do this by implementing the same data mining process used in our algorithm and saving the results. Then we can build the local search index using tools such as Apache Solr[7] or similar tools.

A second change that can significantly increase the search accuracy is image feature analysis. Currently, our algorithm is limited by the top results given by the recipe website itself, so the variance of recipes returned is quite low even though the website contains thousands of recipes, and the search is biased toward the most popular or highly

[6]https://arxiv.org/pdf/2103.00020.pdf

[7]https://solr.apache.org/

scored recipes. By building an image feature identifier model, we would be able to compare the similarity between the features in the input image with features in the images that appear in the search. This way, we can find more accurate recipes than just the top results of our search queries.

References

[1] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing.

[2] K. Kheyer. Food101-classification. https://github.com/kheyer/Food101-Classification, 2019.

[3] Y. Lu. Food image recognition by using convolutional neural networks (cnns), 2016.

[4] 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.

[5] 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.

[6] 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.

[7] A. Singla, L. Yuan, and T. Ebrahimi. Food/non-food image classification and food categorization using pre-trained googlenet model. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, MADiMa '16, page 3–11, New York, NY, USA, 2016. Association for Computing Machinery.

[8] D. L. Trong. 101-food classification using transfer learning with inceptionv3. https://github.com/DucLeTrong/food101-classification, 2020.

[9] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization, 2017.