RecipeNet: Food to Recipe/Nutritional Information Generator

Sabrina Lu, Luciano Gonzalez, Dorian Raboy-McGowan

slu12@stanford.edu, lucigon@stanford.edu, dorianrm@stanford.edu

Video: https://youtu.be/9TBHQBt1u4M

Abstract

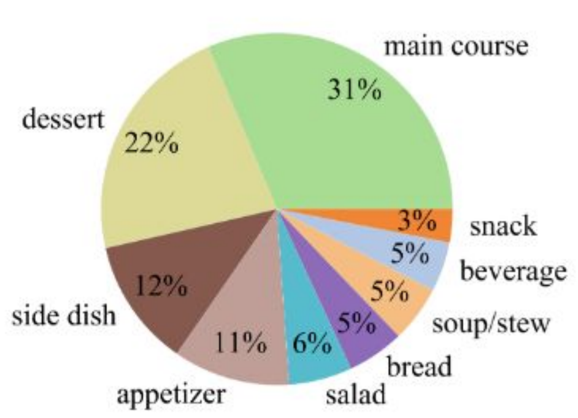
Food is fundamental to the human experience. It has deep ties to our health, our livelihood, our emotions,and our culture. More and more these days, people want to learn about what they are eating in order to make better nutritional decisions, but many struggle to find a place to start.

Our application aims to empower people to better understand and have control over their food intake in order to help achieve their health goals.

RecipeNet
Given an input image, RecipeNet returns several related recipes to give options for users to pick from. Each recipe includes ingredients and corresponding instructions for how to make it. Nutritional labels such as "under 500 calories" or "vegetarian" are provided to help guide users.

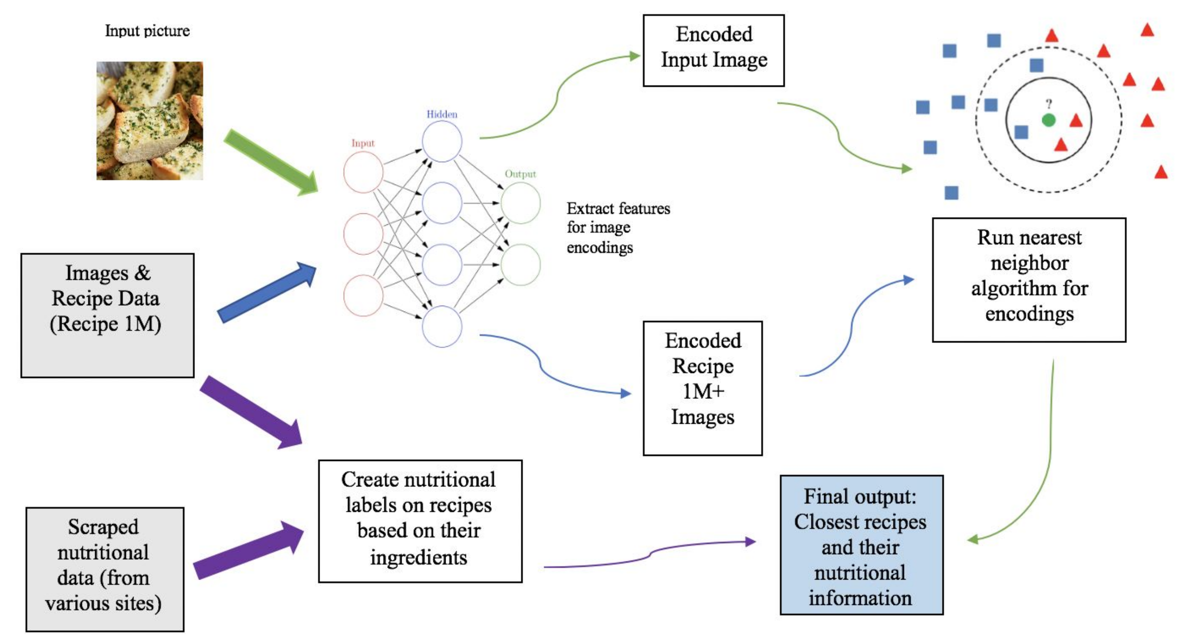
Data/Features

- **Recipe 1M dataset**: 402, 760 recipes with 887, 536 associated images



- **Dataframe 1**: Recipe id, ingredients, title, instructions, and original url location of all 1M recipes in the Recipe1M+ set.
- **Dataframe 2**: Recipe id, list of associated image ids.
  - *Preprocessed* to a dataframe of images ids and their associated recipe id
- **Dataframe 3:** Recipe id, ingredients, instructions, nutritional information by ingredient (fat, sodium, protein, etc.)
- **Nutritional List:** Scraped from various sites to compile lists of non-vegetarian, non-vegan, and non-pescatarian ingredients

Approach



Preprocessing:

- Used various CNNs to pre-process our Recipe 1M images such that we could extract encodings for each of the images to be used for similarity metrics. These encodings represent features within the images.
- Extracted nutritional information based off of the recipe ingredients as well as a separate nutritional dataset on a portion of the recipes.
  - Created nutritional labels for each recipe
- Preprocessed data frames such that image to recipe extraction would be efficient

1. <u>User input:</u> used the same CNNs to encode the input image.
2. Search linearly through our encoded dataset to calculate the k nearest neighbors to the input image.
   a. Cosine similarity
   b. Euclidean distance
3. Found the corresponding recipes and nutritional labels to the nearest images and printed them to the user

Model

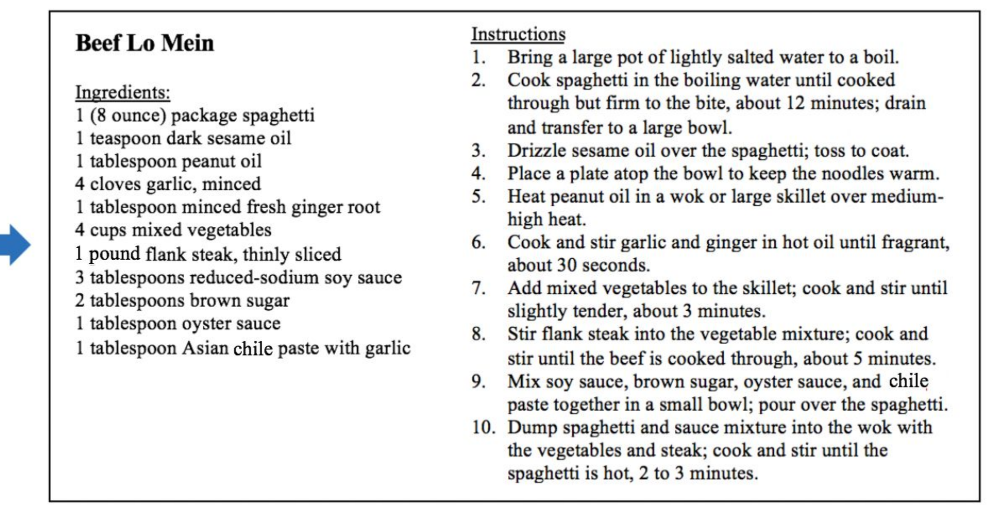
- **Transfer Learning:** Used ResNet 50, ResNet 101, and DenseNet 121 to extract the encodings for ~900K images
  - All have been pretrained to ImageNet dataset's 1,000 classes
  - ResNet 50 is a 50 layer CNN (made up of 3-layer blocks), ResNet 101 has 101 layers (also made up of 3-layer blocks), and DenseNet 121 has 121 layers (made up of 4 dense blocks)
  - Replaced the final, fully-connected layer with a single avg-pool layer to get the encoding
  - *Output*: 2048 dimensional encoding
- ResNet and DenseNet both solve the problem of vanishing gradients and slow learning in deeper networks
- DenseNet is meant to improve flow of information and gradients throughout the network by connecting each layer with every other layer

Above is a visualization of 1% of our image encodings, showing that there are similarities within encodings that may represent classes of foods

References

Salvador, Amaia, Hynes, Nicholas, Aytar, Yusuf, Marin, Javier, Ofli, Ferda, Weber, Ingmar, and Toralba, Antonio. Learning cross-model embbeddings for cooking recipes and food images. In \textit{Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition}, 2017.
Gao, Hao. "The Efficiency of Densenet." Medium. Last modified August 15, 2017. Accessed March 24, 2020. https://medium.com/@smallfishbigsea/densenet-2b0889854a92.

Results



Beef Lo Mein

<u>Ingredients:</u>
1 (8 ounce) package spaghetti
1 teaspoon dark sesame oil
1 tablespoon peanut oil
4 cloves garlic, minced
1 tablespoon minced fresh ginger root
4 cups mixed vegetables
1 pound flank steak, thinly sliced
3 tablespoons reduced-sodium soy sauce
2 tablespoons brown sugar
1 tablespoon oyster sauce
1 tablespoon Asian chile paste with garlic

<u>Instructions</u>
1. Bring a large pot of lightly salted water to a boil.
2. Cook spaghetti in the boiling water until cooked through but firm to the bite, about 12 minutes; drain and transfer to a large bowl.
3. Drizzle sesame oil over the spaghetti; toss to coat.
4. Place a plate atop the bowl to keep the noodles warm.
5. Heat peanut oil in a wok or large skillet over medium-high heat.
6. Cook and stir garlic and ginger in hot oil until fragrant, about 30 seconds.
7. Add mixed vegetables to the skillet; cook and stir until slightly tender, about 3 minutes.
8. Stir flank steak into the vegetable mixture; cook and stir until the beef is cooked through, about 5 minutes.
9. Mix soy sauce, brown sugar, oyster sauce, and chile paste together in a small bowl; pour over the spaghetti.
10. Dump spaghetti and sauce mixture into the wok with the vegetables and steak; cook and stir until the spaghetti is hot, 2 to 3 minutes.

Model	ResNet-50	ResNet-101	DenseNet-121
Cosine Loss	0.598	0.588	0.719
Average Euclidean Distance	33.52	33.47	17.97

- **Metric** - randomly sample 10,000 recipes with more than 1 corresponding image to them. Compare encodings of images associated with the same recipe
- Cosine Loss = 1 || Euclidean Distance = 0 = Perfect match of images
- Results indicate relatively high similarity/correlation of a given input image to the images of the corresponding recipe
- DenseNet-121 performed the best in terms of both cosine loss and euclidean distance

Discussion & Future Steps

Classification Improvement
- Implementing advanced classification architectures (ResNet-152 & DenseNet166)

Recommendation Improvement
- More data = better recipe recommendations (especially for higher values of k)
- Recipe 1M+ - 1 million recipes + 13 million images