CS230 Project: What's on My Plate? Identifying Different Food Categories

Mo Islam, Surbhi Maheshwari, Nate Nunta
Graduate School of Business, Stanford University
{moislam, surbhim, nnunta}@stanford.edu

Abstract

Understanding nutritional value in food is an important part of maintaining a balanced diet for a healthy lifestyle. We work towards this goal by building a multi-class food classification system that can recognize 101 different types of food items. We train using transfer learning with 30k images across these categories with four different models pretrained on the ImageNet dataset: VGG19, InceptionV3, ResNet50 and Xception. Xception produces the best results from our model search. We then conduct various hyperparameter tuning experiments, and use dropout and L2 regularization to overcome overfitting problems to produce a multi-class food classifier with precision of 0.68, recall of 0.60, and F1 score of 0.62.

1. Introduction

Balanced diet is an important component of a healthy lifestyle which in turn is crucial for happiness and fulfilment. One of the biggest hurdles to achieve a balanced diet is awareness. While the FDA publishes and enforces a rigorous food code to ensure that consumers are aware of the key nutrients in their food, this information helps only during shopping decisions. People do not have the information handy while they are eating and hence, cannot make smart choices.

Today, given the vast amount of food images present online and ubiquitous nature of mobile phones with cameras, we can create an "on-demand" nutrition fact checker. In this project, we work on the first step of this problem - recognizing the food items. Our system takes as input image of a food item and outputs the category the food item belongs to, out of 101 pre-defined categories. The data used is from Food-101 dataset [1]. Some example categories are: Apple pie, club sandwich, ice-cream, fried rice, seaweed salad, tacos and waffles.

To predict output category from input images, we deploy transfer learning on Inception, Xception, Resnet and VGG. We train the models on 30K images and compare performance across the four scenarios. Xception performed the best. So, we tuned its hyperparameters to further improve accuracy and reduce overfitting and achieved 60% validation accuracy.

2. Related work

Food recognition technologies find their applications in a lot of fields including food manufacturing and processing industry, medical and healthcare services, e-commerce businesses, nutritionists and consumer apps. Majority of the current systems rely on nutrition experts or on Amazon Mechanical Turk for recognizing food items in an image and then mapping them to their nutritional value.The initial efforts to automate this process included use of Support Vector Machines (SVMs) classifiers. Mei Chen et al [2], in 2009, published PFID: Pittsburgh fast food image dataset [3] of 4545 images and used SVM classifiers for two baseline methods. They achieved a classification accuracy of 11% with color histogram method and 24% with bag-of-SIFT-features method. While the system had limited accuracy, it set stage for future developments.

Bossard, Guillaumin and Gool, in 2014, created the Food-101 dataset [1], one of the first detailed, high quality food image dataset and used random forests to mine discriminative parts and identify various food items. They achieved an accuracy of 50.76% which outperformed most of the then-existing systems.

A lot of improvement in accuracy and performance has happened for image recognition tasks since then especially because of introduction and improvements of convolutional neural networks, availability of better labeled and organized food images and more processing power. For example, in 2016, Chang Liu et al [4] achieved an accuracy of 77.4% on Food-101 and UEC-256 datasets [5] by deploying CNNs.

In 2017, Simon Megzec et al [6] developed an algorithm called NutriNet that is a modification of AlexNet architecture. NutriNet achieved classification accuracy of 86.72% and detection accuracy of 94.47%. Nutrinet is an example of the new wave of advancements in the field which deploy transfer learning on already established, powerful CNN architectures and adapt them to specific applications. This project is an attempt in a similar direction of building on established image recognition networks and comparing performances of different such networks.

3. Dataset and features

We use Food-101 dataset. It was originally developed by Bossard, Guillaumin and Gool at ETH and re-packaged and shared on Kaggle by Mader, also from ETH. The dataset has a total of 101K images - 1000 images in 101 categories ranging from apple pie to waffles. We used total 40k images out of 101k images. These 40k images were spread out uniformly across the 101 categories. We also used a pre-packaged set of a total of 1000 images from the full Food-101 dataset to set our models up before training them on the entire data.

Before setting up or training our models, we cropped all images to size 299*299 pixels to align with the models' input requirements. Then, we created 101 arrays in h5 format by looping through each of the individual images. The data was divided in train, dev and test: 30k for training, 5k for dev and 5k for test.

4. Methods

We first conducted a model search across the four families of the Xception, Inception, VGG and ResNet, and then used hyperparameter tuning to optimize the model for our food classifier. The base models we used are described below.

I. InceptionV3

InceptionV3 was developed by Christian Szegedy et al. in 2015 [7]. It is a 48 layer deep CNN (fig 1) which uses a base inception module in repeated successions. It has one of the highest accuracy models for a relatively low number of operations required for a single forward pass amongst most popular neural network architectures [8]. It uses the added computation of extra layers efficiently by suitably factorized convolutions and aggressive regularization.

II. VGG19

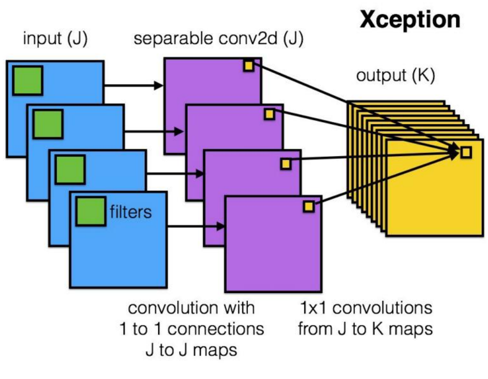
The VGG network architecture was introduced by Simonyan and Zisserman in 2014 [9]. VGG network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a softmax classifier.

III. ResNet50

ResNet50, short form for Residual Network of 50 layers, was developed by Kaiming He et al in 2015 [10]. It is also a network-in-network like InceptionV3 and has accuracy very close to InceptionV3. ResNet's salient feature is of bypassing 2 layers to feed input of one layer, say L1 to 2 layers away, say L3. Alongside the bypass, the output of L1 is fed to L2 and output of L2 is fed to L3. If dimensions of L1 and L3 are same, the identity matrix x stays the same. But if dimensions increase, zero-padding is deployed.

IV. Xception

Xception was developed by Francois Chollet in 2017 [11] as an extension of the Inception architecture which replaces the standard Inception modules with depth-wise separable convolutions. Xception reports improved performance due to a more efficient use of model parameters.



For each model, we used a categorical cross-entropy loss function as below and an Adam optimizer.

$$J(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right]$$

5. Experiments/Results/Discussion

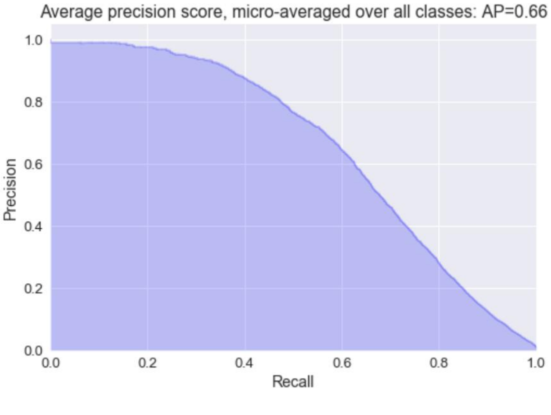
Phase 1: We trained and evaluated the four baseline models (InceptionV3, VGG19, ResNet50 and Xception) on the training and dev/validation dataset.. For each of these models, we start with each of these pretrained ImageNet models and then use transfer learning on the last 8 layers of each model for training. We also add 6 additional layers with dropout of 0.5 and 10 epochs each. As shown in the results table below, Xception performed the best, which is what we expected since it was the best-performing model on the ImageNet dataset. When calculating accuracies, we used "Top-1" accuracy, counting an image as accurately predicted if the top returned softmax percentage matched with the input image.

Model	**Training accuracy**	**Training loss**	**Val. accuracy**	**Val. loss**
InceptionV3	36%	2.62	36%	2.83
VGG19	35%	2.63	37%	2.84
ResNet50	62%	1.40	0.9%	12.26
Xception	58%	1.58	44%	2.85

Phase 2: After conducting our model search and settling on Xception, we ran a number of experiments to do hyperparameter running. We experimented with number of training layers, number of passes with variable numbers of epochs. Given our Xception had a problem with overfitting on the training set, we decided to experiment with various regularization techniques like dropout and L2 regularization in order to reduce the variance in our results. Our best performing model has the following features: 7 additional layers, two dropout layers of 0.7 and 0.7. We train with one pass of 30 epochs on 7 layers, another pass of of 20 epochs on 7+4 layers, and third pass of 8 epochs on 7+8 layers.

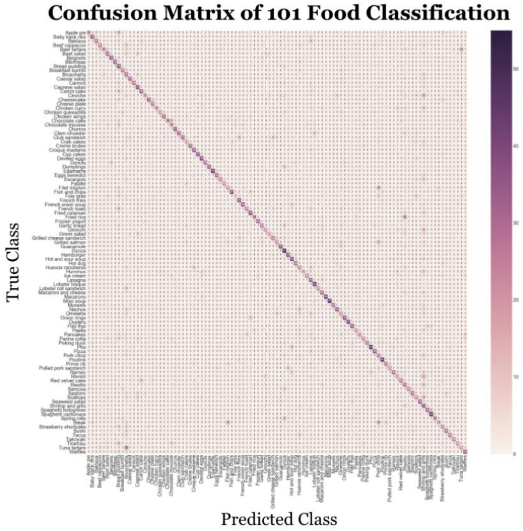
Thus we trained a total of 15 layers and were able to achieve precision of 0.68, recall of 0.60, and F1 score of 0.62. The results are presented in the figure below. We were thus able to improve our baseline Xception model with hyperparameter tuning and significant reduce the problem with overfitting, with an increase in training accuracy from 58% to 63% and increase in validation accuracy from 44% to 60%.

Error Analysis: We produced a confusion matrix for all 101 classes of food item we classified. We conducted an error analysis to show that our food classifier was systemically confused around different types of food that looked very similar, e.g. steak tartare and tuna tartare or spaghetti carbonara and spaghetti bolognese. Examples are shown below.



Spring rolls predicted as Gyoza

Spaghetti Bolognese predicted as Spaghetti Carbonara



The classifier seemed to do well even with images of various lighting conditions and food presented at different angles. We believe we can improve performance with more training examples across the 101 classes. For this dataset we trained with about 300 images per class, but can increase to almost 1000 images per class to reduce variance. Moreover, we can reduce the bias in our network by potentially training with more layers and train for longer with more epochs. These are methods we plan to employ for future work to improve our classification performance.

6. Conclusion/future work

Balanced diet is the first step towards a healthy lifestyle. Lack of easily available information is often a hurdle in maintaining a balanced diet. With advent of neural networks, it is possible to develop and train food recognition systems and further analyze the nutrient composition of each food item. In this paper, we use transfer learning to train 4 models, tune their hyperparameters and compare their results. Our best model is Xception with validation accuracy of 60%

For future work, we would like to do three types of development. Firstly, to further improve accuracy, we would train the model on more data. Secondly, the system can be extended to recognize multiple food items in an image rather than only one. That would entail a two step process of detection and then recognition. Lastly, right now we deploy a simple accuracy and confusion matrix approach (top-1 score) to compare different models. For a better comparison, we would include top-5 score. Eventually, we hope to match the predicted images with their nutrition information to provide an end-to-end solution for understanding the nutritional composition of one's plate.

7. Contributions

Each person in the team led one part and everyone collaborated on all components. Nate led cleaning data, and setting up and running experiments. Mo led design and implementation of experiments and analysis of results. Surbhi led AWS set-up and poster and report creation. We would like to thank our TA, Patrick Cho for his support and guidance.

8. Github repository

https://github.com/miislam/cs230-food

References

[1] Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool. "Food-101–mining discriminative components with random forests." *European Conference on Computer Vision*. Springer, 2014

[2] Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R., & Yang, J. (2009, November). PFID: Pittsburgh fast-food image dataset. In Image Processing (ICIP), 2009 16th IEEE International Conference on (pp. 289-292). IEEE.

[3] http://www.cs.cmu.edu/~yang/projects/diet.html

[4] Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., & Ma, Y. (2016, May). 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, Cham.

[5] http://foodcam.mobi/dataset256.html

[6] Mezgec, S., & Koroušić Seljak, B. (2017). Nutrinet: A deep learning food and drink image recognition system for dietary assessment. *Nutrients*, *9*(7), 657.

[7] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826).

[8] (2017, March 23). Neural Network Architectures – Towards Data Science. Retrieved from https://towardsdatascience.com/neural-network-architectures-156e5bad51ba

[9] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[11] Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*.