Computer Vision and Food Classification
Springboard Datascience Intensive Capstone Report

Simon Bedford

01 September 2016

Contents

1 Introduction

Cooking, and more broadly food in general, is a very popular content category online. Indeed, according to the FT.com, cooking has become a key category to focus on for driving growth in both audiences and advertising alike.

When someone is looking for a specific recipe, or even just inspiration, it is now perfectly natural for them to go online and consult one of the myriad recipe curating and aggregating websites.

From big names such as Allrecipes.com, Food.com and the Food Network, to more focused sites like Epicurious and Serious Eats, there is an increasing proliferation of online properties where consumers can find, view and share recipes and cooking ideas. Nowadays even companies like Youtube and Buzzfeed are trying to drive traffic through cooking videos and websites [1].

One only has to look at some of the numbers to see how big these websites are:

- In December 2016, allrecipes.com drew 50m unique visitors, a 23% increase from the same month the previous year [1]
- Epicurious boasts 30,000 professionally tested and created recipes, along with 150,000 member-submitted recipes.[2]
- The New York Times cooking website receives 7-8m monthly visitors, has 650,000 subscribers and stores more than 17,000 recipes online[1]

Perhaps even more impressive is the related growth in photo and video content:

- There are 168m+ posts on Instagram with the hashtag #food, and 76m+ for #foodporn [3]
- Food related videos were viewed 23bn times in 2015, a 170% increase over 2014 [1]

Furthermore, and inline with broader trends, mobile is where this contents is increasingly produced and consumed. Allrecipes.com for instance states that 66% of their page views are now via mobile devices. [4].

Given both the rapid growth in and focus on food, and the increasing dominance of photo and video media, there is a clear opportunity for computer vision techniques to be applied to the world of food content.

For example, image recognition and classification techniques could be integrated into a seamless mobile experience to enable faster and more accurate recipe search and suggestions, and even enable content producers to display more relevant and targeted advertising specific to the types of food that a user is interested in.

However applications are not just limited to search and advertising, but can also have health benefits too. For example food classification algorithms could help people keep food diaries and daily calorie counts, an idea explored by both Microsoft [5] and Google research teams [6].

The aim of this project is as proof-of-concept, to explore whether it is possible to create a food classification algorithm that could feasibly be implemented into a production environment to enable automated food recognition and recipe retrieval.

As such, it is key to consider the right measures of performance:

1. Classification Accuracy
2. Resources and Cost
3. Ease of Production Deployment

Classification Accuracy

We could use a very simple metric:

$$Simple\,Accuracy = \frac{\#\,Correct\,Predictions}{All\,Predictions}$$

However this is a slightly naive measure of accuracy where we are just looking at the proportion of correct predictions we make. Instead we will use the F1-score as a more refined measure of accuracy:

$$F1\,Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

To better understand the F1-score, consider a very simple classifier attempting to distinguish between pizza and not-pizza. There are four possible prediction outcomes:

		True Categories	
		Pizza	***Not Pizza***
Classifier Predictions	***Pizza***	True Positive (TP)	False Positive (FP)
	Not Pizza	False Negative (FN)	True Negative (TN)

We can then define:

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives}$$

and

$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives}$$

Another way to think about this is Precision tells us, out of all images predicted to be pizza, what proportion are actually pizza. In some senses this tells us about how noisy are our pizza predictions, or how much does the classifier confuse other types of food with pizza.

Similarly Recall asks, out of all true pizza images, what proportion were correctly identified as being pizza. This is a measure of how good our classifier is at correctly identifying pizza images and placing them in the correct category.

The F1-score is the harmonic mean and is a measure of the balance between Precision and Recall. It is measured on a scale from 0 to 1, with 1 being the best possible score.

Resources and Cost

For Resources what we care about is how costly is it in terms of both time and money to create an accurate classifier. From a practical perspective, any company evaluating implementing this type of technology would need to perform some type of ROI analysis, including taking into account the costs of training an algorithm with sufficient accuracy.

During our analysis we will measure the time taken in training different algorithms, and also the cost of any external computing resources required. For the sake of this project we will ignore both man-hours and also negligible costs such as electricity etc.

Ease of Production Deployment

It is also important to consider how practical it would be to implement a successful model in a production environment in a way that makes sense and would benefit the customer or user experience.

For example, some questions to consider are:

Can the algorithm perform accurate classification in real time? Is it possible to run the model locally on a mobile device, or does it need to be run on a central server?

During this project we will not spend a lot of time analyzing these questions in detail, however in the conclusions and recommendations we will briefly discuss some possibilities for integrating the best model into production systems, along with associated challenges.

Classification Models

Finally, with regards to models, we take three approaches, which will be explored in more detail in later sections:

- Traditional machine learning
- State of the art deep learning techniques
- A hybrid approach fusing deep-learning with traditional machine learning

As will be seen, modern deep-learning techniques enable us to achieve significant results in a relatively short amount of time, with a good possibility of applicability to different contexts.

2 Exploratory Data Analysis

The dataset used is The Food-101 Data Set from the ETH Zurich Computer Vision Laboratory [7]. This dataset contains 101 food categories, with 1,000 images per category for a total of 101,000 images. Each image has a maximum side-length of 512 pixels, and the total dataset comprises approximately 5GB of data.

Early on in the project, it was decided that, given the available computing resources, the original dataset was too large for practical experimentation.

In order to create a smaller dataset the existing categories were compared to the 100 most popular recipe-related Google search terms in 2015 in the US. In total, there was a direct match with 13 classes in the dataset, and for the remainder of the project, the analysis is focused on the top 12 matching categories:

Pork Chop	Lasagna	French Toast
Guacamole	Apple Pie	Cheesecake
Hamburger	Fried Rice	Carrot Cake
Chocolate Cake	Steak	Pizza

Noisy Data

The images in the Food-101 dataset are of mixed quality. Some are very clear, well-lit and framed on the food item in question. Others however are more noisy, poorly lit, containing other irrelevant items and, in some cases, mislabeled. Some examples of both high and low-quality pictures are shown in figure 1.

Image Sizes

Looking at the different image sizes and shapes within the dataset, we see that the top 5 most common shapes are:

Shape	% of Images
(512, 512, 3)	59.4
(384, 512, 3)	15.8
(512, 384, 3)	6.4
(512, 382, 3)	2.8
(382, 512, 3)	2.5
Other	13.2

The most common image size is 512 x 512 accounting for 60% of the images, and so the whole dataset was standardised to this size for future analyses.

RGB Histograms

One of the first exploratory steps was to compare RGB histograms for the images across the different food categories.

Figure 1: Select pictures of Pizza, Hamburger and Steak categories, illustrating both high and low-quality images.

Histograms for the RGB channels or the average image for each class show that each class is quite different, as can be seen in figure 2.

Principal Component Analysis

The next step was to visualize the data by plotting images in 2-D to see if any clear patterns emerged, for example images from the same class being clustered together.

For dimensionality reduction to 2-dimensions, the following approach was used:

1. Initial reduction using Randomized PCA with n_components = 50
2. Further reduction to 2-D using TSNE (t-distributed Stochastic Neighbor Embedding)

Given that in total the dataset comprises 12,000 images, for practical purposes we used only the 40 nearest neighbors to the mean image for each category (480 images in total).

As we can see from figure 3, based upon the extracted components, there are no discernible patterns or groupings in two-dimensions.

3 Traditional Machine Learning

It is quite hard to find a good definition that clarifies the difference between Machine Learning and Deep Learning. There are many similarities in both the language used when discussing them,

Figure 2: Histograms of RGB pixel values for mean images for the Guacamole, Chocolate Cake and Pizza categories.

and in the domains to which they are applied. In fact, in some senses Deep Learning can be considered to be a subset of Machine Learning.

However, for the purposes of this project, when we talk about traditional Machine Learning, we will mean the use of manually or semi-manually extracted features that are used in conjunction with computer-enabled algorithms to attempt to solve our classification problem.

In particular, we will focus on using *Supervised Learning* algorithms, whereby we attempt to train a model using a set of features that represent our images along with know labels, with the hope that the model will be able to predict the class correctly for unseen images.

The key point when we talk about Machine Learning is the use of *manually or semi-manually extracted features*. Here the key to success is being able to find a set of features that:

1. In some sense are similar enough for images of the same class
2. Are different enough between classes to enable differentiation

Over the course of the project, the types of features considered were:

For the very earliest attempts, some features were tested individually, but in most cases, different types of features were chained together before being used for model training.

In some cases these features were extracted from the image as a whole, and in other cases the images were divided up into sub-sections or windows and then features extracted from each sub-section before being chained together to create one long vector of features for the whole image.

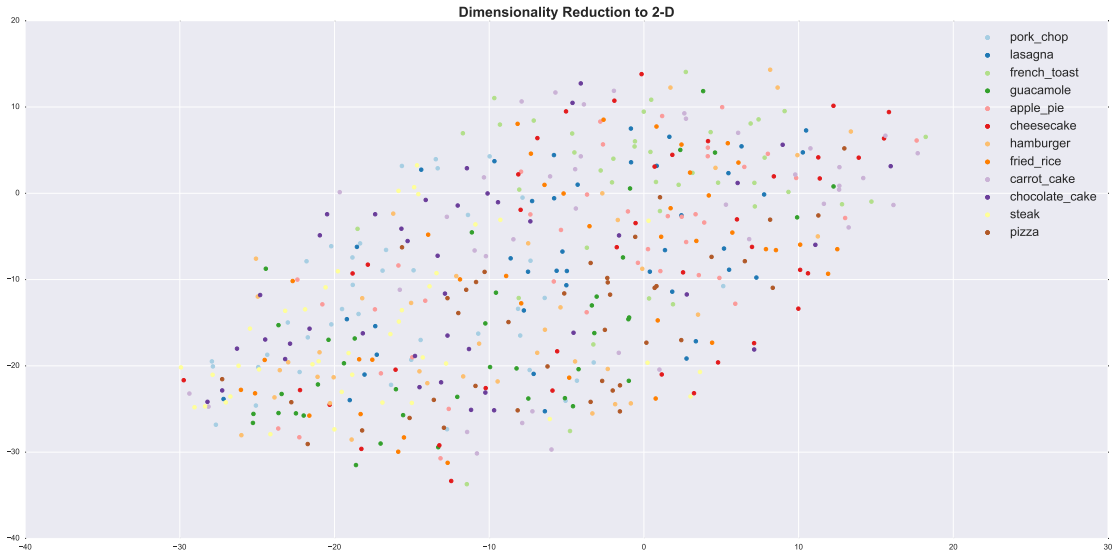

Figure 3: Images plotted in 2-D; First PCA was applied to reduce from 786,432 to 50 dimensions, and then TSNE further applied to reduce to 2 dimensions.

RGB Histograms	As we saw in figure 2, it initially appears that different types of food images have sufficiently different distributions of Red, Green and Blue pixel values to potentially be able to differentiate between them.
Pixel Values	In some models, we used individual pixel values as features, typically for images scaled to a smaller size.
Edges	These features are extracted using image processing algorithms that look for...
Corners	The intersection of two edges
Meta Approaches	One approach that can sometimes be of benefit is applying unsupervised learning techniques in order to reduce the dimensionality of other extracted features in order to make supervised learning more practical and faster to run. Two approaches tried were Principal Component Analysis as well as K-means Clustering.

Models

As well as trying a number of different combinations of different features, we also started by testing a number of different models on the simplest features, including:

- k-Nearest Neighbours
- Linear SVM
- SVM with Polynomial Kernel
- Decision Tree
- Random Forest
- ADA Boost Classifier

- Gaussian, Multinomial and Bernoulli Naive Bayes
- Linear & Quadratic Discriminant analysis

Based on the results of training and testing these models, see figure 4, it was decided to mainly use Random Forest classifiers, as the RF gave the best accuracy and F1 scores using the simple features, and also had a very fast training and testing time.

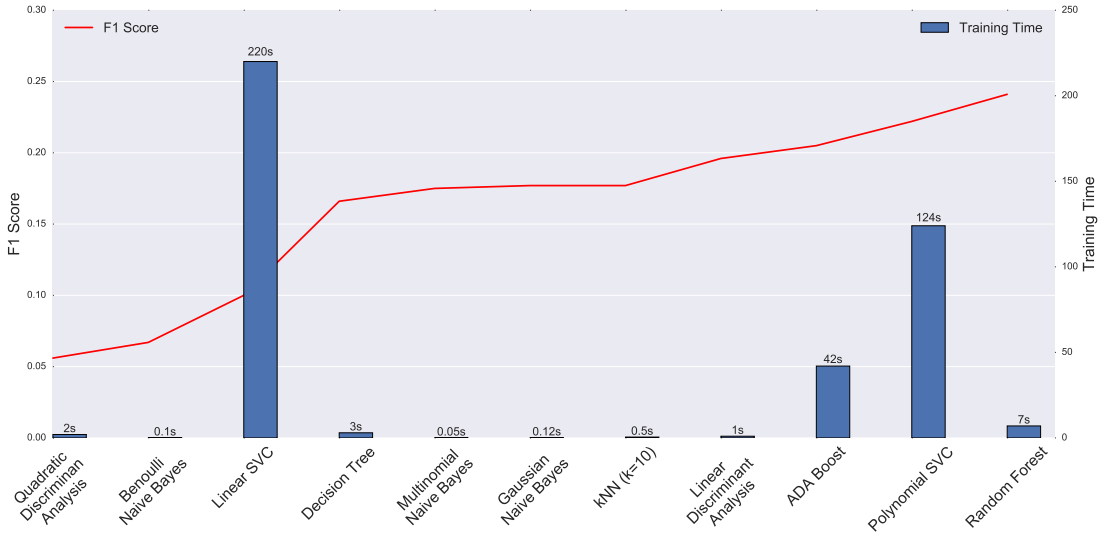

Figure 4: Comparison of F1 score and training time of different classifiers using RGB Histogram-based features.

All models, both supervised and unsupervised, were implemented using the python Scikit-learn library.

In total we tried approximately XX approaches using a mix of features and classifiers. The full list of approaches and results can be seen in Appendix X, however a selection of approaches is below:

The best approach was based upon the following procedure:

Feature extraction

1. Split each image into non-overlapping boxes of side 32 pixels; for the images of size 512 x 512, resulting in a grid of 256 boxes per image
2. For each box, extract the following features:
   - Average red pixel value
   - Average blue pixel value
   - Average green pixel value
   - Number of edges (using skimage canny edges algorithm)
   - Number of corners (using skimage corner-fast algorithm)
3. Concatenate all features together into one long feature vector of length 1,280

Classifier

Table 1: Results of varying combinations of features and supervised learning models.

Classifier	**Features**	**F1 Score**
Random Forest	100-d vector extracted using K-Means Clustering on individual pixels	0.09
Random Forest	100-d vector extracted using PCA on individual pixels	0.07
Random Forest	Images re-scaled to 32 x 32, pixels as features	0.19
Random Forest	RGB Histogram for complete image + individual pixels of 32 x 32 image	0.26
Random Forest + Grid-search	RGB Histogram + Edges + Coreners; zero-variance features removed; PCA to reduce to 300-d vector of features	0.27
Random Forest + Grid-search	Image split into 32x32 boxes; for each box extract Avg Red, Green, Blue pixel value, # Edges, # Corners	0.31
Random Forests trained on each feature type; Bayesian Net training on probabilities from each classifier	Image split into 32x32 boxes; for each box extract Avg Red, Green, Blue pixel value, # Edges, # Corners	0.29
Random forest trained on segments; overall prediction based on average of segment predictions.	Image split into segments (using SLIC algorithm) and for each segment use Avg, Max, Min, Range of Color values & normed histograms for RGB	0.21

The model used was a Random Forest. Parameter optimization was performed using a grid-search on the number of estimators and maximum tree depth, and then a classifier was trained using the best values. The results obtained are shown in table 2.

It can also be instructive to look at the confusion matrix and see which categories the classifier tends to get right or confuse with other classes, see figure 5.

We can see that Pizza and Cheesecake appear to be the categories where the classifier performs best, although with correct predictions for 133 and 126 instances respectively out of 250, there is still a long way to go to have a model that could be usefully applicable.

In terms of where the classifier struggles, by looking at the darker shaded cells off-diagonal, we see that the model has particular difficulty distinguishing between Steak and Pork Chops, and also between the different categories of cakes.

4 Deep Learning

The second approach was based upon deep learning techniques, and in particular Convolutional Neural Networks (CNNs).

Deep learning refers to X. Currently CNNs are recognised as being state of the art models for image classification (reference), and in fact various image classification and captioning competitions have been won consistently over the past X years using CNNs (reference).

The challenge is that training a full CNN from scratch requires a lot of data, and takes a long time (reference). However all is not lost as there exist a number of techniques whereby it is possible to take a pre-trained model, and adapt it in some way shape or form to a different but

Table 2: Per-class results from the best machine learning classifier.

Class	Precision	Recall	F1	Support
Pork Chop	0.29	0.29	0.29	250
Lasagna	0.31	0.21	0.25	250
French Toast	0.37	0.28	0.32	250
Guacamole	0.28	0.32	0.30	250
Apple Pie	0.19	0.14	0.16	250
Cheesecake	0.35	0.50	0.41	250
Hamburger	0.30	0.23	0.26	250
Fried Rice	0.35	0.46	0.39	250
Carrot Cake	0.26	0.16	0.20	250
Chocolate Cake	0.36	0.36	0.36	250
Steak	0.26	0.29	0.28	250
Pizza	0.38	0.53	0.44	250
Overall	**0.32**	**0.31**	**0.31**	-

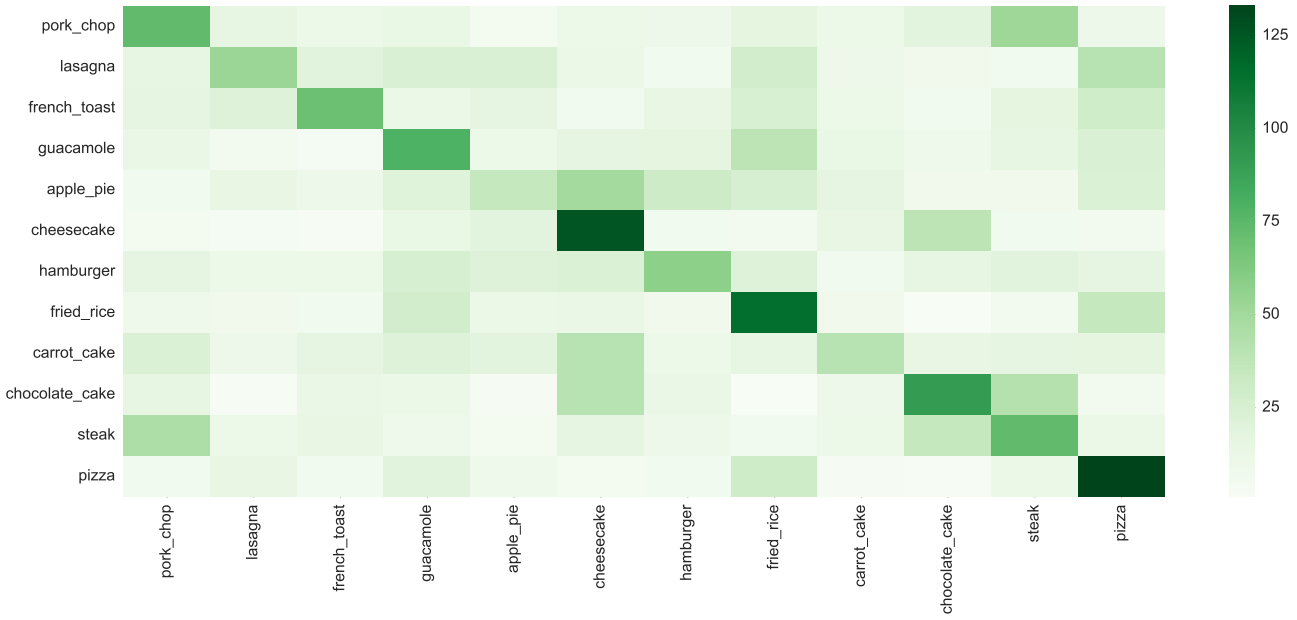

Figure 5: Confusion matrix for classifier predictions. Darker shaded cells indicate more correct predictions for that class.

related problem, through a series of techniques called Transfer Learning.

The two types of transfer learning used were:

1. Extracting features from the pre-trained model and then training a separate classifier on those features.
2. Fine tuning the weights of the network by training it on the dataset.

Benefits of these approaches.

5 Comparison of Results

6 Fail Cases

7 Conclusion and Recommendations

References

[1] http://www.ft.com/cms/s/0/f609954c-1d46-11e6-a7bc-ee846770ec15.html

[2] http://www.epicurious.com/about/press-center

[3] http://www.business.com/social-media-marketing/food-photo-frenzy-inside-the-instagram-craze-and-travel-trend/

[4] http://press.allrecipes.com/

[5] http://research.microsoft.com/en-us/um/redmond/projects/menumatch/

[6] http://www.popsci.com/google-using-ai-count-calories-food-photos

[7] https://www.vision.ee.ethz.ch/datasets_extra/food-101/

[8] The Japan Reader *Imperial Japan 1800-1945* 1973: Random House, N.Y.