Food Image Classification with Convolutional Neural Networks

Malina Jiang
Department of Computer Science
Stanford University
malinaj@stanford.edu

Abstract

Food images dominate across social media platforms and drive restaurant selection and travel, but are still fairly unorganized due to the sheer volume of images. Utilized correctly, food image classification can improve food experiences across the board, such as to recommend dishes and new eateries, improve cuisine lookup, and help people make the right food choices for their diets. In this paper, we explore the problem of food image classification through training convolutional neural networks, both from scratch and with pre-trained weights learned on a larger image dataset (transfer learning), achieving an accuracy of 61.4% and top-5 accuracy of 85.2%.

1 Introduction

As it is frequently said, "we eat with our eyes". With the continued proliferation of social media platforms such as Instagram (now at 500 million daily active users [1]) as avenues for experience sharing and marketing, our digital experience becomes more and more photo-driven, and of these, over 360 million photos are photos of food (looking at just #food). Food images almost single-handedly drive dining experiences, food festivals, cooking classes, and the rise of gastro-tourism [2], with over 88% of respondents in a 2015 survey [3] considering food to be the defining element in selecting travel destinations. Most of these photos may be associated with a location or a tag, but are otherwise unlabeled, making the food search experience largely disorganized and difficult to navigate. This project explores food image classification with convolutional neural networks (CNNs) for better image labeling and clustering by dish, which in turn may improve the recommendation and search flows for a better digital food user experience overall. Specifically, the goal of the project is to, given an image of a dish as the input to the model, output the correct label categorization of the food image.

2 Related Work

In the original paper that introduced the dataset (Food 101) used in this project, Bossard et al. [4] employed a weakly-supervised mining method that relied on Random Forests (RFs) to mine discriminative regions in images, which yielded an accuracy of 50.76%, outperforming all other alternative classification methods at the time, except for the CNN approach, which as implemented by the paper achieved 56.40% accuracy on the same dataset.

A subsequent study on food image classification focused solely on the use of CNNs [5] constructed a five-layer CNN to recognize a subset of ImageNet data [6], which consisted of ten food classes. Lu's approach showcases the higher potential of CNN versus a bag-of-features (BoF) model, with the CNN model outperforming BoF by 74% to 56% accuracy. Additional data augmentation techniques were applied to bring up accuracy to 90%, which far outpaces the best BoF performance. However, given the much reduced number of classes, the paper's model performance cannot be directly mapped to model performance on the Food-101 dataset.

More recently, Liu et al. implemented DeepFood [7], a CNN-based approach inspired by LeNet-5 [8], AlexNet [9], and GoogleNet [10], employing Inception modules to increase the overall depth of the network structure. DeepFood achieved 77.4% top-1 accuracy on the Food-101 dataset after 300,000 epochs. On a separate food dataset (UEC-100 /

CS230: Deep Learning, Fall 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

UEC-256), they were able to further improve their model performance by utilizing bounding boxes to crop the image to just the dish, eliminating background noise.

Given the sparsity of studies on food images specifically, we also looked at the broader field of image classification on the ImageNet dataset, which has often been used as a benchmark for model performance. AlexNet [9], a top contender in the 2012 ImageNet Challenge, consisted of 5 convolutional (CONV) and 3 fully connected (FC) layers and used the less common (at the time) ReLU activation function to address the vanishing gradient problem, reaching a top-5 accuracy of 84.7%. VGGNet [11] in the later 2014 ImageNet Challenge differed from other top-performing models in that Simonyan et al. used fixed-size, smaller 3x3 filters to decrease the number of parameters and train a deeper model, reaching an accuracy of 92.3%. InceptionNet [10] (also known as GoogleNet) increased both the depth and width of the model using an Inception module with kernels of different sizes. Szegedy et al. were able to achieve a top-5 accuracy of 93.3% in the 2014 ImageNet Large-Scale Visual Recognition Challenge. ResNet [12] in 2015 developed even deeper models (152 layers, 8x deeper than VGG) using Residual blocks to solve the vanishing gradient problem. By increasing the depth of the model substantially, He et al. were able to further improve classification on ImageNet to a top-5 accuracy of 95.51%.

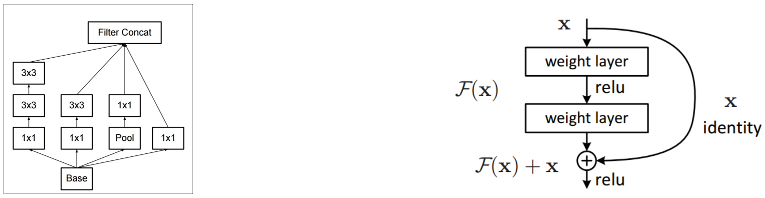
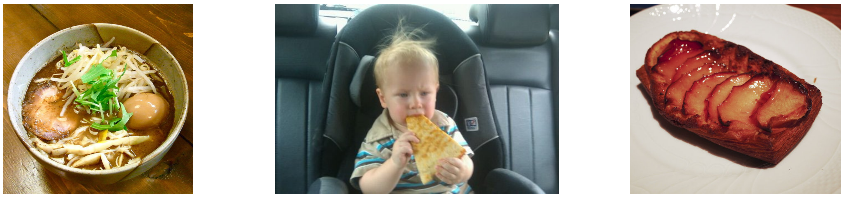


Figure 1: Inception module [10] with dimension reduction (left) and residual block [12] (right)

3 Data

A total of 101,000 images from 101 classes of food were used from the Food-101 dataset [4], with 1000 images for each class. Of the 1000 images for each class, 250 were manually reviewed test images, and 750 were intentionally noisy training images, for a total training data size of 75,750 training images and 25,250 test images. Compared to the 10-class food image dataset from ImageNet [6], this Food-101 dataset presents some additional challenges. For one, the ImageNet food image dataset contains relatively distinct and few food categories (apple, banana, broccoli, burger, egg, french fries, hot dog, pizza, rice, and strawberry), while Food-101 contains some food items that are similar in both content and presentation (e.g. pho vs. ramen). Additionally, the training dataset images were very dissimilar in lighting, coloring, and size, and also contained mislabeled images, which were left in the training dataset to encourage models to be robust to labeling anomalies. We also utilized ImageNet weights during transfer learning to boost model accuracy, though not the ImageNet dataset directly.

Images were normalized and resized appropriately, either to 128x128 or 256x256 in the initial model implementations, or to model specification when using transfer learning. Image data was augmented through rotation, shifting, and horizontal flipping to avoid overfitting. During transfer learning, images were also preprocessed using the custom model preprocessing functions, which were implementations of the image preprocessing in the original model papers.



(a) Labeled 'ramen'

(b) Labeled 'pizza' (noisy)

(c) Labeled 'apple pie'

Figure 2: Images from Food-101 dataset

4 Methods

4.1 Setup

Models were run on an Amazon Web Services Elastic Compute Cloud (AWS EC2) instances with Deep Learning AMI (Ubuntu 18.04). Models were written in TensorFlow [13] and later, Keras [14], a high-level library for deep learning. Models were saved on each epoch to make training runs more robust to failures and allow training to pick up from the last saved epoch. We also wrote several utility classes to resize / preprocess images, as well as to test models on a randomly-chosen smaller subset of classes. Performance on the full Food-101 dataset could be extrapolated from performance on the smaller subset, which accelerated model design iterations as poor-performing models could be abandoned earlier.

The loss function used across all models was categorical cross-entropy, represented below:

$$L(y, \hat{y}) = -\sum_{c=1}^{M} y * log(\hat{y})$$

4.2 Training from Scratch

As a first approach, we trained a baseline model on 64x64-sized images with 4 convolutional and 2 fully-connected layers. Within a few epochs, the highest accuracy achieved was 28.2% on the val set, but from then on the model started overfitting to the train set and accuracy only declined from there.

We hypothesized that the lower image resolution, which was sufficient for the MNIST and SIGNS datasets, did not provide enough detail for the model to differentiate food images, which tend to take on a much less structured form and can look amorphous at low resolution. Increasing the image resolution and model complexity (by adding 2 additional layers to the original baseline model) increased the accuracy to 36.3%, though again the model was overfitting to the training set.

After optimizing on the initial model design, we began looking at models proposed in image classification papers, starting with the model proposed by Lu [5], which despite the different dataset (ImageNet instead of Food-101) was still optimized on food images. The architecture of this model was composed of 3 convolutional layers of various sizes with max-pooling, followed by a fully-connected layer. Despite duplicating the paper's model architecture, there was no improvement in performance, likely due to the difference in number of classes (10 vs. 101).

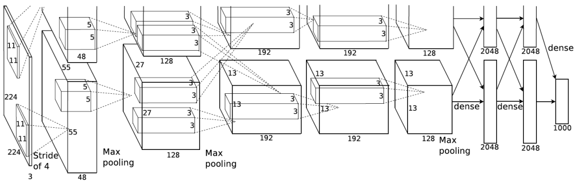


Figure 3: AlexNet architecture [15]

We introduced data augmentation at this point to address the overfitting problem by applying transformation functions to the original images, and also implemented AlexNet, which has a similar structure as the baseline model, but used filters of several different sizes instead of just 3x3 filters. Data augmentation allowed the model to train for more epochs before overfitting to the training set, and AlexNet achieved an accuracy of 32.8%

4.3 Transfer Learning

At this point, we decided to implement transfer learning from different models trained on the ImageNet dataset to take advantage of the features learned by those model using deeper architectures and with more training time, specifically VGG16, ResNet50, and InceptionV3. Transfer learning was implemented by loading the ImageNet weights into each model and freezing the base layers of each model while removing the top layers that were trained specifically on the ImageNet classes. These top layers were then replaced with trainable layers meant to learn classification on the Food-101 classes.

VGG16 was a call back to the baseline model in terms of using fixed-size 3x3 filters and was composed of a deeper model architecture, and was surprisingly slow to train. For faster training, we began looking more into ResNet50 and

InceptionV3, with both models training sans the top layer. In ResNet50, the model architecture remedies the common issues with deeper neural networks (such as vanishing gradients) through residual blocks, which allow the model to take advantage of skip connections between earlier and later layers. This essentially allows models to skip layers do not improve the overall accuracy and choose the optimal number of layers during training, boosting accuracy to 42.84%.

For InceptionV3, we experimented with unfreezing some of the base model layers, and found that training with the top few layers unfrozen for training improved performance over just training the top layer. In InceptionV3, the inception modules allow the model to train with different filter sizes at each layer (to capture both global and local information) without risking the model overfitting or being too computationally expensive. We also attempted full-layer training, but found the training extremely slow and computationally expensive. Finally, in addition to the image-preprocessing on all images done before training, we also applied InceptionV3's custom image preprocessing function to all images during training, which increased accuracy to 61.35%.

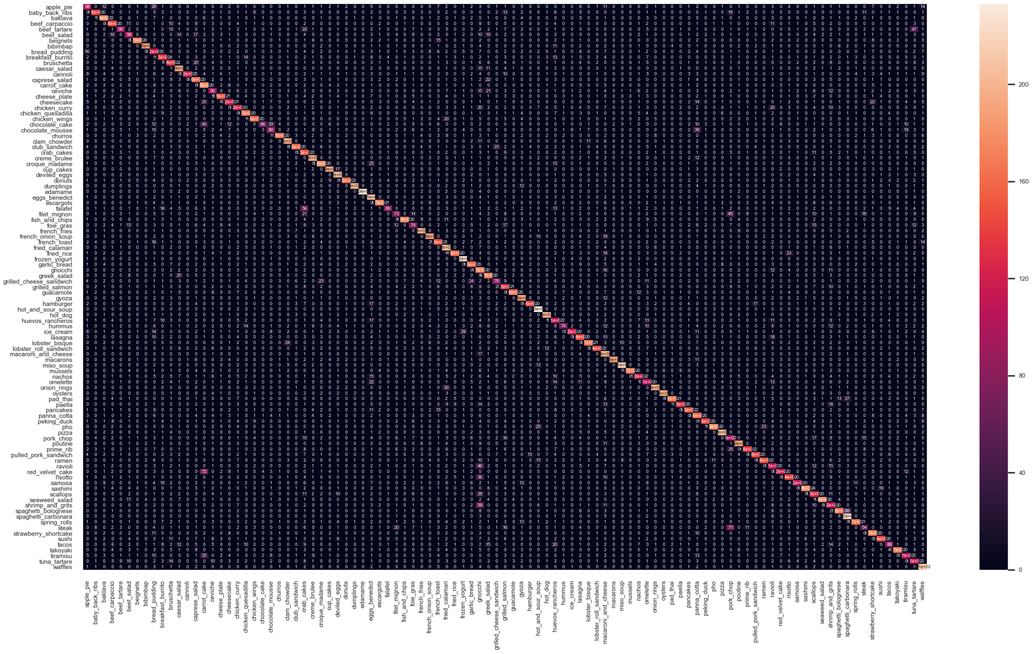


Figure 4: Confusion matrix for InceptionV3 model, actual vs. predicted

In the case of all the models pre-trained with ImageNet weights, the initial accuracy hovered around that of the base model, but then continued to increase with more epochs. The additional features on ResNet50 and InceptionV3 also allowed the models to train longer and with improved accuracy before beginning to overfit to training data.

5 Results and Discussion

In addition to the different model types trained on Food-101, for each model, we also tuned features such as the layers on top of the base models, methods of data augmentation and image preprocessing, dropout and learning rate / optimizer hyperparameters (Table 1).

The primary evaluation metrics for these models was top-1 and top-5 accuracy. Underfitting was an issue in earlier baseline models, while overfitting was a problem for deeper models when transfer learning, though this was mitigated through data augmentation, dropout, and model architecture components such as the residual block. During transfer learning, model optimizers were chosen based on their originating papers, and hyperparameters such as learning rate and momentum were chosen empirically.

From the confusion matrix (Figure 4), edamame was the most accurately labeled food class, while steak was the least accurately labeled, due to the consistency of both images in presentation, with steak taking on many more variations.

Model	Res	Epochs	Layers	Params	Model Details	Top-1	Top-5
Base7	64	8	7	-	-	28.3	-
Base9	128	5	9	-	-	36.3	-
AlexNet	227	50	9	29M	-	25.7	-
AlexNet	227	50	9	25M	data augmentation, removed dropout	32.8	61.9
AlexNet	227	50	9	59M	same padding	32.5	61.4
VGG16	224	50	19	15M	terminated early, too slow	18.8	43.5
ResNet50	224	23	52	24M	-	39.0	67.1
ResNet50	224	14	52	24M	modified optimizer	42.8	71.4
InceptionV3	299	8	50	24M	top-layer training	43.1	-
InceptionV3	299	50	50	24M	top-*N*-layer training, custom preprocessing	**61.4**	**85.2**

Table 1: Model accuracy results

Looking at the food classes most likely confused with each other, it is clear that top-5 accuracy is higher due to food classes that present very similarly visually, where the model can narrow the image class to one of several, but does not always select the right label.

Figure 5: Spaghetti bolognese (left) often confused with spaghetti carbonara (right)

6 Conclusion and Future Work

We tested different model architectures against the same Food-101 dataset and classification problem, both models trained from scratch and transfer learning with AlexNet, VGG16, ResNet50, and InceptionV3 models pre-trained on ImageNet weights. The highest performing model was a pre-trained InceptionV3 model with top layers unfrozen in stages, with total accuracy of 61.4%, which outperforms the performance of the original Food-101 paper model, but not DeepFood by Liu et al. Transfer learning was the most successful because the earlier pre-trained layers had already learned a lot of the general features needed to identify food images.

Future work would involve more optimization on hyperparameters and model aspects such as which layers to freeze versus make trainable during transfer learning. Due to computing resource and time constraints, most model implementation decisions were made by examining the convergence of the model and relative metrics from training versus validation, but an exhaustive hyperparameter search would have been a more empirical approach.

Model performance could be further improved by adding bounding boxes to the images. Some of the images from the Food-101 dataset (e.g. Figure 1b) are not properly cropped on just the food image and contain other noisy elements, which could be addressed by training another model to just tightly bound the food itself, before passing that output as an input to the food image classification model trained in this paper. Another possibility is to train models to recognize images within a subset of food (e.g. fruits vs. noodles vs. pastries), since many of the errors from the model are a result of confusing similar food items with each other (e.g. tiramisu vs. chocolate cake).

Finally, given the relatively high top-5 accuracy, we can utilize other non-image features to improve top-1 accuracy. For example, by using a food location's menu or cuisine definition, we can more confidently classify food images from the place (e.g. if the classifier identifies a noodle dish as pho or ramen but the restaurant is Japanese, we can more confidently label the image as ramen).

7 Contributions and Code

This was a solo project. All code can be found at https://github.com/malinajiang/cs230-food-model.

References

[1] J. Clement, “Instagram stories daily active users 2019.” https://www.statista.com/statistics/730315/instagram-stories-dau/, 2019.

[2] S. McGuire, “Instagram makes for hungry travelers.” https://www.business.com/articles/food-photo-frenzy-inside-the-instagram-craze-and-travel-trend/, 2015.

[3] S. McGuire, “Here’s what 88.2% of people travel the world for.” https://venngage.com/blog/88-2-of-people-travel-the-world-to-get-their-hands-on-this-infographic/, 2019.

[4] L. Bossard, M. Guillaumin, and L. Van Gool, “Food-101–mining discriminative components with random forests,” in *European Conference on Computer Vision*, pp. 446–461, Springer, 2014.

[5] Y. Lu, “Food image recognition by using convolutional neural networks (cnns),” *arXiv preprint arXiv:1612.00983*, 2016.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[7] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, “Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment,” in *International Conference on Smart Homes and Health Telematics*, pp. 37–48, Springer, 2016.

[8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.

[14] F. Chollet *et al.*, “Keras.” https://keras.io, 2015.

[15] J. Gallego, “Glomerulus classification with convolutional neural networks - scientific figure on researchgate.” https://www.researchgate.net/figure/AlexNet-CNN-architecture-layers_fig1_318168077.