Computer Vision & Food Recognition

Simon Bedford
September, 2016

Food is an increasingly important content category

50m visitors in Dec 2015

30,000 professional recipes
150,00 user-submitted recipes

Cooking website:
7-8M monthly visitors

168M+ posts #food
76M+ posts #foodporn

Food-related videos viewed 23bn times in 2015

Opportunities for Machine-learning + food images

Machine Learning Algorithm

Classified Food item(s)

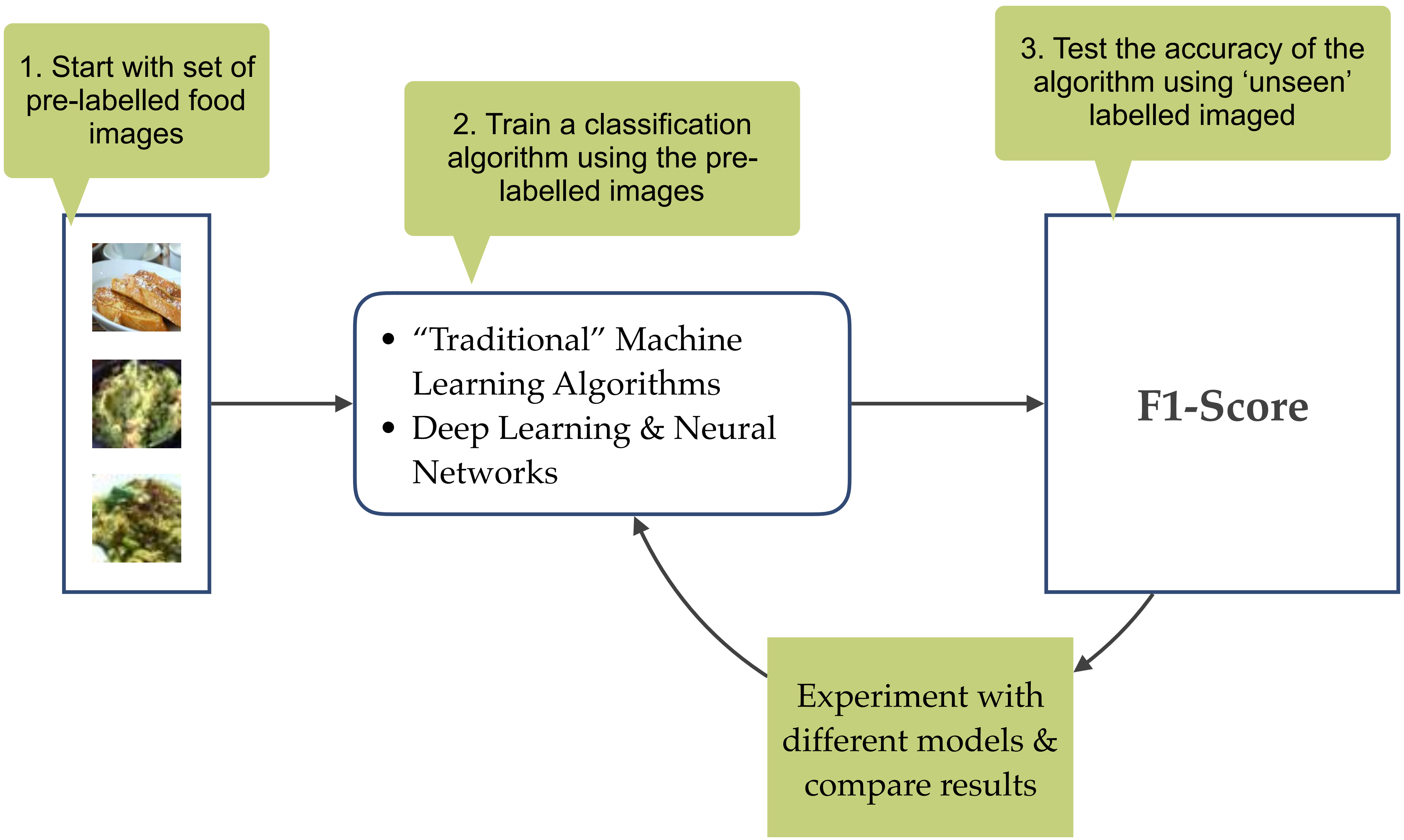
Find relevant recipes

Estimate calorie count

Identify ingredients

Problem and approach

The aim is to try and create a food classification algorithm that can take images of prepared food dishes as input and output a prediction for the type of dish



Requirements for algorithm:

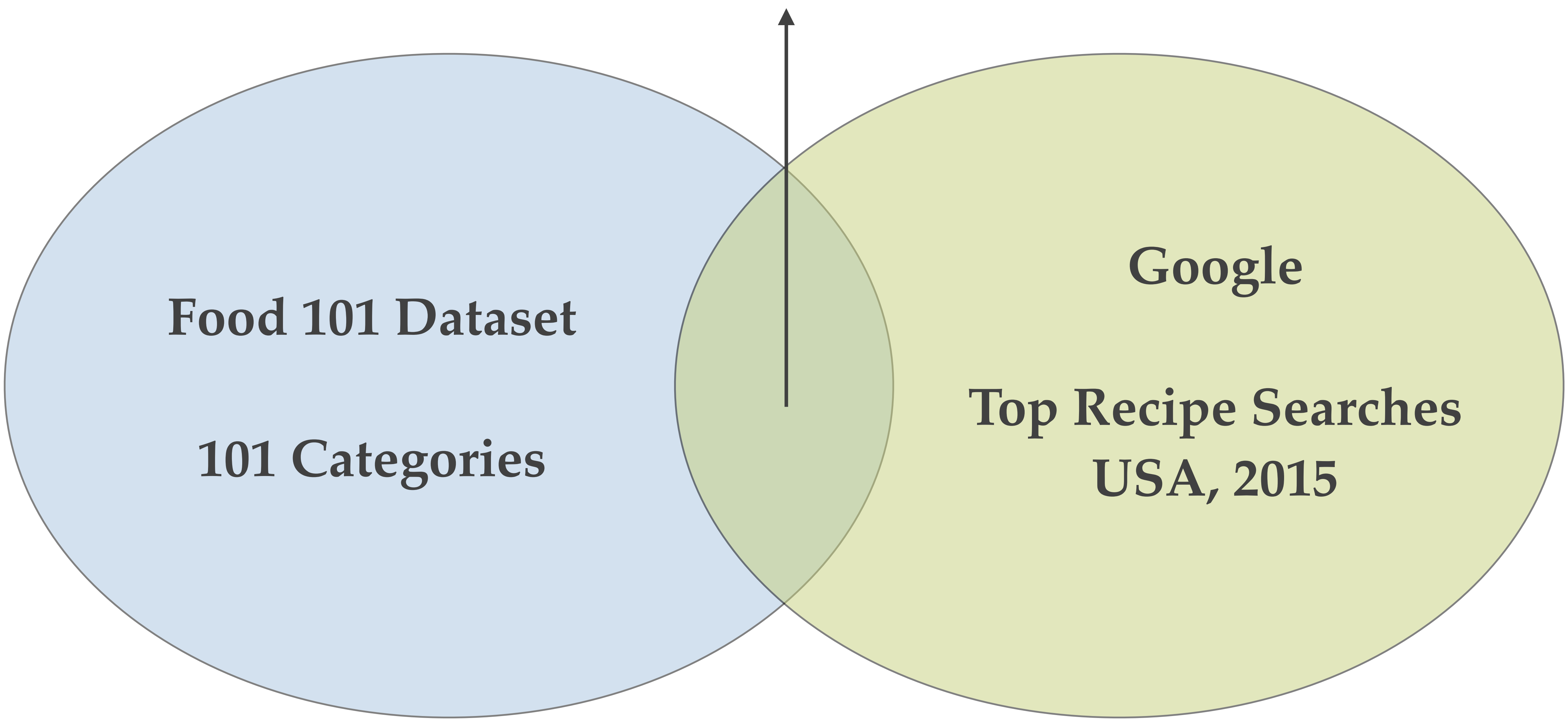
- Able to train using reasonable resources
- Able to make predictions quickly (<1s)
- Sufficient level of accuracy

The dataset

- Food-101 Data Set from the ETH Zurich Computer Vision Laboratory
- 101 Categories
- 1,000 images per category
- Most common shape (512, 512)

Focus on 12 categories

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



Mixed data quality

Differences in RGB histograms between image categories

Approach 1: Machine-learning models + features

Models:

- k-Nearest Neighbours
- Support Vector Machines
- Decision Trees
- Random Forests
- ADA Boost Classifier
- Naive Bayes Classifiers
- Linear & Quadratic Discriminant Analysis

"Supervised" learning algorithms

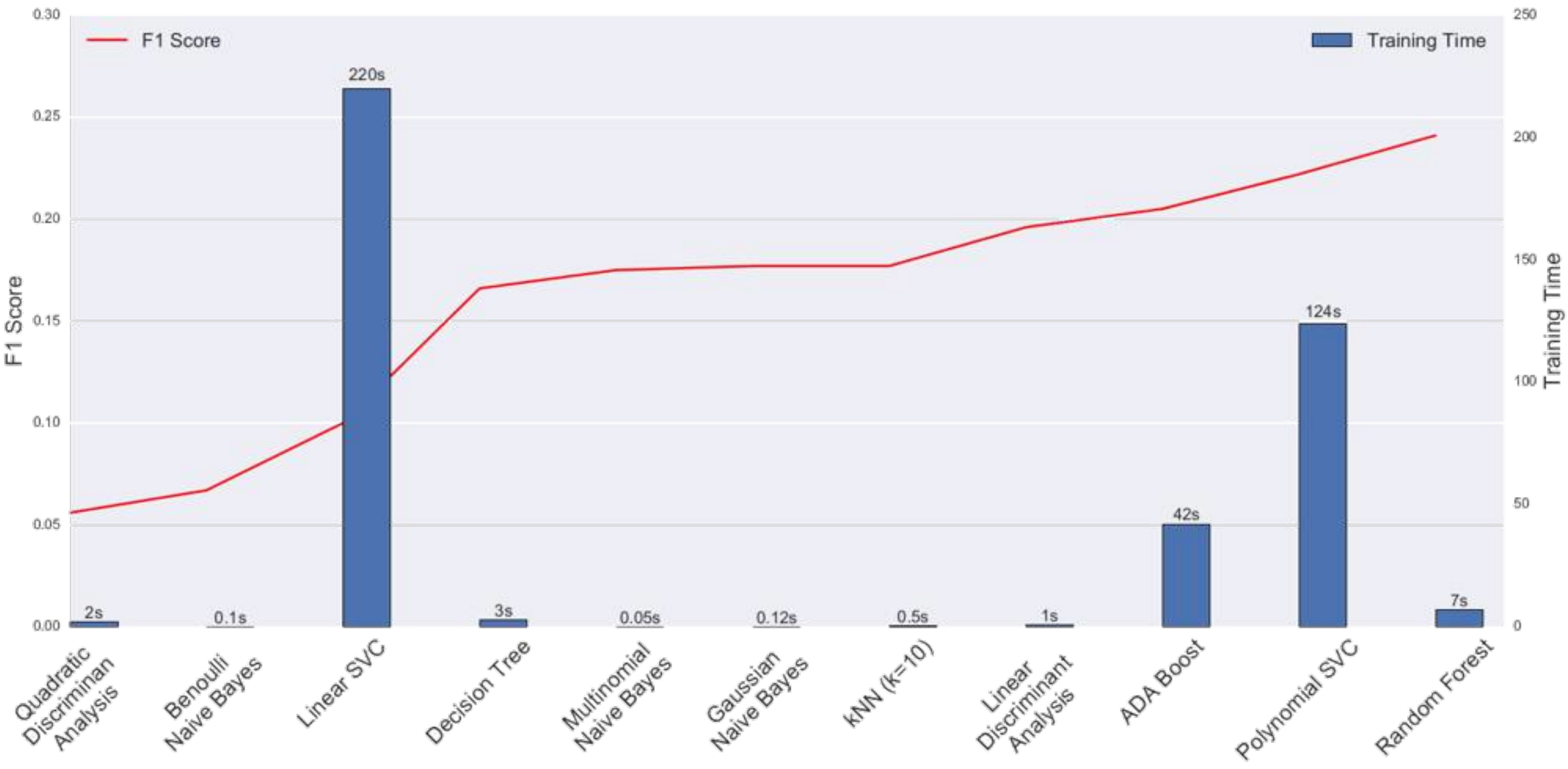
+

Features:

- RGB Histograms
- Individual Pixel Values
- Number of Edges
- Number of Corners
- Unsupervised methods (e.g, Principal Component Analysis, k-Means Clustering)

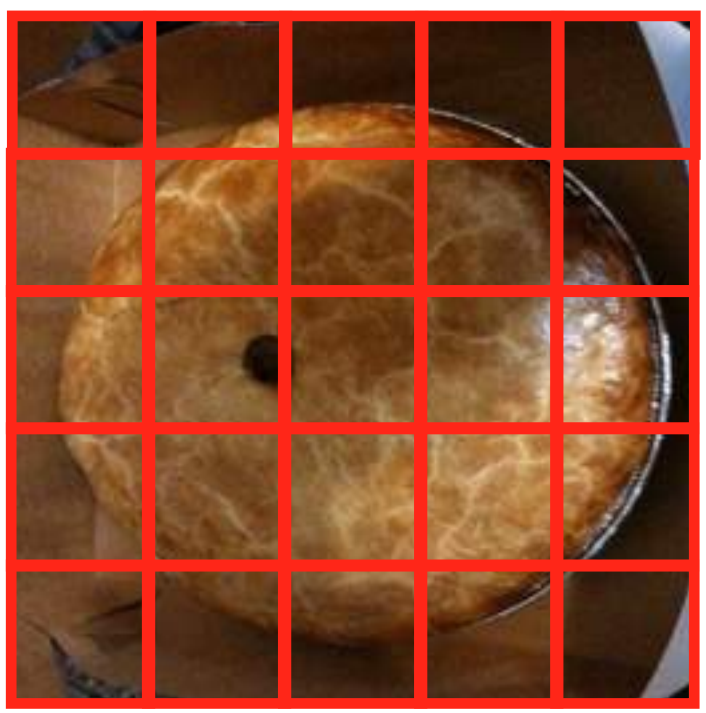
Manually chosen features

Comparison of classifiers



Best machine learning procedure

Image divided into 32 x 32 grid (256 cells in total)



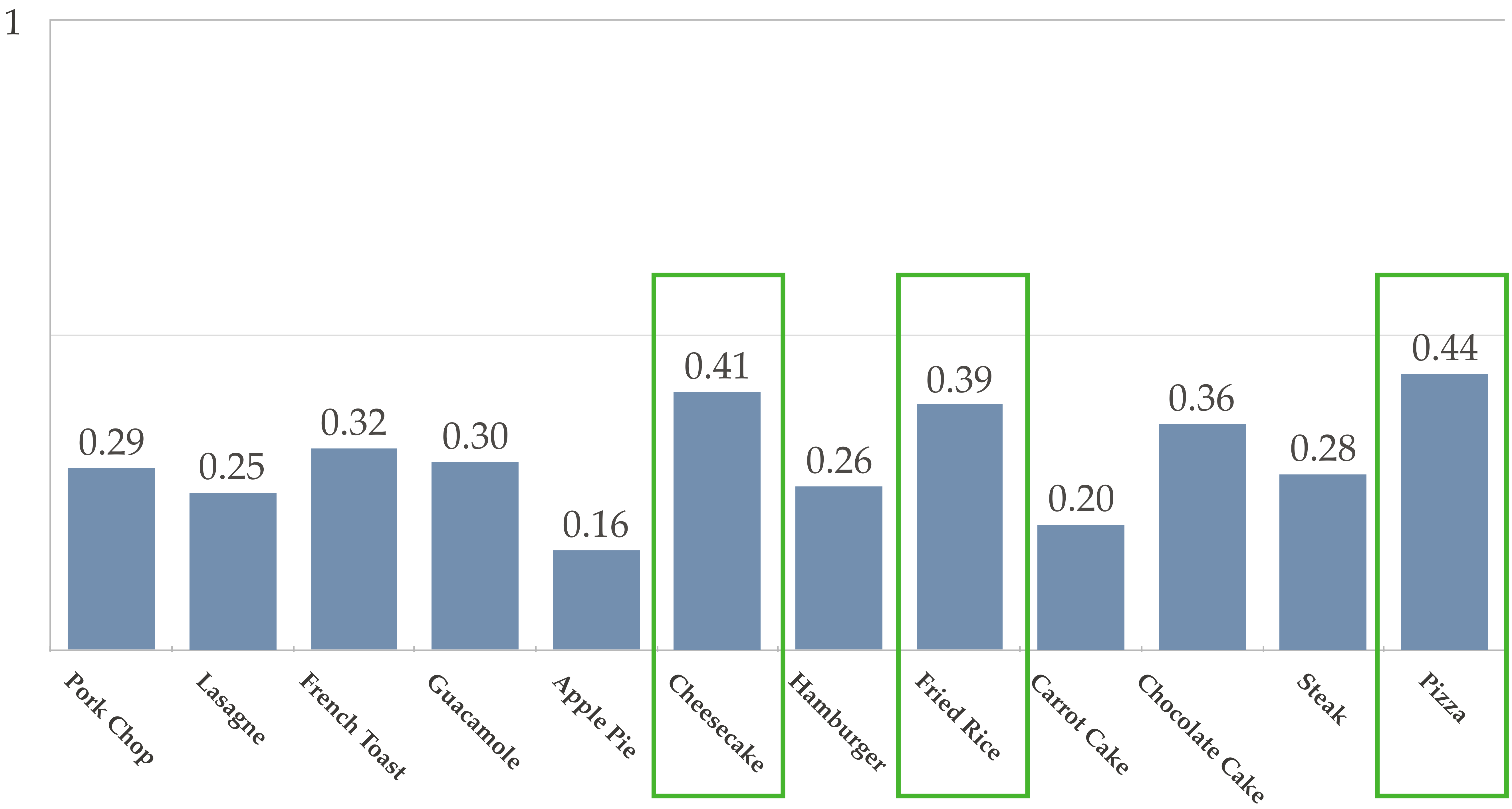
For each cell calculate:

- Average red pixel value
- Average green pixel value
- Average blue pixel value
- Number of edges
- Number of corners

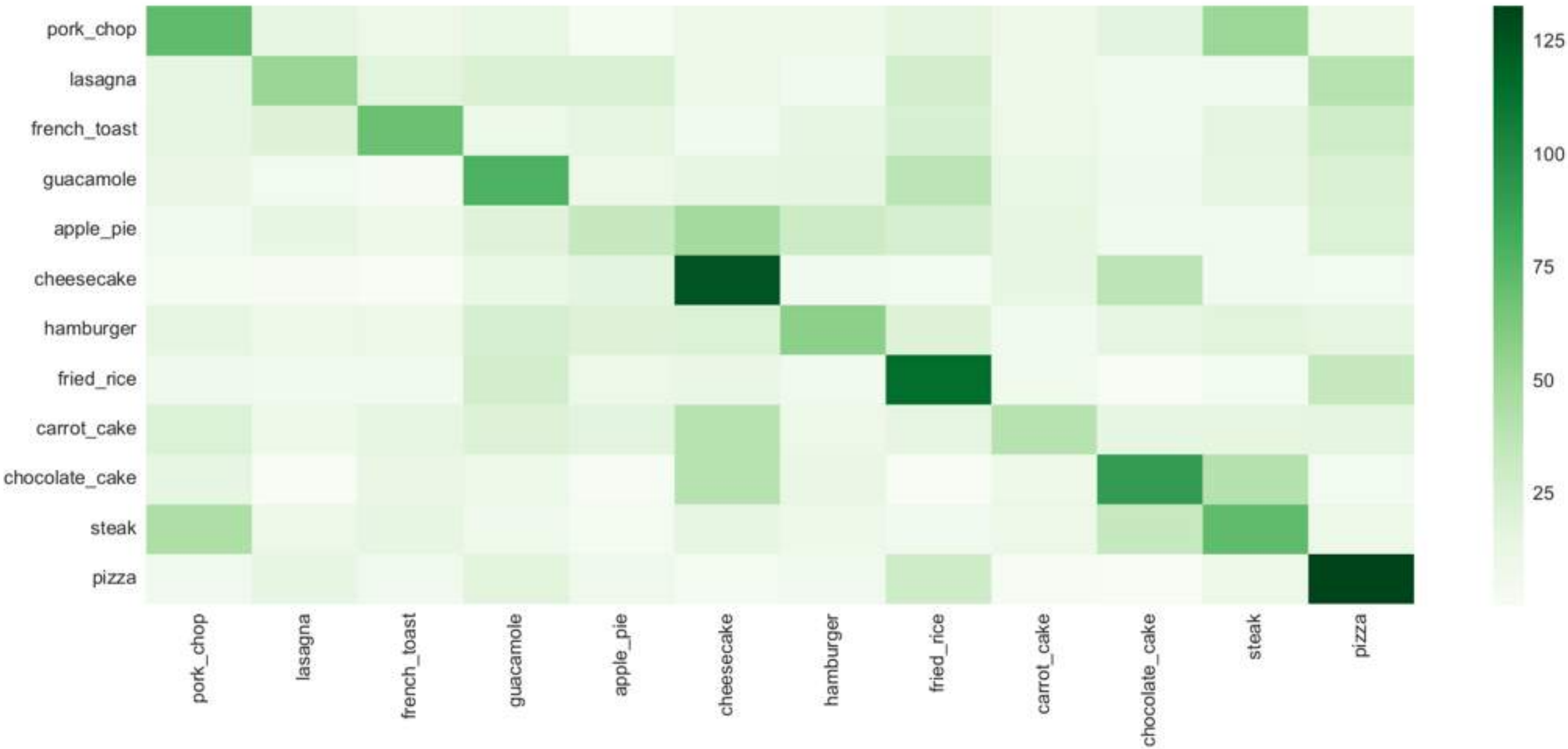
Chain all features together into one vector (..1,280 features..)

Random Forest
+
Grid Search for Hyperparameter Optimization

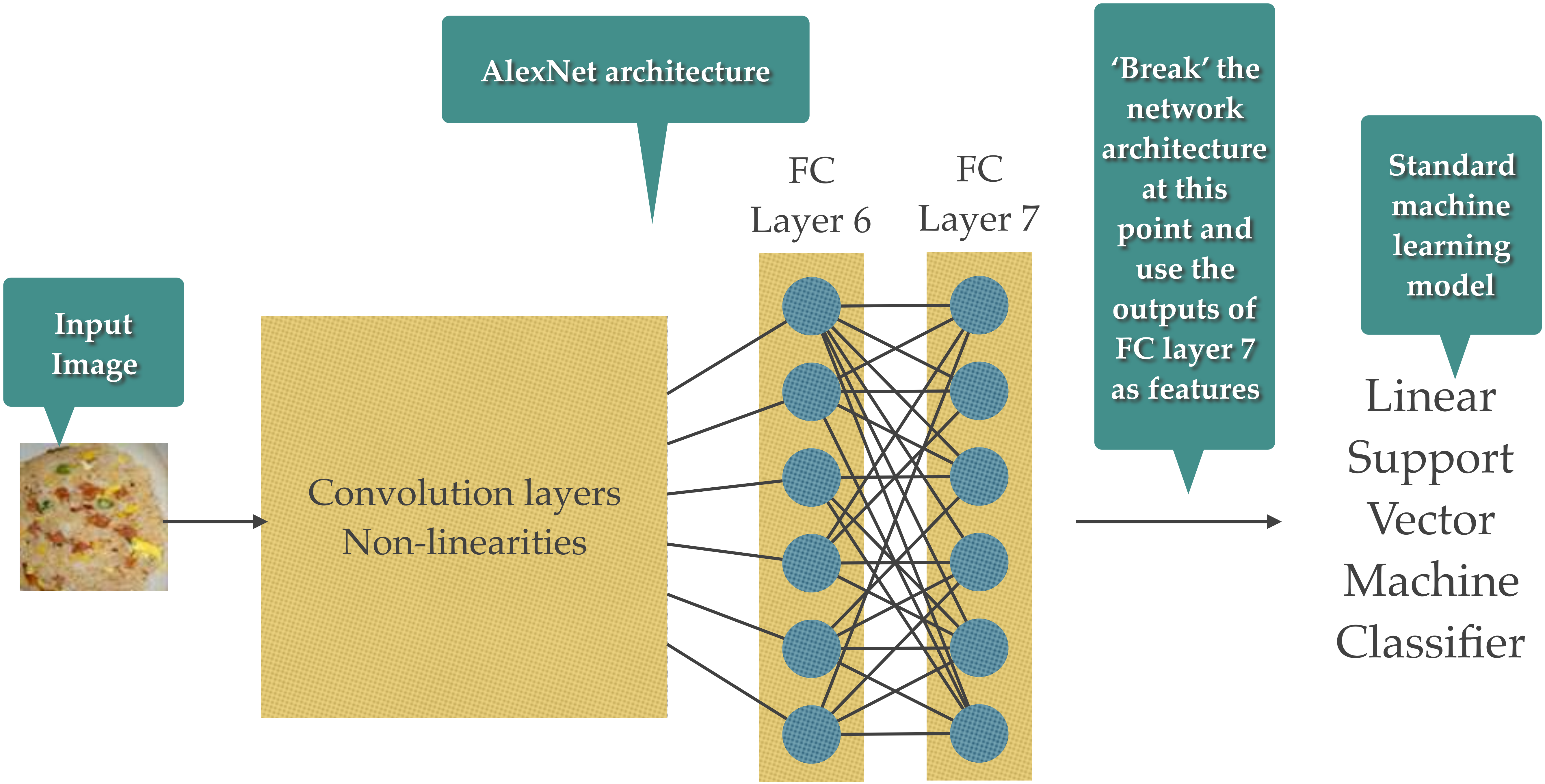
Per-class results



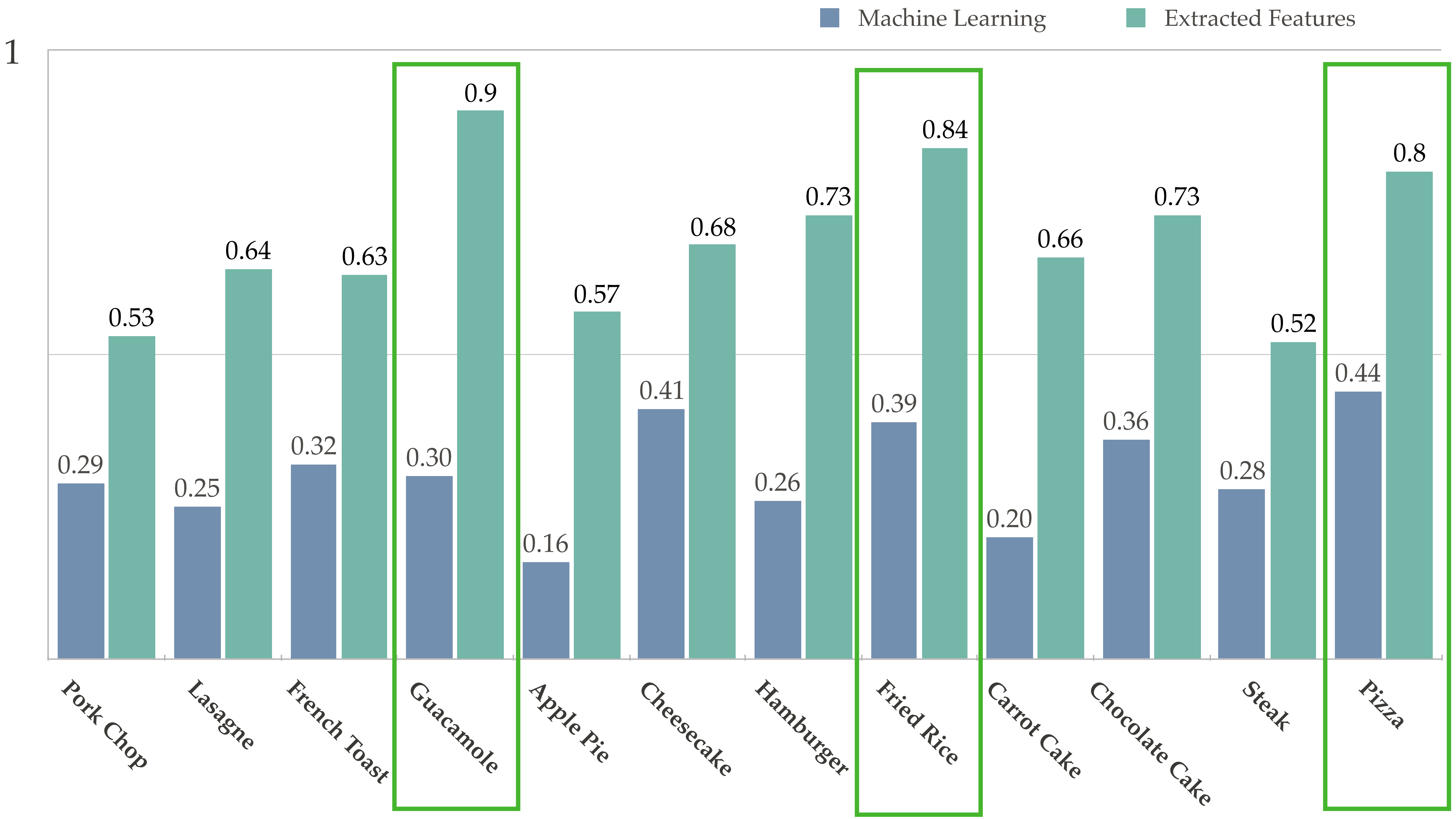
Confusion Matrix



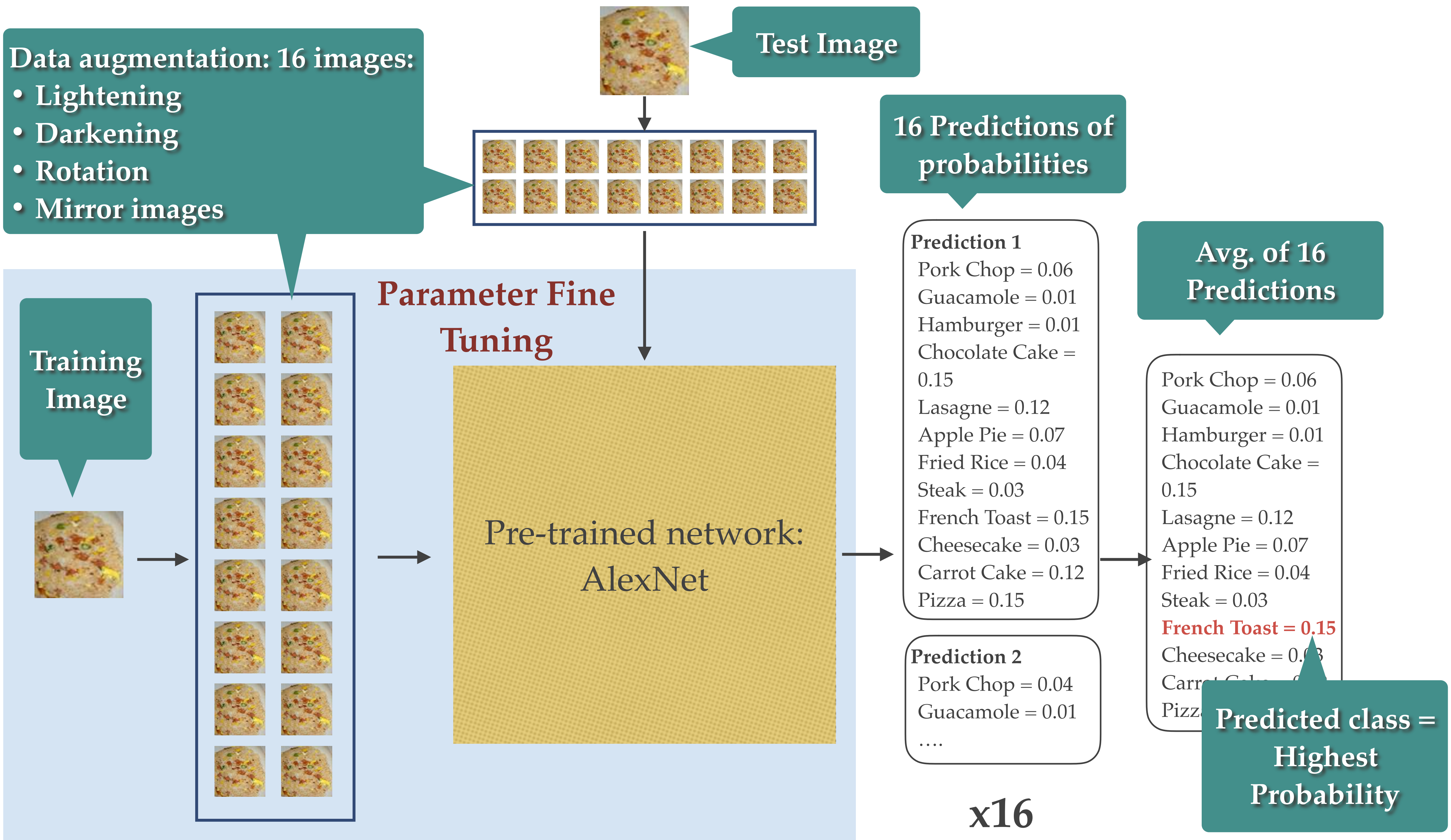
Approach 2: Feature extraction from a Convolutional Neural Network



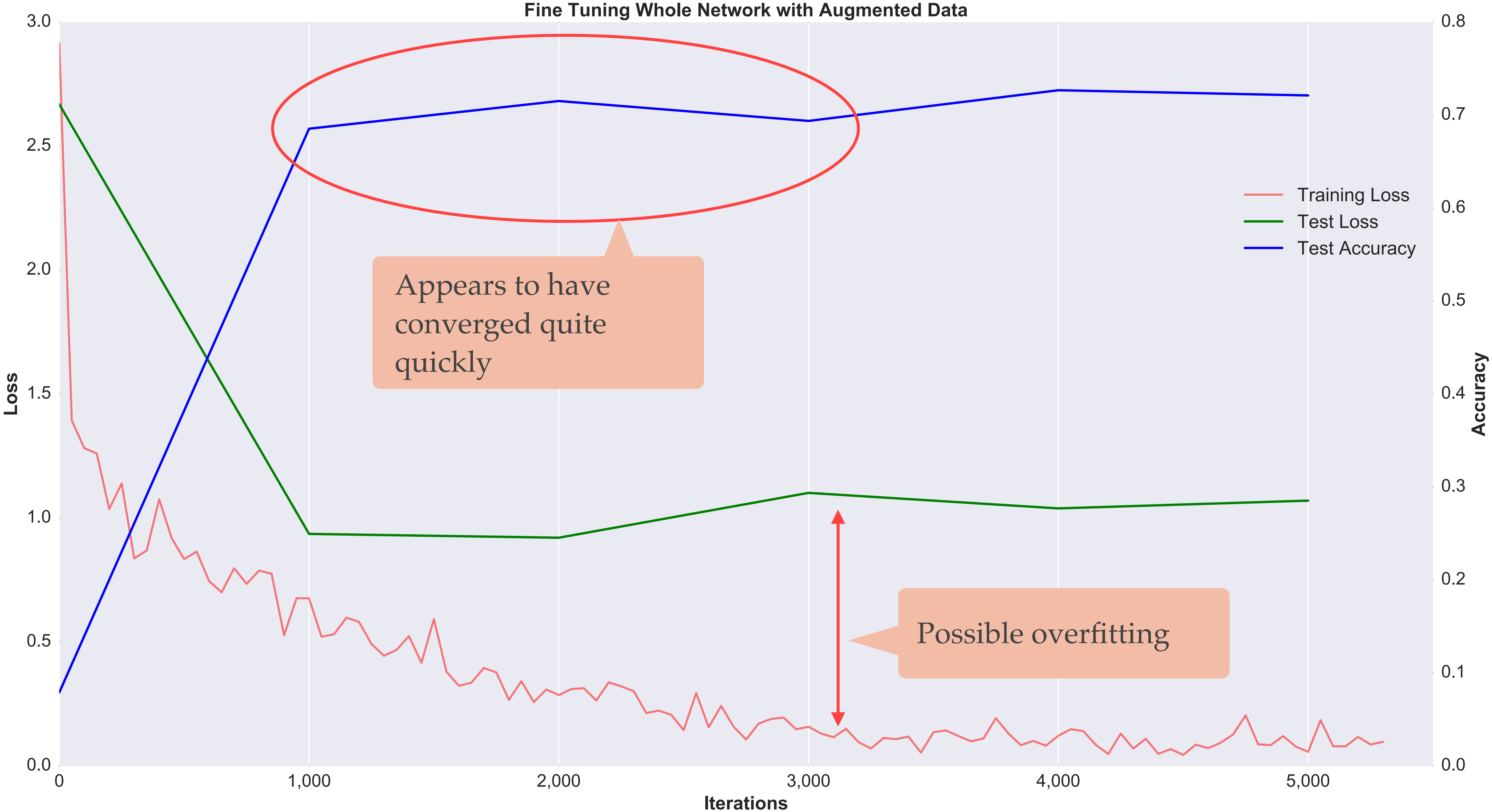
Per-class results



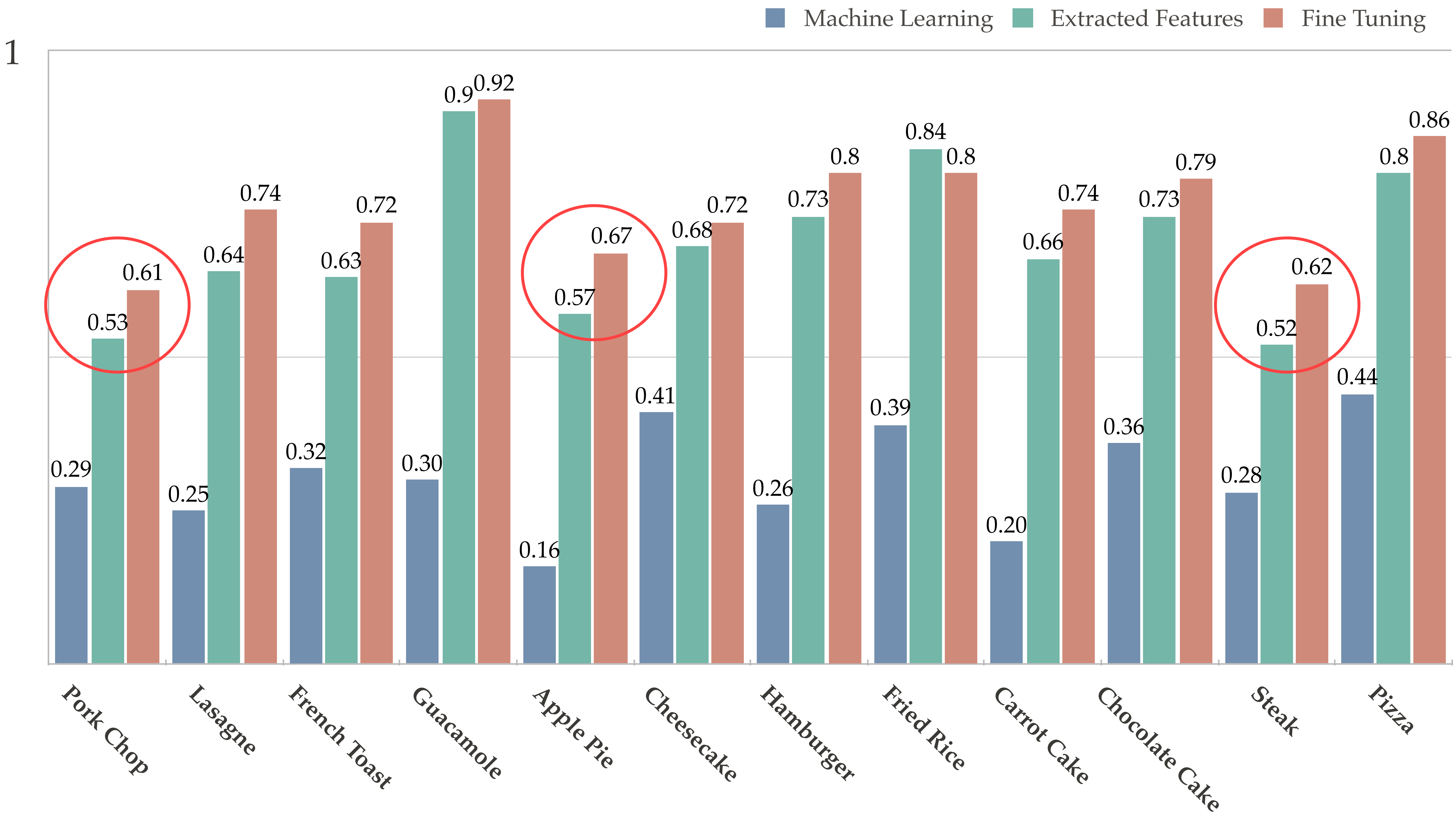
Approach 3: Fine Tuning a Convolutional Neural Network



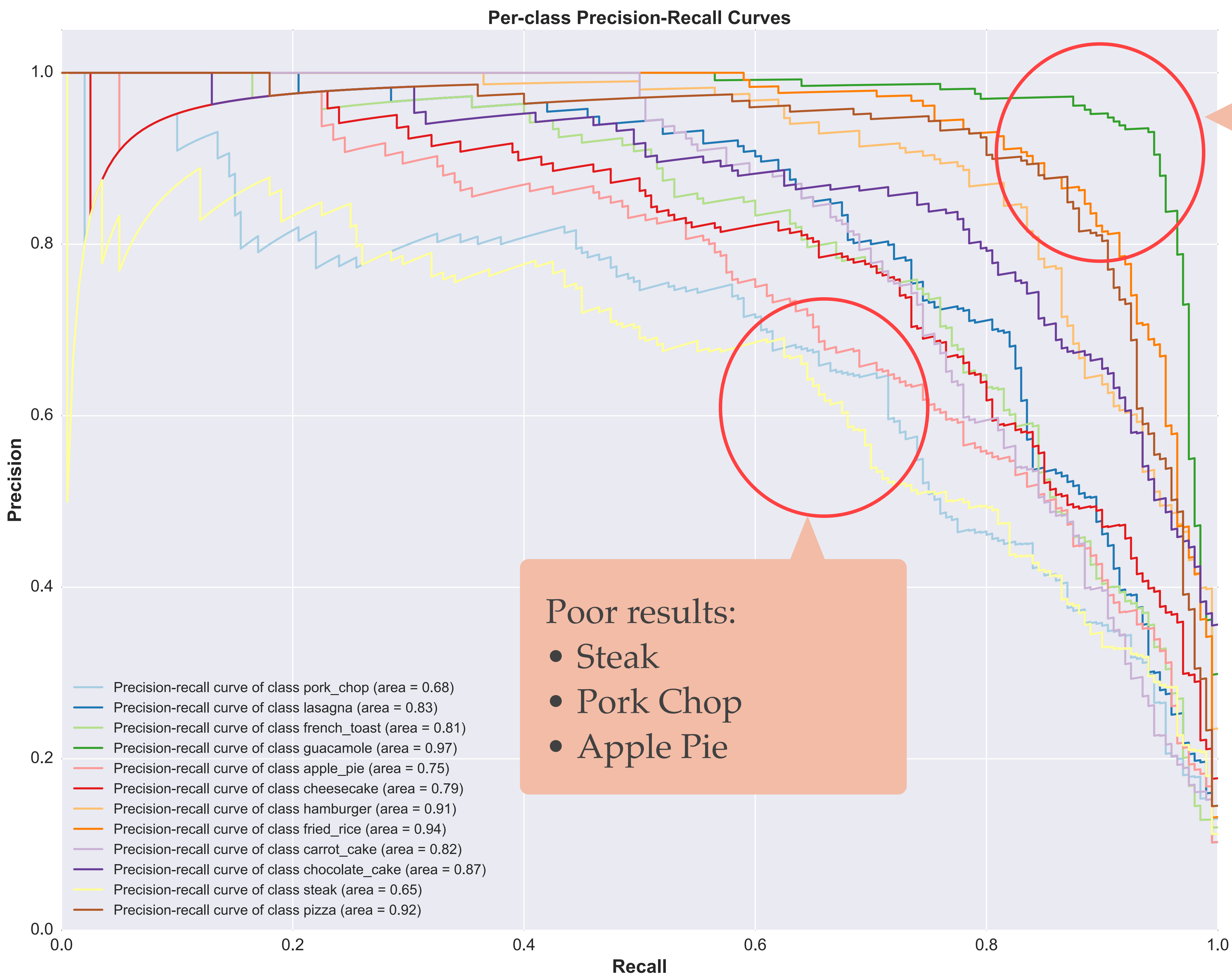
Training Curve



Per-class results



Precision-Recall



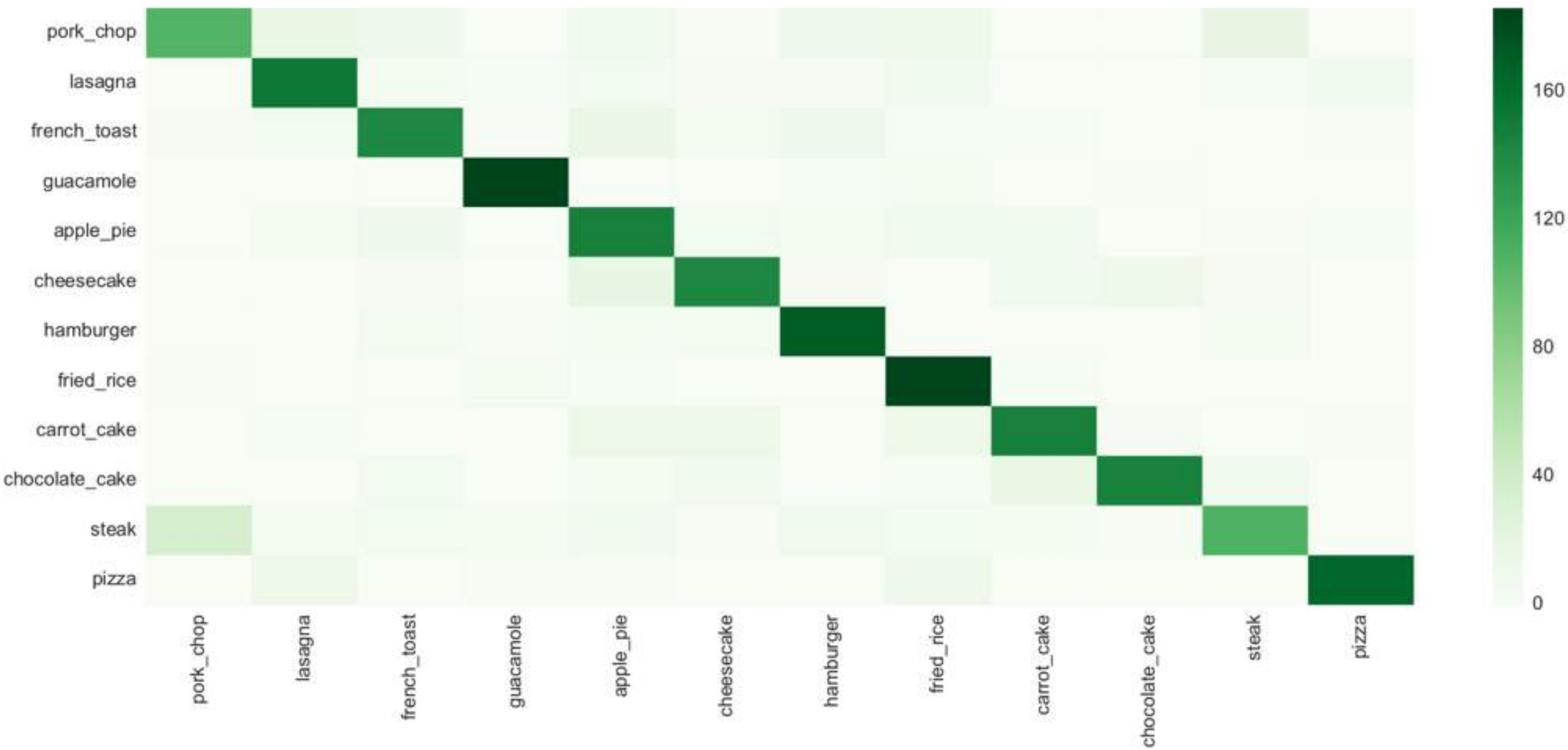
Good results:

- Guacamole
- Pizza
- Fried Rice

Poor results:

- Steak
- Pork Chop
- Apple Pie

Confusion Matrix



Fail Cases Examples

Steak Predicted as Pork Chop

Pork Chop Predicted as Steak

Future Work for Optimisation

- Fine-tuning hyper-paremeters e.g., dropout rate
- Increase training batch size (currently 150)
- More data augmentation
- Fine tune more recent models e.g., VGGNet, GoogleNet
- Ensemble of fine-tuned models

Machine Learning vs CNNs for Image Classification

	Machine learning	CNNs
Classification Accuracy		
Training Speed		
Testing Speed		
Ease of getting started		
Resources required		
Feature Selection		
Overall		

- Able to test 40+ models quite quickly
- Makes you think more about underlying images & their content
- Ultimately, not accurate enough

- Harder to get going and needed to use external computing resources
- But once started, training models was easier than expected
- Increase in accuracy makes up for everything else

Note: Personal opinions based upon experience with this project

Recommendations

1. Expand the model to include all 100 food categories from the existing dataset.
2. Seek to increase the number of images by looking for other sources of data.
3. Invest more time in optimising the model
4. Consider a pilot based on using a smaller set of 10-15 consolidated food categories.