ChefNet: Image Captioning and Recipe Matching on Food Image Dataset with Deep Learning

Kaylie Zhu [*1] **Harry Sha** [*1] **Chenlin Meng** [*1]

Abstract

In this paper, we develop an algorithm that, given pictures of food, is able to predict the ingredients and generate the cooking procedures. Our algorithm, ChefNet, is a joint embedding model consisting of a 121-layer convolutional neural network (CNN) and a recipe model made up of two long short-term memory network (LSTM) encoders, one for ingredients and one for instructions. Our model is trained on a subset of Recipe1M dataset (Salvador et al., 2017), which is a new large-scale, structured corpus of over one million cooking recipes and 800,000 food images. Instead of using ResNet-50 to learn the image's encodings, we use DenseNet-121 before the softmax layer in our model. We compare the performance of our model to that of the original one and find that our model has better performance on a reduced dataset.

1. Introduction

Food is central to all human life. Its significance extends beyond nutrition and health, shaping our lives as it provides us with comfort, relaxation, and reward. All of us, at one time or another, have turned to food for comfort, for coping with stress, for controlling our emotions, and for satiating desires. However, finding nutritious, satisfying and easy-to-cook food may not be a easy task for many. Food recognition technology, if developed well, may help people gain insights into the food they consume, resulting in a healthier and more satisfying diet.

These days, with the plethora of online recipe collections and user-uploaded pictures of food, it becomes possible to train machines to automatically understand food preparation by jointly analyzing ingredient lists, cooking instructions and food images (Salvador et al., 2017). In this context, we introduce our food recognition algorithm ChefNet, which studies the ingredients and cooking procedure of given food images and thereby matches given food images to their respective recipes.

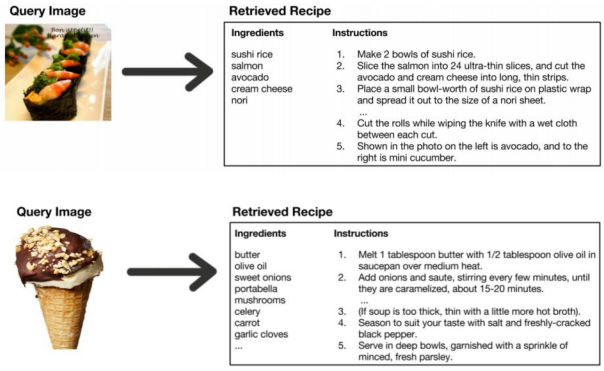

Figure 1. Illustration of instances of query image paired with its matching recipe (top) and non-matching recipe (bottom)

Our model, ChefNet, is a joint embedding model consisting of a 121-layer convolutional neural network and a recipe model made up of two LSTM encoders, one for ingredients and one for instructions. Our model takes a food image as an input and outputs the ingredients and cooking instructions corresponding to that of the image's. We train ChefNet on a subset of the recently released Recipe1M dataset (Salvador et al., 2017), which contains structured corpus of over 1m cooking recipes and $800k$ food images.

2. Related Work

In 2014, Bossard et al. (Lukas Bossard & Gool, 2014) introduced the Food-101 visual classification dataset and set a baseline of 50.8% accuracy. Later, with the improvement of both the size and the content of food image dataset, the accuracy of food recogniton has also be improved. With the impetus for food image categorization, subsequent work by (Chang Liu et al., 2016), (Austin Myers et al., 2015) and

[*]Equal contribution [1]Stanford University CS 230. Correspondence to: Kaylie Zhu <kayliez@cs.stanford.edu>, Harry Sha <harry2@cs.stanford.edu>, Chenlin Meng <chenlin@cs.stanford.edu>.

(Ferda Ofli et al., 2017) improved this result to 77.4%, 79% and 80.9%, respectively. In 2017, Salvador et al. introduced Recipe1M, a new large-scale, structured corpus of over 1m cooking recipes and 800k food images. As the largest publicly available collection of recipe data, Recipe1M affords the ability to train high-capacity models on aligned, multi-modal data (Salvador et al., 2017).

Previously, food recognition algorithms are used to enhance the accuracy of dietary measurement. For instance, Ofli et al. used image recognition to study the misalignment of how people describe food images versus what they actually depict, using the ideas from CNN. They showed that the difference between how machines and humans label images relates to a number of health outcomes observed at the county level (Ferda Ofli et al., 2017).

Food recognition algorithms have also been used to solve the practical and socially relevant problem of demystifying the creation of a dish that can be seen but not described. In 2010, Yang et al. proposed a representation for food items that calculates pairwise statistics between local features computed over a soft pixel level segmentation of the image into eight ingredient types (Shulin, Lynn). In 2016, Liu et al. proposed a new CNN-based food image recognition algorithm to derive the food information (e.g., food type and portion size) from food image effectively and efficiently (Chang Liu et al., 2016). In 2017, Salvador et al. trained a neural network to find a joint embedding of recipes and images that yields impressive results on an image-recipe retrieval task (Salvador et al., 2017).

Known for their powerful performance in image recognition tasks, convolutional neural networks have seen no short of remarkable developments in recent years, with the continuous advents of newer, deeper and better-performing CNN architectures. In 2015, ResNet-50 and ResNet-101 architectures were introduced for their relative ease to optimize and their higher accuracy gained from considerably increased depth. (Kaiming He et al., 2015) In 2016, Dense Convolutional Network (DenseNet) was introduced, which benefits from connections between each layer to every other layer in a feed-forward fashion. (Gao Huang et al., 2016) DenseNets generally achieve much better performance on image recognition tasks as they alleviate the vanishing-gradient problem and strengthen feature propagation and reuse, substantially reducing the number of parameters needed.

3. Data

We use the same dataset as in the im2recipe model (Table 1) (Salvador et al., 2017). Duplicates and recipes sharing the same image were removed, as were instances with unwanted characters or without discriminative food properties. We investigate model performance with the reduced training and validation sizes of 20,000 and 2,000 in alignment with our computational resources (Figure 2).

Partition	Number of Recipes	Number of Images
Training	720,639	619,508
Validation	155,036	133,860
Test	154,045	134,338
Total	1,029,720	887,706

Table 1. Recipe 1M Dataset. Number of samples in training, validation and test sets, accordingly.

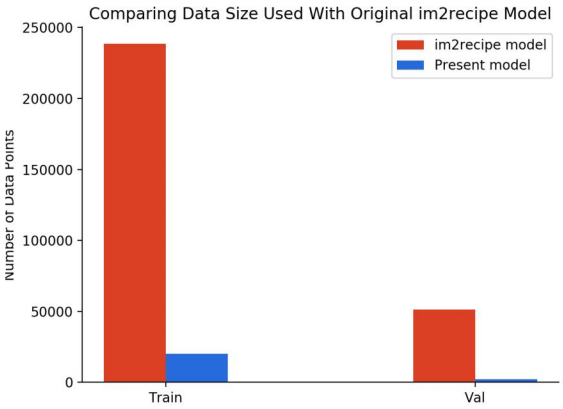

Figure 2. Size of data used in the present study (blue) compared to the size of data used on im2recipe.

4. Model

Our model is based on the im2recipe model (Salvador et al., 2017) with several modifications. The model is summarized in Figure 3.

4.1. Problem Formulation

Our goal is to maximize the similarity between the encodings of food images and the encodings of their matching recipe, and minimize the similarity between the encodings of non-matching recipe pairs.

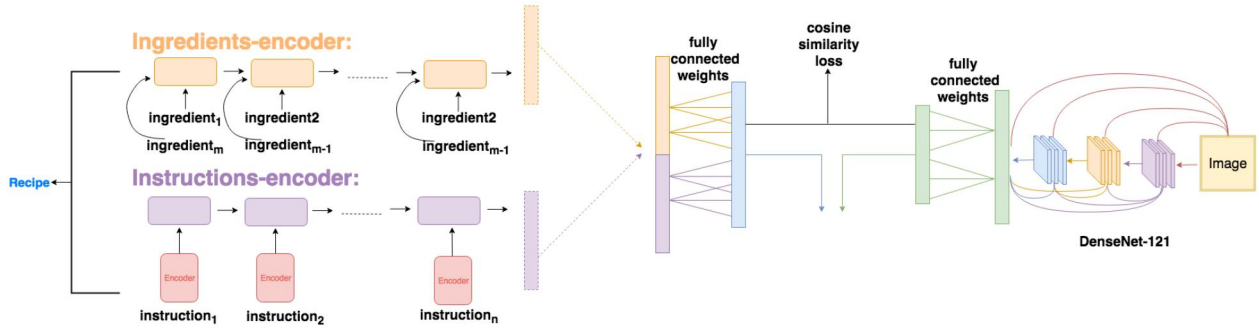

Figure 3. ChefNet Model Architecture

4.2. Image Embeddings

Salvador et al. used the ResNet-50 (Kaiming He et al., 2015) models to encode the image, and in the present study, we extend this model by experimenting with newer and deeper state art convolutional neural networks, namely DenseNet-121 (Gao Huang et al., 2016), and ResNet-101(Kaiming He et al., 2015) .

We used a DenseNet, as they have been shown to improve flow of information and gradients through the network, making the optimization of very deep networks tractable (Gao Huang et al., 2016). Furthermore, ResNet-101 was used to explore whether a deeper network may perform significantly better.

We employ transfer learning to accelerate the training process by using CNNs that were pre-trained on ImageNet(Deng et al., 2009).

4.3. Recipe Embedding

The embedding model for recipes utilizes two LSTM neural networks. A bi-directional LSTM was used for the ingredient list due to the unordered nature of the data. A forward LSTM was used to output a fixed length representation of the instructions. The ingredient and instruction embeddings are then concatenated to form the encoding of the recipe.

4.4. Joint Neural Embedding

Let v_k and r_k be the image and recipe embeddings respectively. We then map the recipe and image representations to a joint embedding space using a fully connected layer: $\phi^R = W^R r_k + b^R, \phi^v = W^v v_k + b^v$. Where W^R, W^v, b^R, b^v are also parameters to be learned.

4.5. Training

The model is trained end-to-end with positive and negative recipe-image pairs (ϕ^R, ϕ^v). We choose matching image-recipe pairs from the training set at random with 20% probability and non-matching pairs with 80% probability. Cosine similarity loss was used as the loss function:

$$L_{cos}((\phi^R, \phi^v), y) = \begin{cases} 1 - cos(\phi^R, \phi^v), & \text{if } y = 1. \\ max(0, cos(\phi^R, \phi^v)) - \alpha, & \text{if } y = -1. \end{cases}$$

with cos(.) as the normalized cosine similarity and α as the margin. $y = 1$ refers to a positive pair, and $y = -1$ refers to a negative pair. Loss was minimized using the ADAM Optimization algorithm (Kingma & Ba, 2014).

Salvador et al. suggest that simultaneously training both modalities (recipe and images) may lead to divergent results (Salvador et al., 2017). Thus, we adopt a two-stage optimization process where we first fix the weights of the image network, which are found from a model pretrained on ImageNet (Deng et al., 2009), and learn the recipe encodings. Then we freeze the recipe encodings, and learn the image network. We alternate the weight fixing every three epochs.

5. Results and Discussion

5.1. Preliminary Experiments

Preliminary experiments are conducted to determine optimal settings for hyperparameters.

First, we compare various data preprocessing techniques such as random crop followed by center crop, center cropping alone, random horizontal flipping, scaling as well as rotation on a training size of 64. We find that random crop followed by center crop performs better than center cropping alone - after 4 epochs the

latter achieves a cosine loss of 0.2283 whereas the former achieves a considerably better cosine loss of 0.1147.

Next, for each of the CNN architectures, we select a learning rate based on the model performance after 10 epochs. An example of such an experiment is shown in Figure 4. The selected learning rates are 10^{-5} for ResNet-50, 10^{-4} for DenseNet-121, and 10^{-6} for ResNet-101.

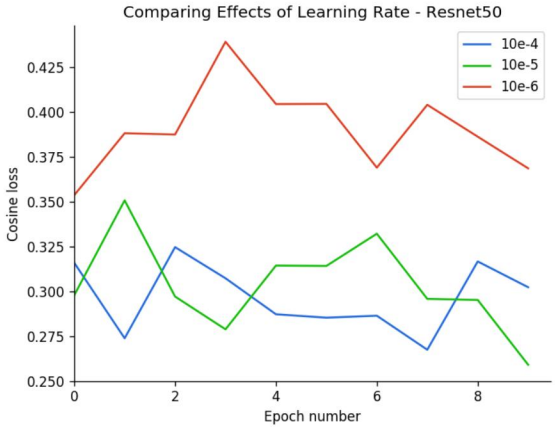

Figure 4. Preliminary experiment used to determine learning rate for the ResNet-50 model.

5.2. Comparison of CNN architectures

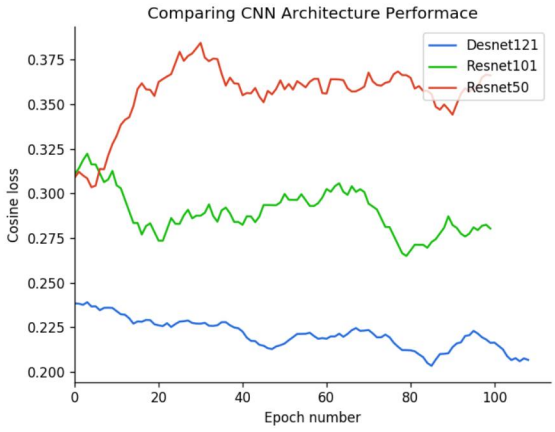

Figure 5. Performance of models using different CNN architectures

We conducted the training procedure on each of the three CNN architectures – ResNet-50, ResNet-101 and DenseNet-121. We used a batch size of 50 and trained for 100 epochs for each model, the performance of which are compared in Figure 5. The results show that using the DenseNet-121 architecture for image embedding achieves the best performance on the image captioning task, followed by ResNet-101 and ResNet-50 (which was used in the original im2recipe models).

Interestingly, DenseNet-121 only has marginally greater computational cost to train compared to ResNet-50 (roughly 26 mins more to train 100 epochs). On the other hand, ResNet-101 took almost twice as long as ResNet-50 (21.3 versus 13.1 hours for 100 epochs). These results are depicted in Figure 6.

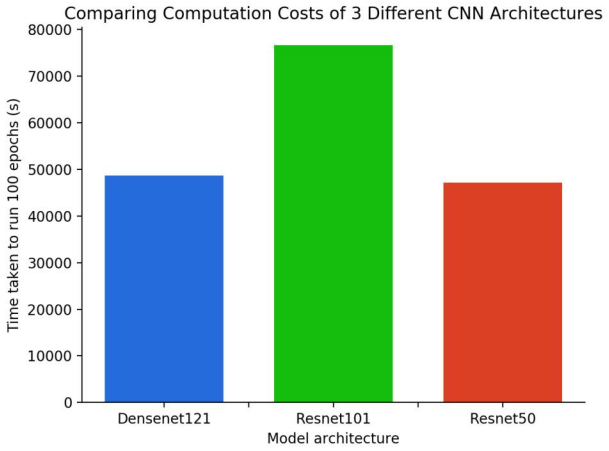

Figure 6. Computational cost for training models using different CNN architectures

5.3. Limitations

The main limitation in the present study is that we were only able train on a 20,000 images, which is 1/40 the size of the original dataset.

The evaluation metric used by Salvador et al. was based on whether the target recipe had a similarity that ranked among the top 1/5/10 recipes among 1,000 randomly chosen recipes. However, given the large variance of food images and the complexity of the problem, training on the reduced image dataset simply was not enough to achieve good performance on this particular evaluation metric. Therefore, we used cosine loss as a heuristic for the model performance. Through our investigation, we confirm that a large training set is essential for performing well in such a task.

Another limitation of our model was the inability to conduct experiments using larger batch sizes, which is known to lead to fewer parameter updates, greater parallelism and shorter training times (Smith et al., 2017). The benchmark model, im2recipe, used a batch

size of 160, while we were limited to a batch size of 50 due to GPU memory constraints.

5.4. Future Directions

At first, obvious extension to this project would be to train a model using the DenseNet-121 architecture on the full dataset. This will allow us to more aptly compare results to the current benchmark set by im2recipe, which uses the ResNet-50. Our results suggest that this model is likely to perform better.

The main modifications performed in this study is on the image encoder portion, namely the structure of the CNN. A related but unexplored issue would be to explore whether or not a different recurrent neural network architectures such as GRUs (Chung et al., 2014) might perform better for a recipe-level encoding and representation.

Finally, it would be interesting to investigate whether or not the current algorithm and model could be generalized to other "recipes" such as assembly instructions i.e., can we retrieve assembly instructions and materials given the image of a product?

6. Conclusion

Food is an indispensable part of human life, happiness and fulfillment. However, nutritious, satisfying and easily cooked food is often hard to find. With the advent of deep learning, food recognition technologies may be enhanced to enable automated understanding and investigation of food preparation by joint analysis of food images, ingredient lists and cooking instructions.

In this paper, we introduce our food recognition algorithm ChefNet, which studies the ingredients and cooking procedure of given food images and thereby match given food images to their respective recipes. Our model is an extension of the im2recipe model(Salvador et al., 2017), but instead of using ResNet-50 to learn the input image's encodings, we use DenseNet-121 before the softmax layer in our model. We also experiment with various means of data preprocessing techniques, learning rate. As it turns out, our best model uses DenseNet-121, which achieves much lower cosine losses than ResNet-50 and ResNet-101 architectures, at a negligibly higher computational cost on our dataset.

Link to repository: https://github.com/kayliez/DeepFood

7. Contributions

Chenlin created various figures including the model overview diagram and the matching versus non-matching image-recipe pair illustrations, as well as the general layout and designs for the poster. She also contributed significantly to doing research and writing up on the related works section, as well as the introduction, abstract and conclusion. Harry and Kaylie built and ran code experiments with different models, training sizes, learning rates and data pre-processing combinations, and wrote up the relevant sections including data, model, results and discussion. Each member of the team contributed substantially to the project.

References

Austin Myers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, and Jonathan Huang, Kevin Murphy. *Im2Calories: towards an automated mobile vision food diary.* 2015.

Chang Liu, Yu Cao, Yan Luo, Guanling Chen, and Vinod Vokkarane, Yunsheng. Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. 2016. doi: arXiv:1606.05675.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv e-prints*, December 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Ferda Ofli, Yusuf Aytar, Ingmar Weber, Raggi al Hammouri, and Torralba, Antonio. Is saki delicious? the food perception gap on instagram and its relation to health. 2017.

Gao Huang, Zhuang Liu, van der Maaten, Laurens, and Weinberger, Kilian Q. Densely connected convo- lutional networks. 2016.

Kaiming He, Xiangyu Zhang, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. 2015.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. 2014.

Lukas Bossard, Matthieu Guillaumin and Gool, Luc Van. *Food-101mining discriminative components with random forests.* 2014.

Salvador, Amaia, Hynes, Nicholas, Aytar, Yusuf, Marin, Javier, Ofli, Ferda, Weber, Ingmar, and Torralba, Antonio. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Shulin(Lynn) Yang, Mei Chen, Pomerleau, Dean, and Sukthankar, Rahul. Food recognition using statistics of pairwise local features. 2010.

Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. Don't Decay the Learning Rate, Increase the Batch Size. *ArXiv e-prints*, November 2017.