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

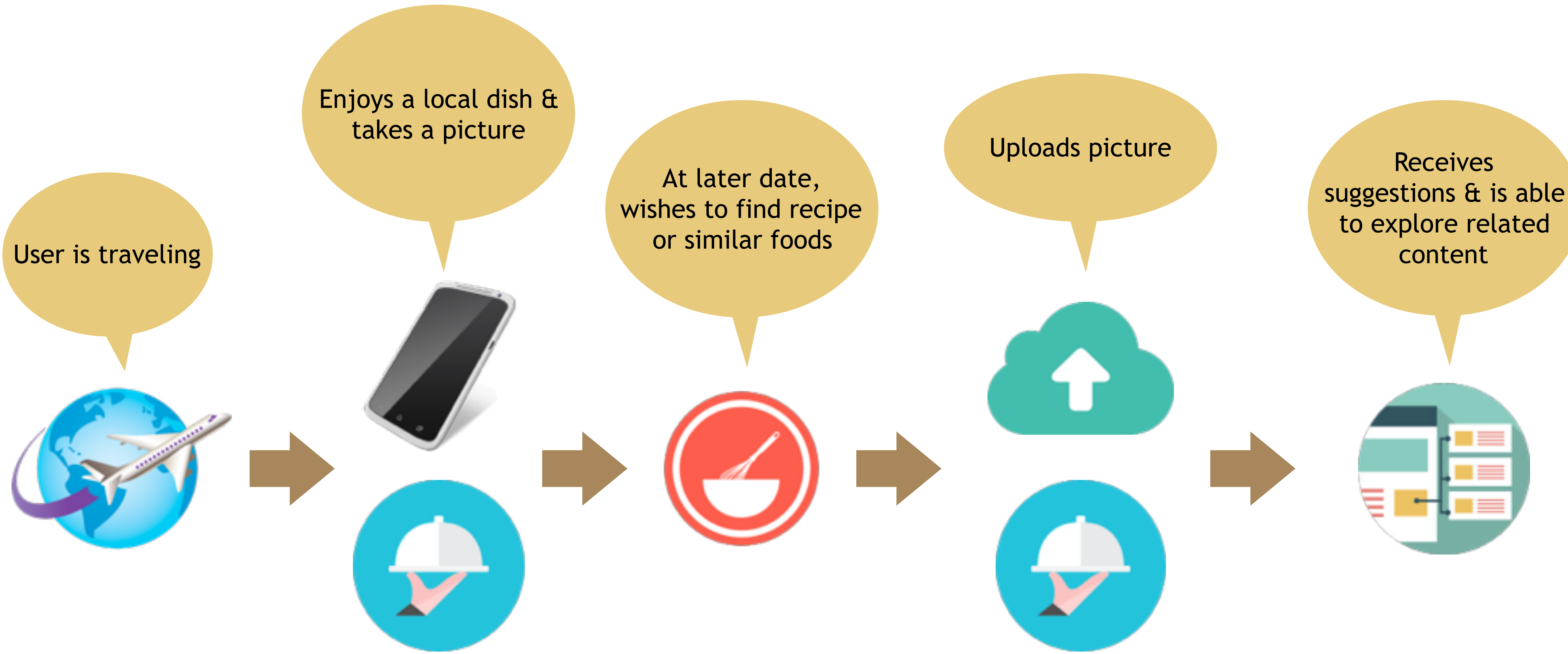
Classified Food item(s)

Find relevant recipes

Estimate calorie count

Identify ingredients

Example use case



Problem and approach

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

1. Start with set of pre-labelled food images

2. Train a classification algorithm using the pre-labelled images; Classification output should be one class per image

- "Traditional" Machine Learning Algorithms
- Deep Learning & Neural Networks

3. Test the accuracy of the algorithm using 'unseen' labelled imaged

F1-Score

Experiment with different models & compare results

Requirements for algorithm:

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

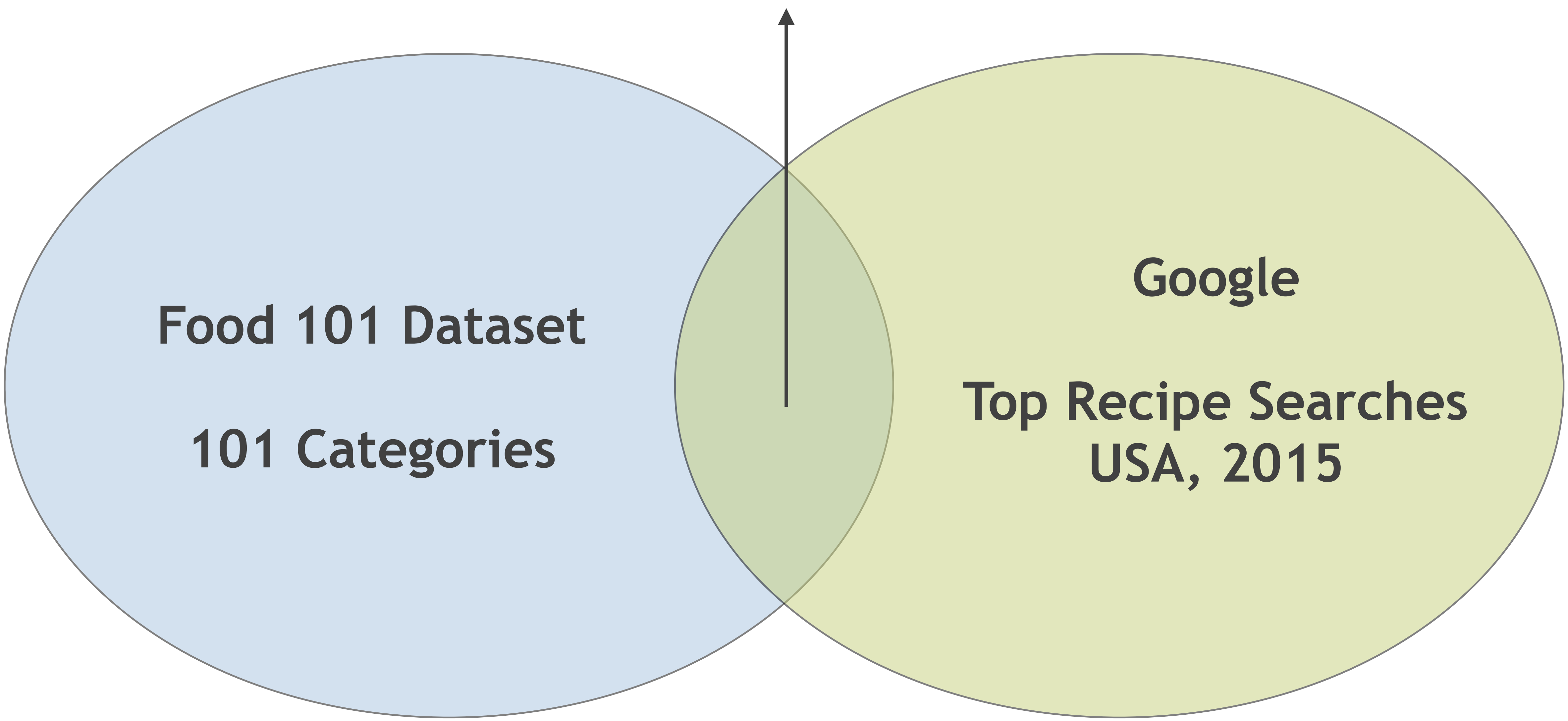
Note: approach is based on classification of whole image

The dataset

- Food-101 Data Set from the ETH Zurich Computer Vision Laboratory
- 101 Categories
- 1,000 images per category
- 6GB in total

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

Approach 1: Machine-learning models + features

Best F1 Score: 0.33

- Classifier not good enough
- Move on to Deep Learning

Models:

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

\+

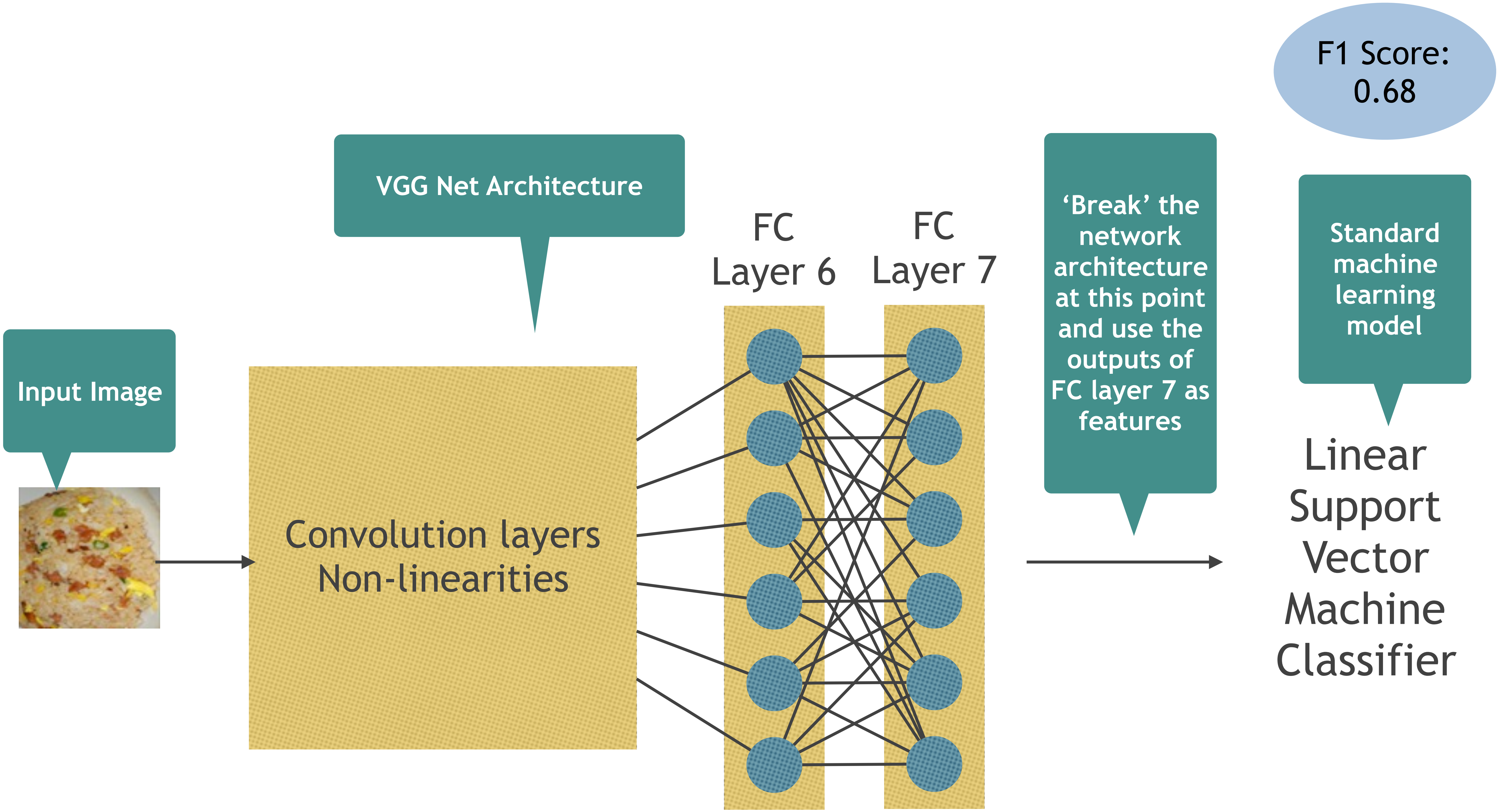
Features:

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

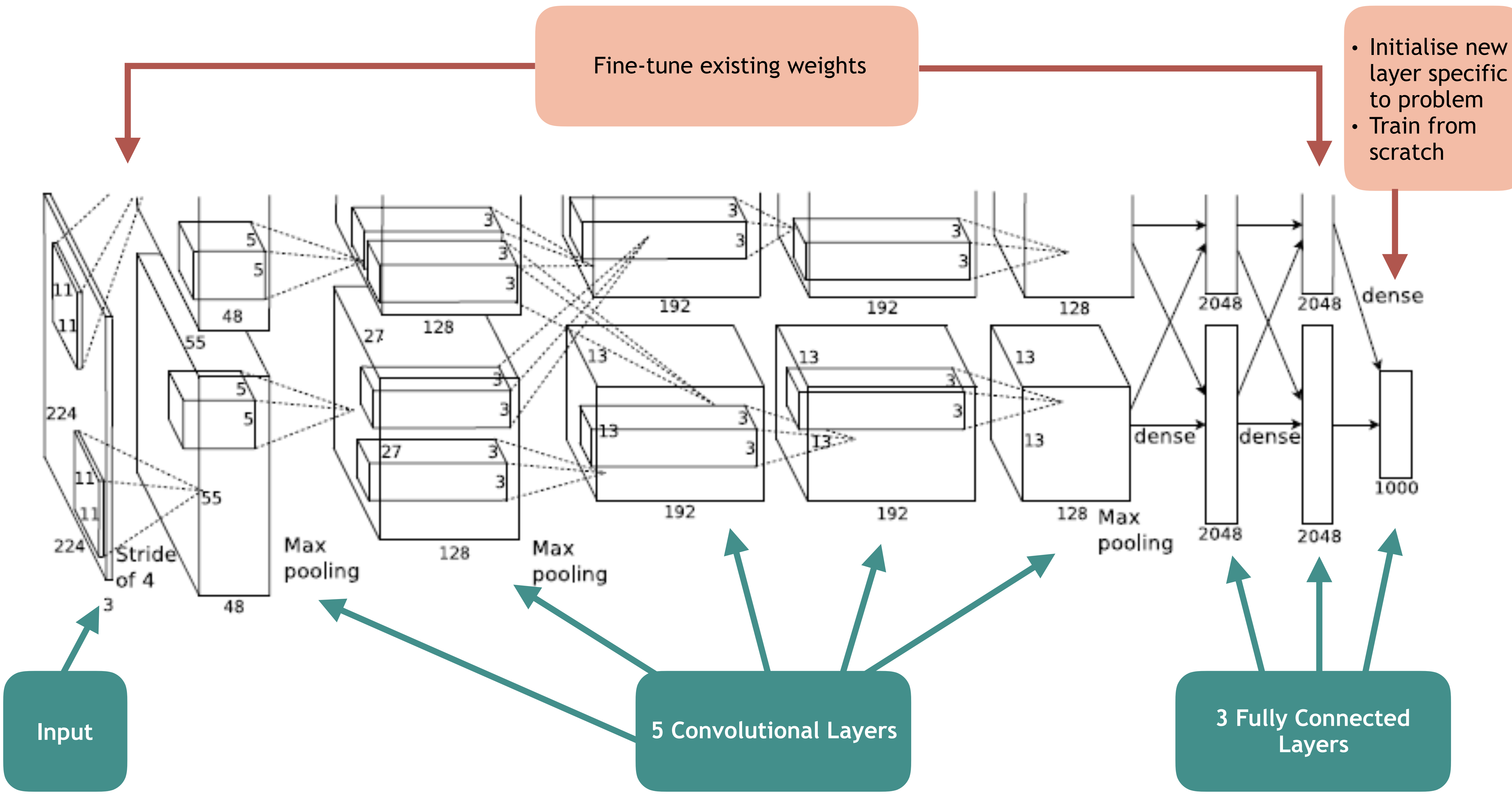
"Supervised" learning algorithms

Manually chosen features

Approach 2: Feature extraction from Convolutional Neural Network (VGG Net)

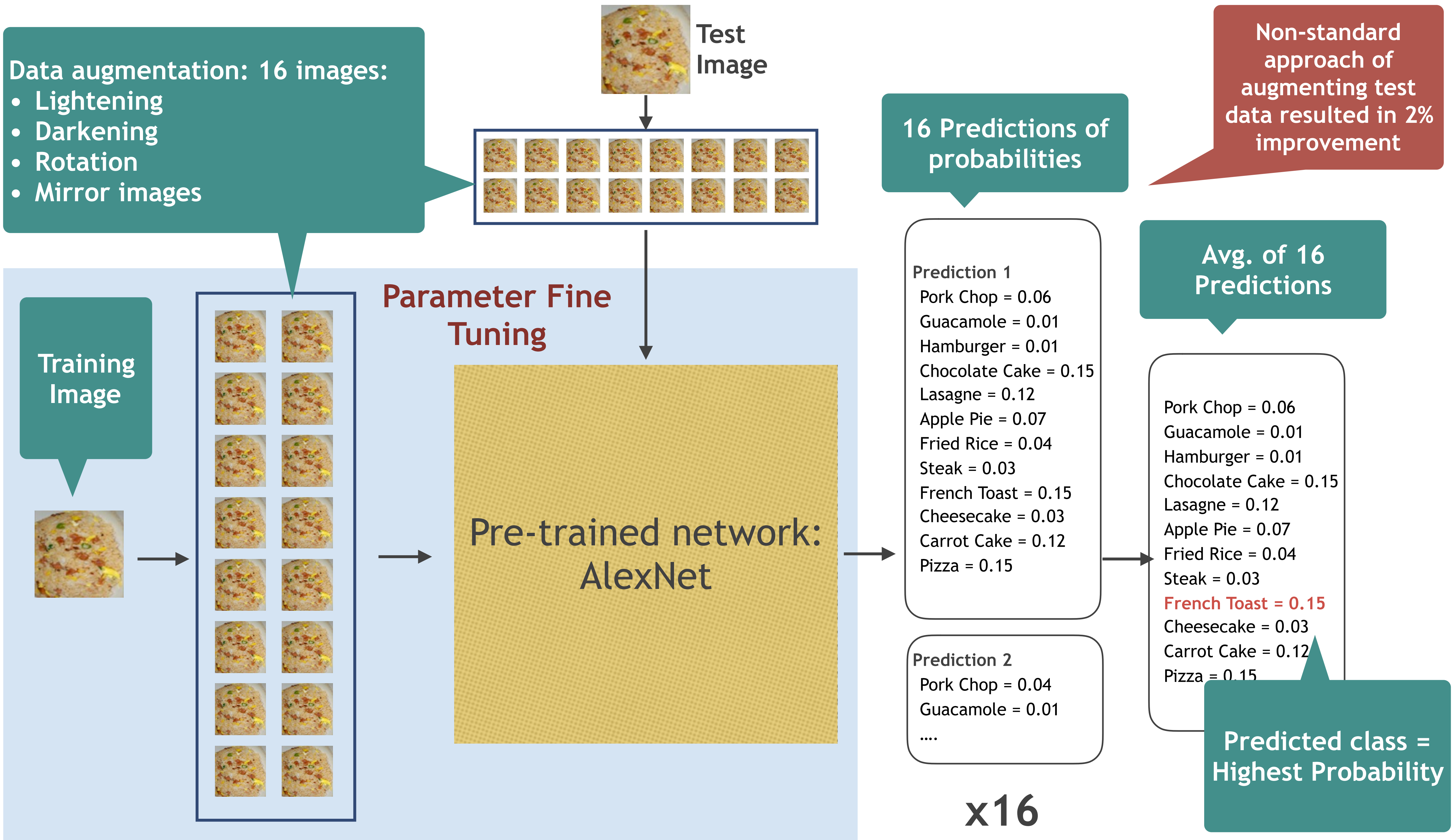


Approach 3: Fine Tuning AlexNet (CNN)

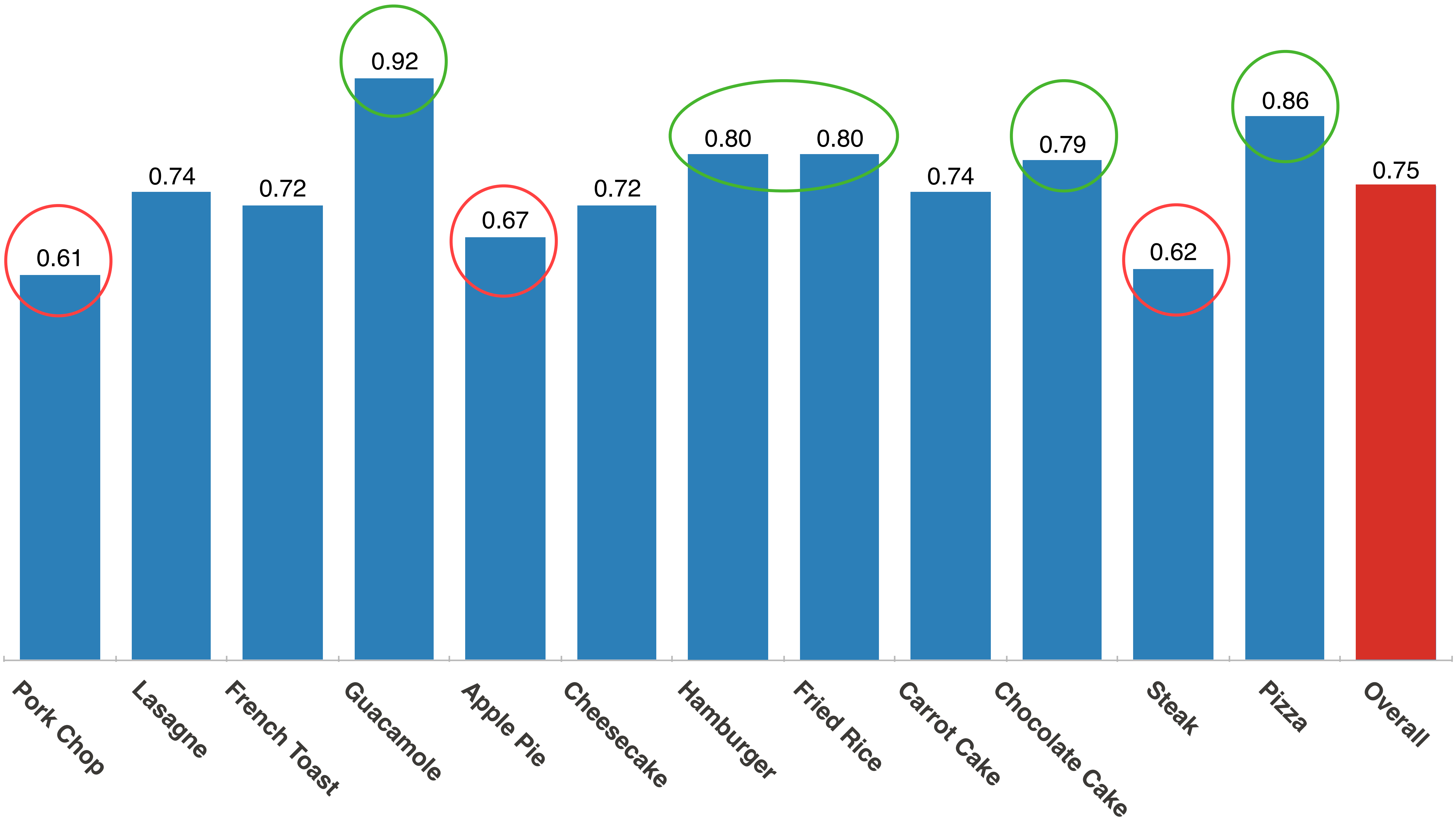


Source: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Fine Tuning Procedure



F1-scores


0.61
0.74
0.72
0.92
0.67
0.72
0.80
0.80
0.74
0.79
0.62
0.86
0.75
Pork Chop
Lasagne
French Toast
Guacamole
Apple Pie
Cheesecake
Hamburger
Fried Rice
Carrot Cake
Chocolate Cake
Steak
Pizza
Overall

Future Work for Optimisation

- Fine-tuning hyper-parameters 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

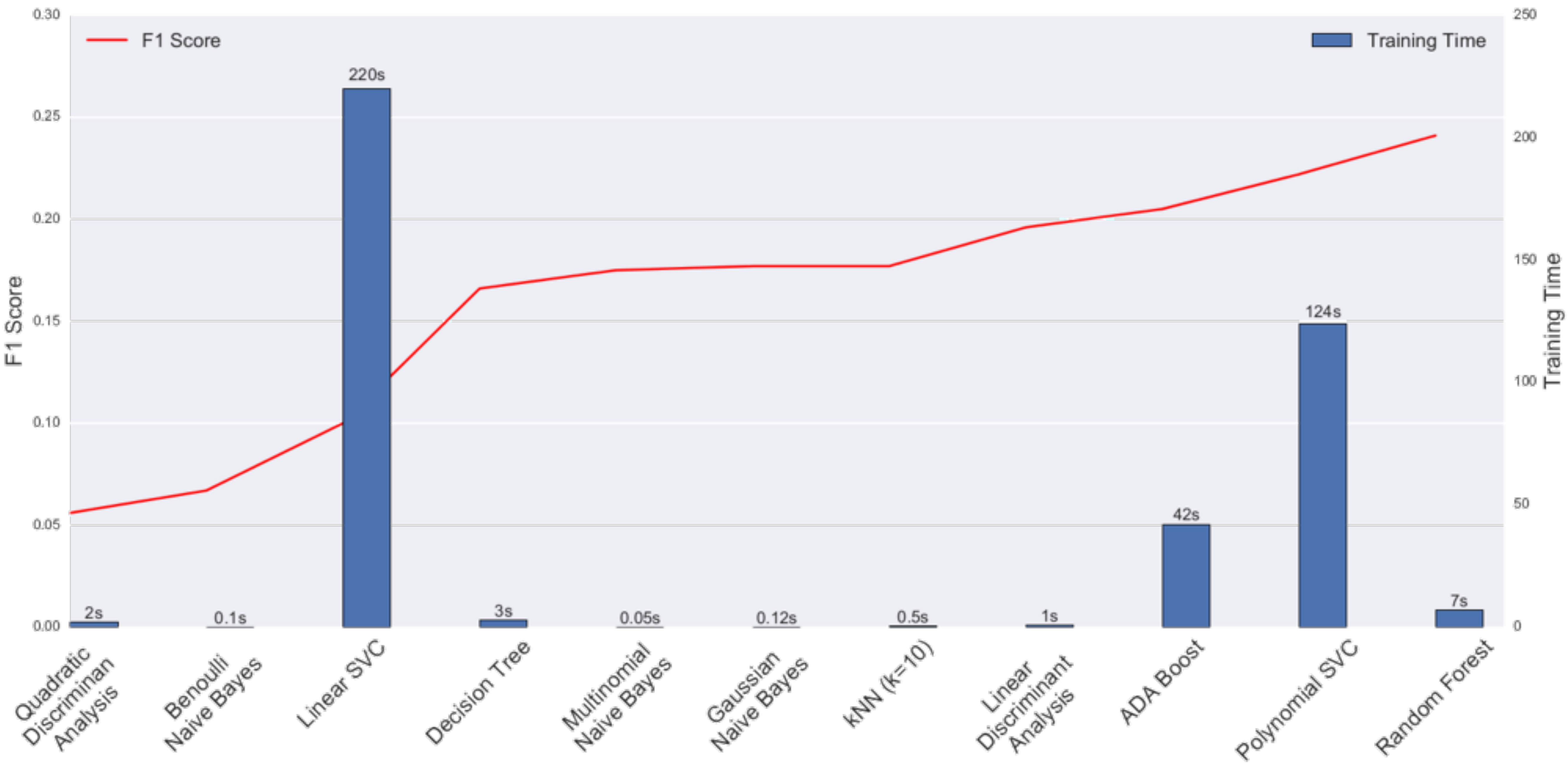
Recommendations

1. Expand the model to include all 101 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
5. Use more sophisticated techniques such as object detection & semantic segmentation for identifying multiple food-types in single image

APPENDIX

Differences in RGB histograms between image categories

Comparison of Machine Learning classifiers

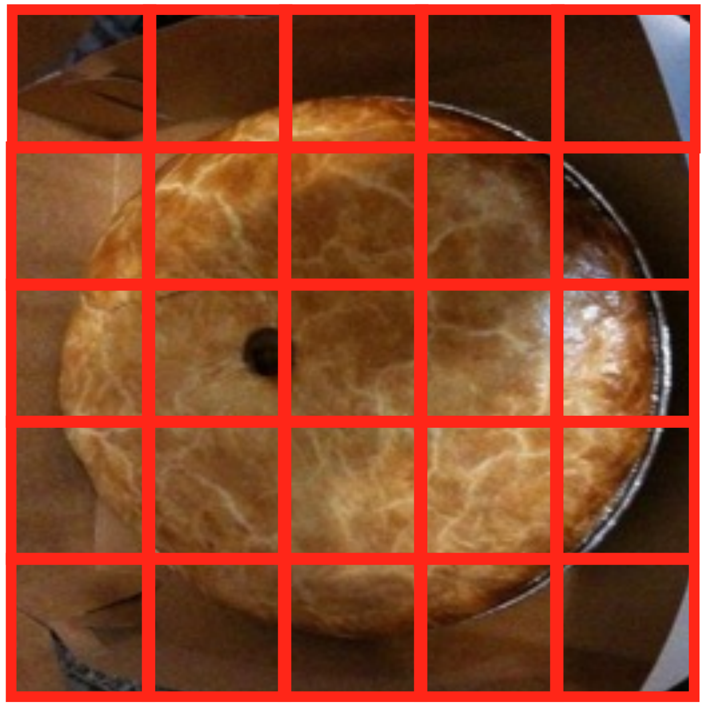


Best machine learning procedure

Overall F1 score 0.33 →

- Classifier not good enough
- Move on to Deep Learning

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



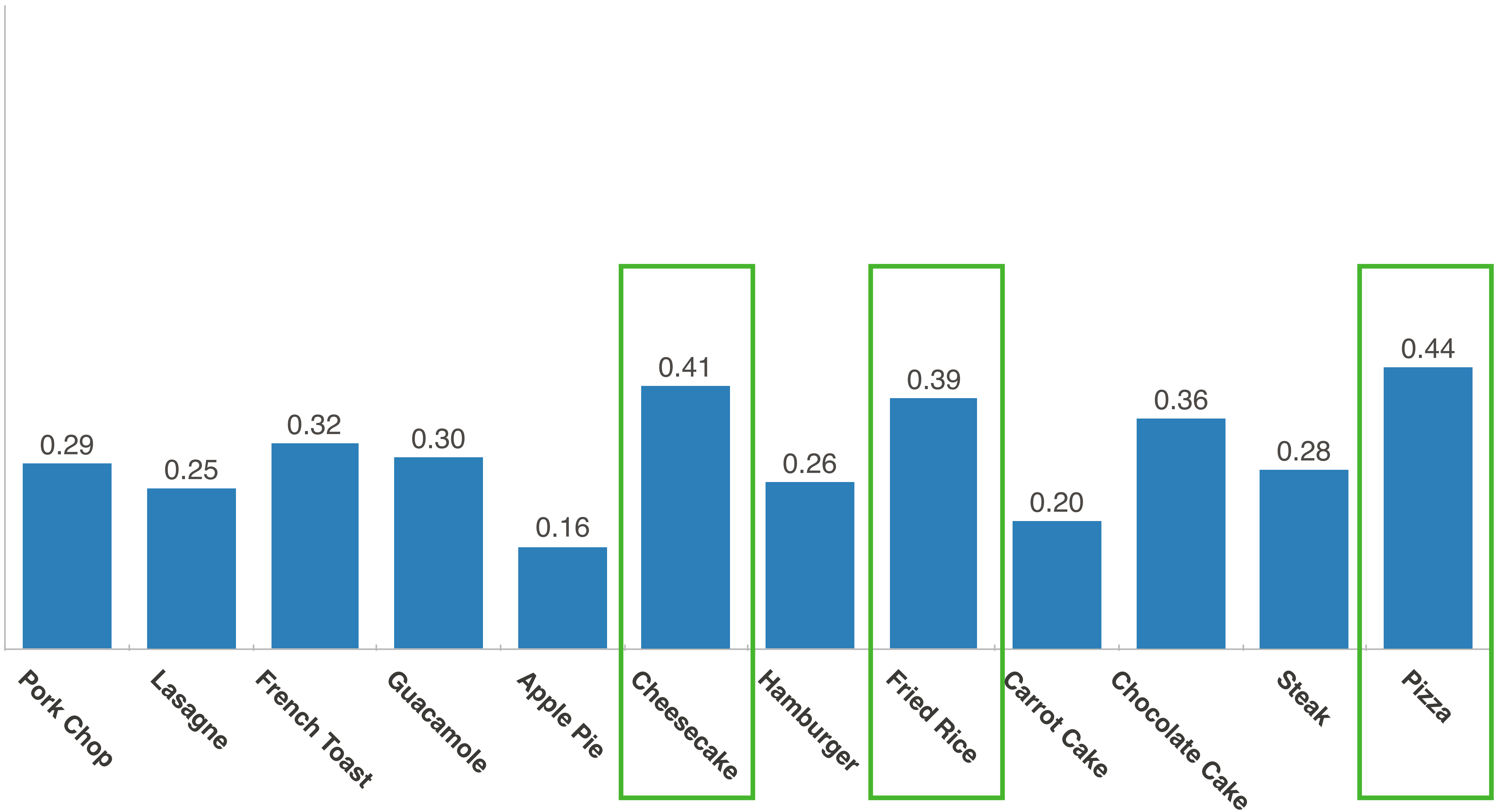
For each cell calculate:

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

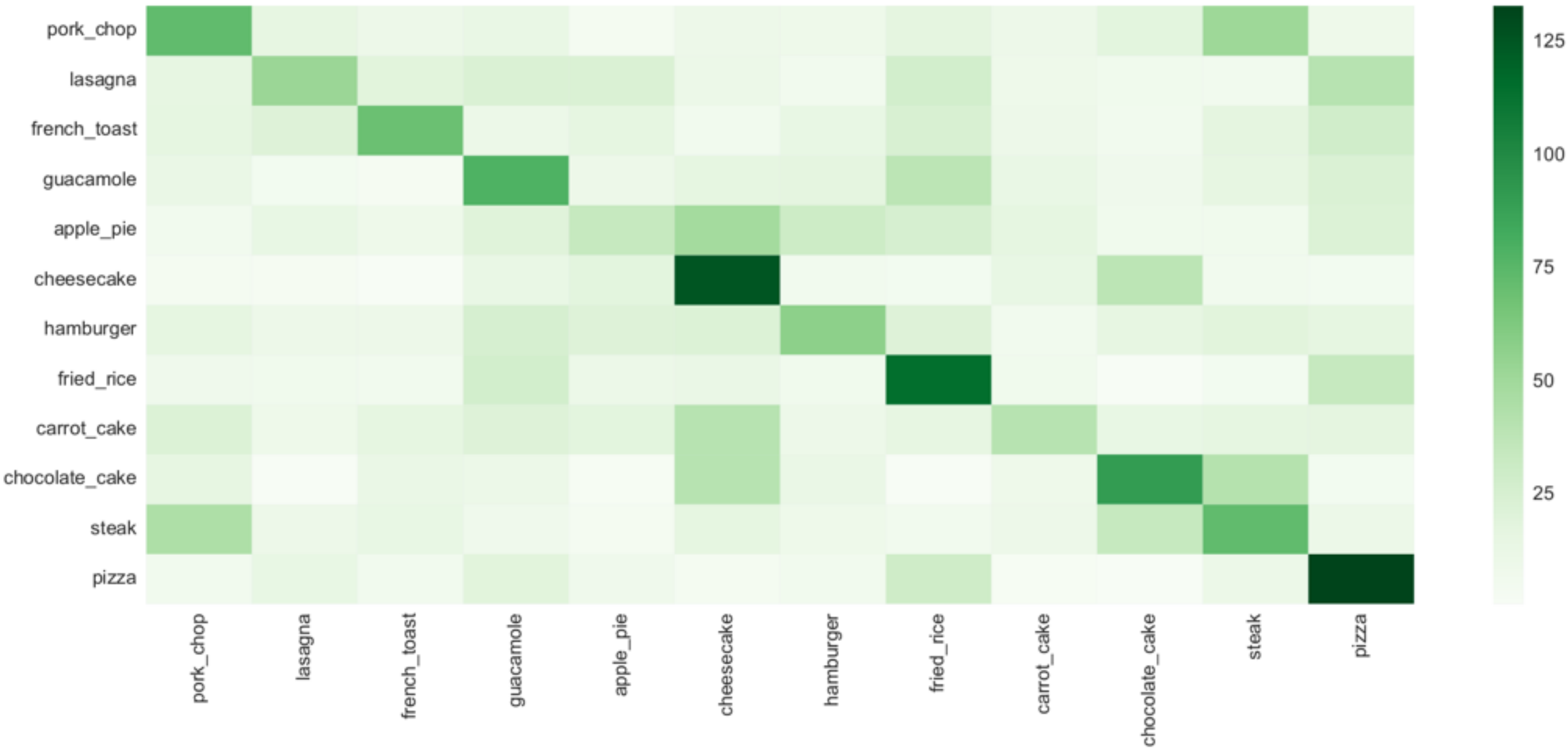
→ Chain all features together into one vector (..1,280 features..)

→ Random Forest + Grid Search for Hyperparameter Optimization

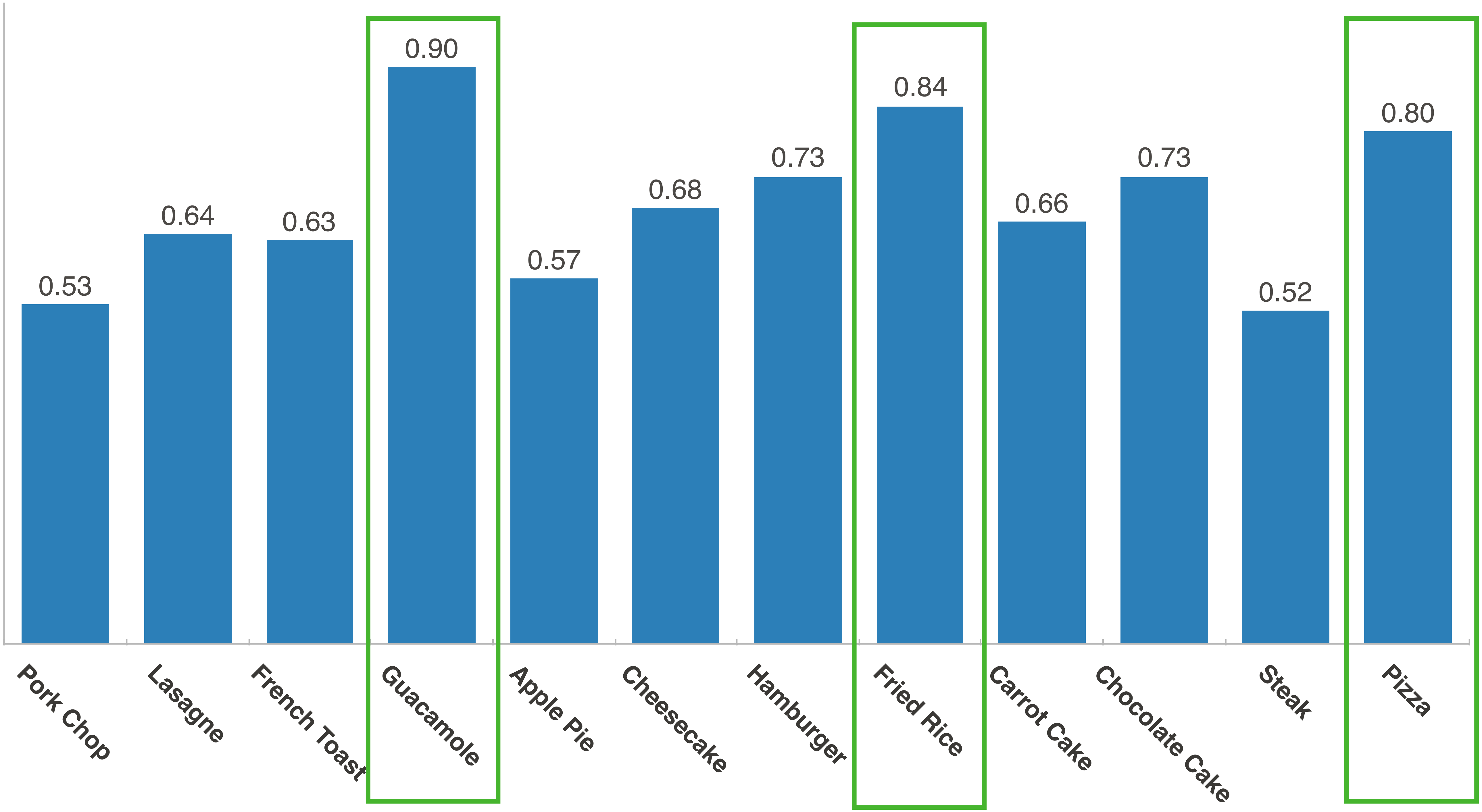
Machine Learning: F1 Scores



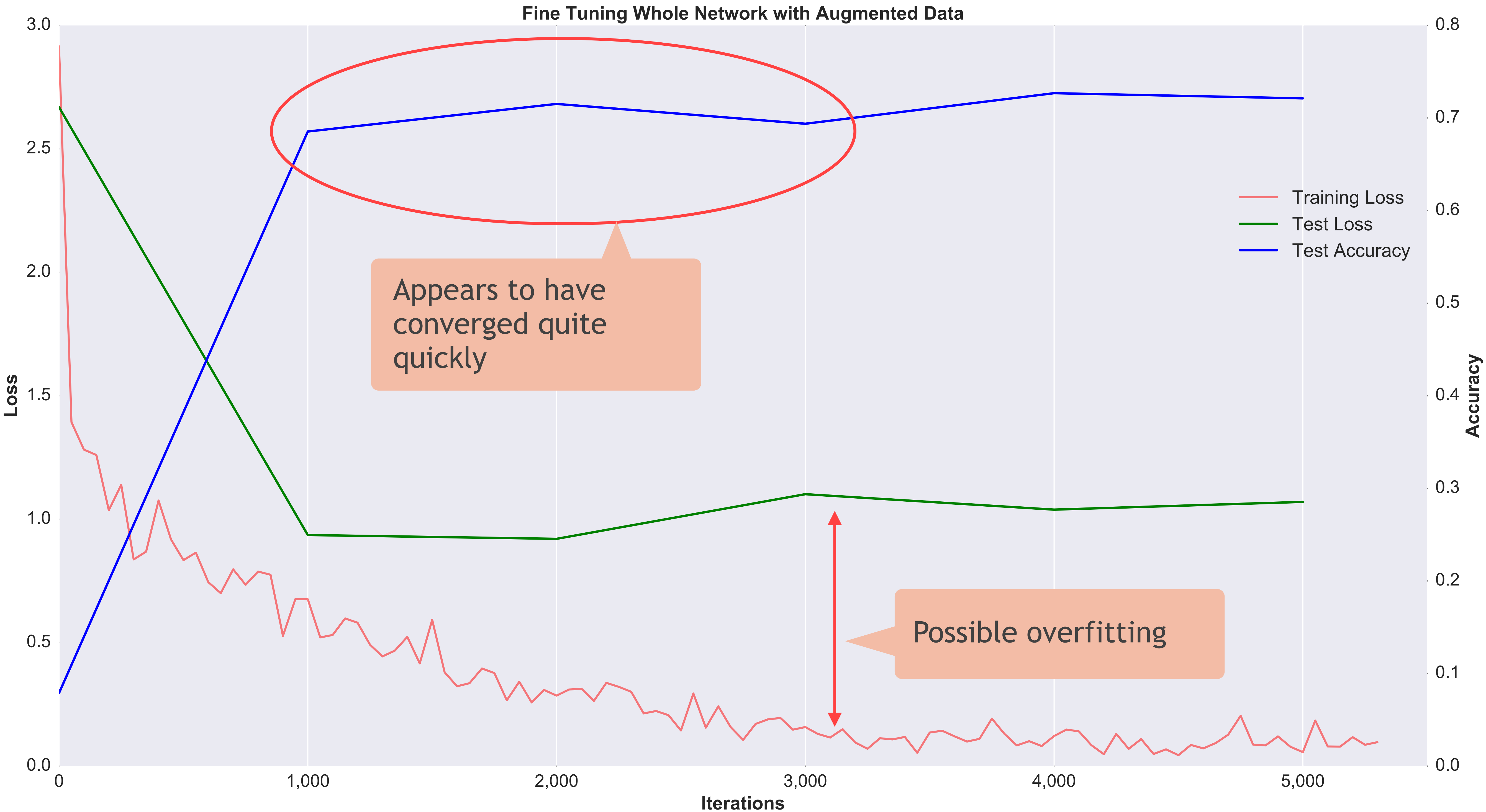
Machine Learning: Confusion Matrix



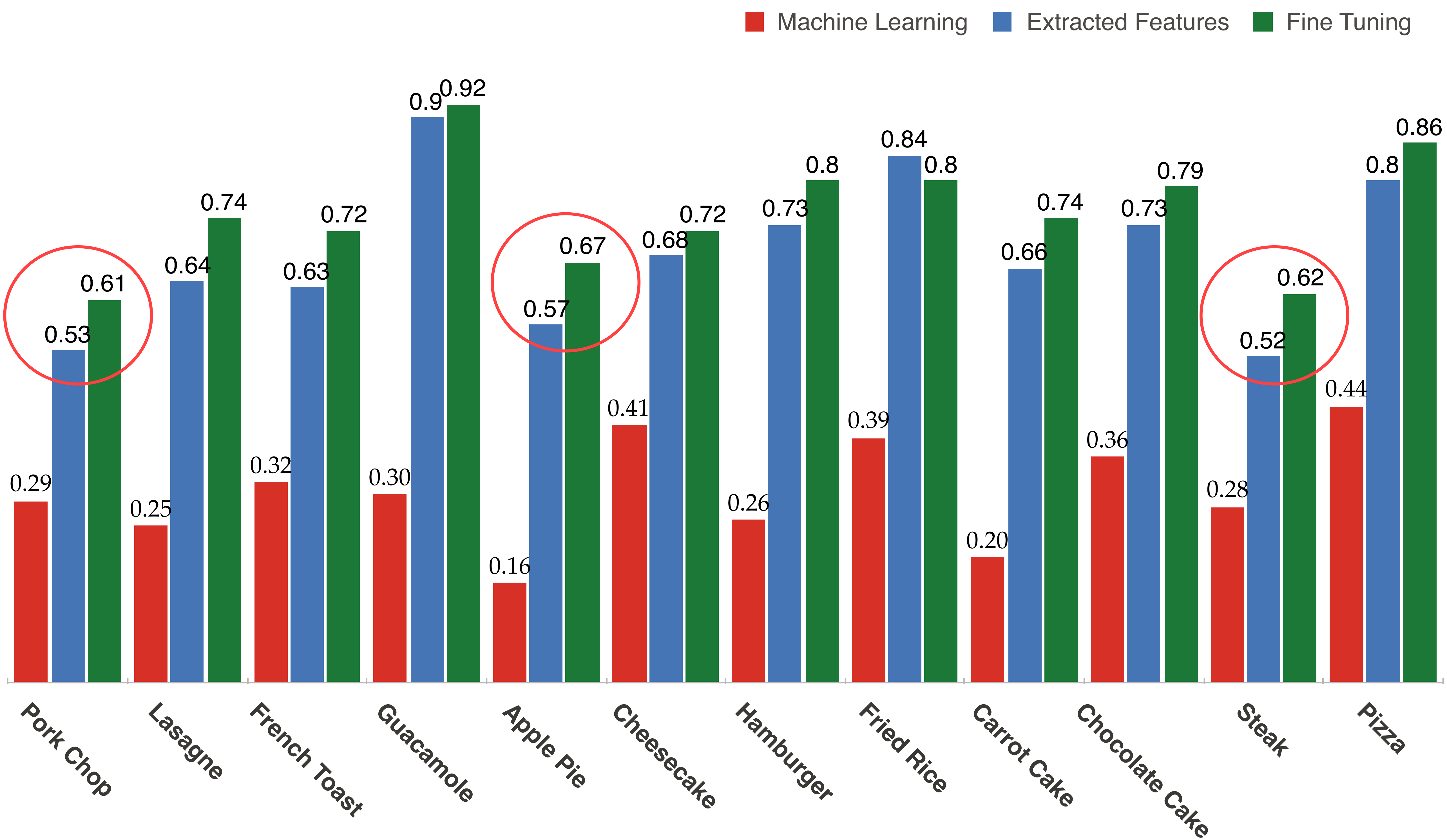
Feature Extraction: F1 Scores



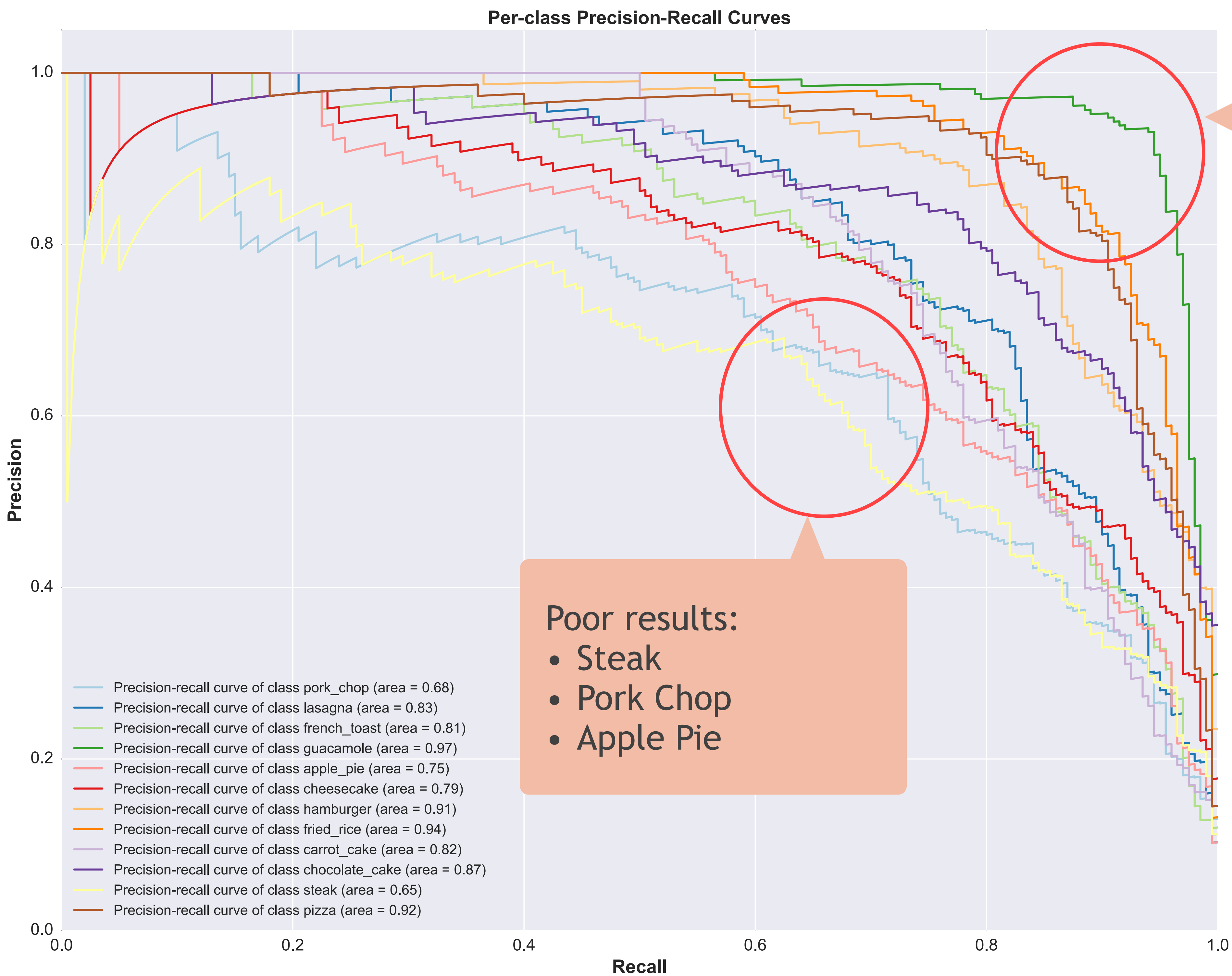
Fine Tuning: Training Curve



Per-class results: Comparison of methods



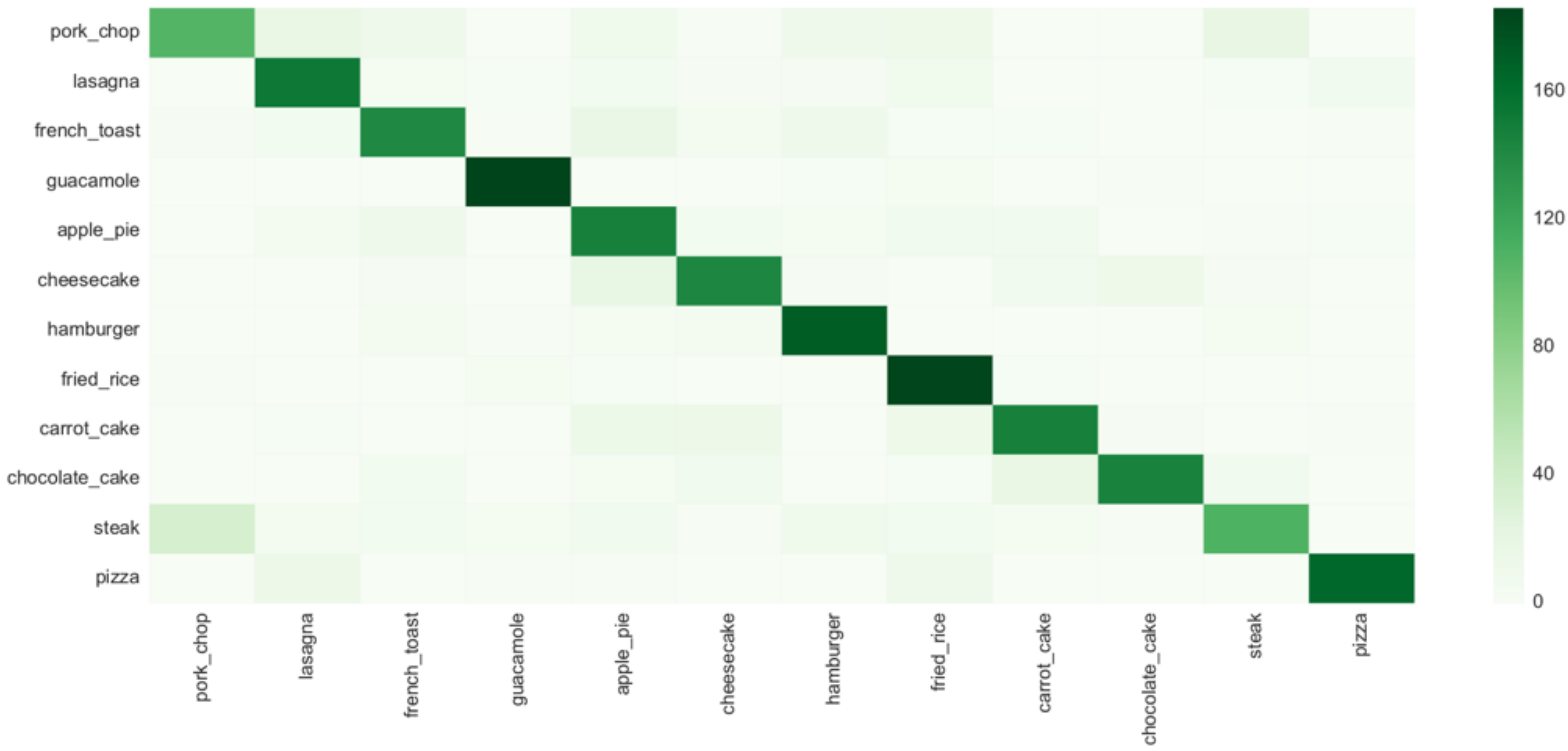
Fine Tuning: Precision-Recall



Good results:
- Guacamole
- Pizza
- Fried Rice

Poor results:
- Steak
- Pork Chop
- Apple Pie

Fine Tuning: Confusion Matrix



Fail Cases Examples

Steak Predicted as Pork Chop

Pork Chop Predicted as Steak

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
- Once started, training models easier than expected
- Increase in accuracy makes up for everything else

Note: Personal opinions based upon experience with this project