CRANFIELD UNIVERSITY

NOGARET BAPTISTE

AUTOMATED FOOD LOG ANALYSIS

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Computational and Software Techniques in Engineering

Master of Science
Academic Year: 2015–2016

Supervisor: Dr RÜGER Stefan
July 28, 2016

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Computational and Software Techniques in Engineering

Master of Science

Academic Year: 2015–2016

NOGARET BAPTISTE

Automated food log analysis

Supervisor: Dr RÜGER Stefan
July 28, 2016

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science.

Declaration of authorship

Abstract

Type your abstract here.

Keywords

Keyword 1; keyword 2; keyword 3.

Contents

List of Figures

List of Tables

List of Abbreviations

BoW	Bag of Words
CNN	Convolutional Neural Network
LBP	Local Binary Pattern
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

Acknowledgements

I am really grateful to Dr. Stefan Rüger, my supervisor for the project, to have proposed this subject. His guidance and valuable advice were particularly helpful to realise the thesis.

Moreover, I would like to thank the University of Technology of Compiègne for giving me the opportunity to study one year in Cranfield University. I would also like to thank Cranfield University for its facilities.

I would like to express my gratitude to M. Kazu Shimoda and Pr. Keiji Yanai of the University of Tokyo that made their datasets available and provided enlightenments and further details on their work.

Chapter 1

Introduction

Over the last few decades, the rate of obesity and overweight people in the World has greeatly increased. As presented for the UK case in the figure 1.1, the obesity rate has increased by 12 % between 1980 and 2013, and by 13 % for the overweight rate. It is forecasted by the World Health Organisation to continue to grow.

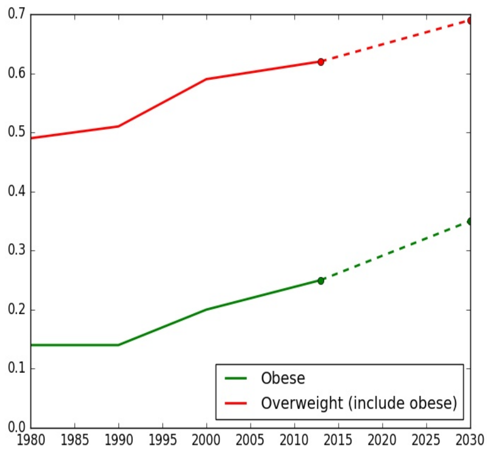

Figure 1.1: Obesity and overweight rate of the adult population in the uk between 1980 and 2030

Being "overweight" is defined as having a Body Mass Index (BMI) – a person's weight in kilograms divided by the square of his height in meters (kg/m^2) – of between 25 and 29.9, and "obese" by a BMI of 30 and above.

As stated in [1]), obesity is strongly associated with several major health risk factors such as stroke, high blood pressure, type 2 diabetes and high cholesterol.

Diabetes: - fast growing (current ... in to ... in) with forecast for ... to be ... - lead to high mortality - treatment cost. [2]: in 2010: 12 % of the total whorlwide health expenditure is spent on diabetes and will continue to increase.

Combination of drugs and food intake control have shown great results

Main reason: junk food: easily found, cheap.

One of the best way to fight it: watch over what we eat. Associated lifestyle changes and lose weight. It can also be used as a prevention tool for population at risk

studies such as [3] show the benefit of reporting its daily diet to lose weight and improve the quality of its foos intake

Also a way to ... eat disorders

Currently, manually ... self reporting, using paper diaries : tedious + time consuming + prone to errors (users tend to underestimate its intake as describe in [4]) + need a trained patient

At the same time, improvement of the classification methods. Imagenet, a 1000 classes and more than 1,2 million images dataset example on Image Net results [5]. Every year since 2010 Numerous institutions (university, tech companies) are participated As described in figure 1.2, the mean error for each class for classification and localization has been greatly reduced between 2010 and 2014

Recently: proposition automize it. With the widespread use of smartphone, people can easily take pictures of a good quality. People are already taking picture of their food and posting them on website such as Food Gawker, Instagaram, Flickr, Yelp or

That's why, over the past few years, people ... automated it. Assist patient and their

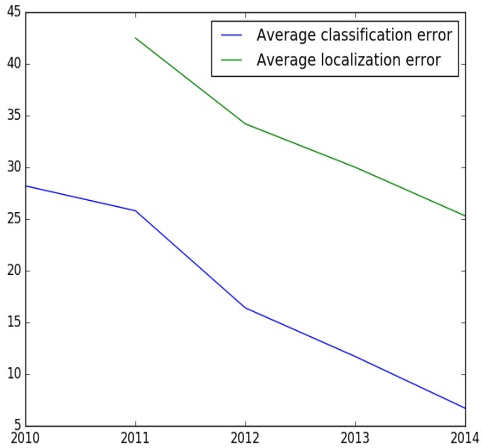

Figure 1.2: Average classification and localization error of the best results for different ImageNet challenges

medical personnel Extends the reach of care in a cost effective ways and counters some of the previous problem (still pb with the elder / people who don't have access to smartphone). Or using weerable device to automatically take picture

Part of the rise of e-healthcare / m-healtchare [6, 7]

food recognition: promising applications of image processing and machine learning. Estimate food intake and people's habit

Overall process: extract characteristic (possible features are invariant of the liminosity, orientation, scale, ...)segmentate, classify, get calori value or a simplified version (using for example the ... systems), keep log and being able to visualize it over the year

Feature description: key to achieve good object detection and image categorization

In this thesis: focus on the first two phases

Already have numerous challenges: large number of food items variation in appearance and shape different way to server it environmental condition –> lead to a high interclass variability challenging task for the human

The organization of this thesis is as follow. In section ..., previous work is reviewed. In section ..., I explain and describe the dataset choosen, then the different image descriptors and classifiers used. In the next section, we present and discuss our results. Finally, in section ..., the limitation and possible future work is discussed.

Chapter 2

Previous work

Food localization process

circle detection if we make the asumption that the food is in a plate / bowl: 2.2 + 6.3 + [8]

color segmentation vs edge segmentation: 11.1 (very limited test)

DCNN: 3.2 + * + 9.1

Food recognition

Using SVM:

Local using BOW: 3.3 gloabal feature: Color and texture description: Spatial pyramid:

Mix of several features:

DCNN: *

Estimate food intake:

Food log Food Cam

General process

- divide the dataset in train and test Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called overfitting. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set. - learn and evaluate - feature description - choose of the classifier

presenting different channel representation RGB, gray, HSV

cite who use it first?

Feature detection - can use sift or surf - dense grid (cite why it is better)

Descriptor

clustering - k means

Chapter 3

Feature descriptors

3.1 Local binary pattern

Local binary pattern is a visual descriptor for texture composition of an image, first presented in 2002 in [9] (although the concept of LBPs were introduced as early as 1993).

The figure 3.1 represents an example of the LBP in which the LBP code of the center pixel (in red color and value 20) is used as a local intensity threshold : the neighbour pixels whose intensities are equal or higher than the center pixel's are labeled as "1"; otherwise as "0". Then, starting always from the same point, we can transform this binary string to decimal. In this example we start at the top-right point and work our way clockwise accumulating the binary string as we go along and obtain the value 24.

We adopt the following notation. Given a pixel $c = (x_c, y_c)$, the value of the *LBP* code of c is defined as:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p$$

where:

- p is a neighbour pixel of c and the distance from p to c does not exceed R. Thus, R

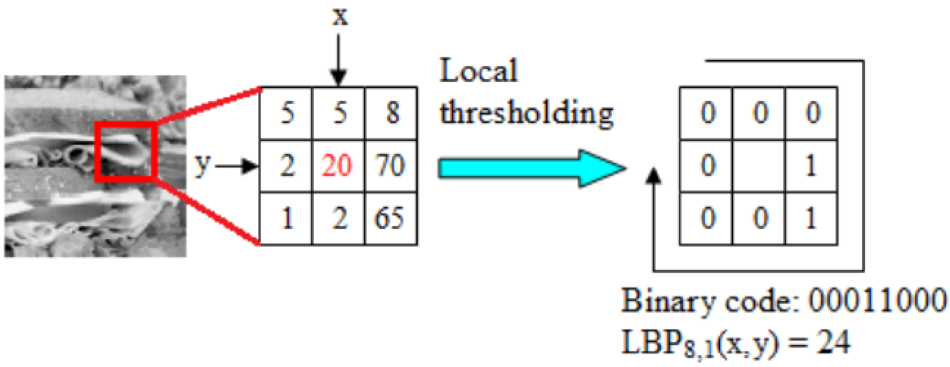

Figure 3.1: Illustration of the LBP descriptor's process

is the radius of a circle centered in c and P is the numbered of sampled points.

- g_p and g_c are the gray values (intensities) of p and c
- $s(x)$ is the function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

In Fig. 3.1, R and P are 1 and 8 respectively.

The number of histograms bins for $LBP_{P,R}$ is 2^P.

Lastly, it's important that we consider the concept of LBP uniformity. A LBP is considered to be uniform if it has at most two 0-1 or 1-0 transitions. For example, the pattern 00001000 (2 transitions) and 10000000 (1 transition) are both considered to be uniform patterns since they contain at most two 0-1 and 1-0 transitions. The pattern 01010010) on the other hand is not considered a uniform pattern since it has six 0-1 or 1-0 transitions.

The number of uniform prototypes in a Local Binary Pattern is completely dependent on the number of points p. As the value of p increases, so will the dimensionality of your resulting histogram. Please refer to the original Ojala et al. paper for the full explanation on deriving the number of patterns and uniform patterns based on this value. However, for the time being simply keep in mind that given the number of points p in the LBP there are p + 1 uniform patterns. The final dimensionality of the histogram is thus p + 2, where the added entry tabulates all patterns that are not uniform.

3.2 Color descriptor

3.2.1 Color histogram

HSV channels: Hue, Saturation and Value. It has been defined to be closer to the way human represents colours. Hue and Saturation corresponds to the chromaticity of the colour, Value to the lightness. As value is really dependant of the condition where the picture were taken, we don't use it for color histogram.

3.2.2 Color moments

3.2.3 The first two moments

For a discrete random variable X, the first two moments are defined as:

- **Expected value**:
$$\mathbb{E}[X] = \mu = \sum_{i=1}^{n} p_i x_i$$

- **Variance**:
$$\operatorname{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \sum_{i=1}^{n} p_i (x_i - \mu)^2$$

3.2.4 Hu moments

Raw moments

For a two-dimensional continuous function f(x,y) the moment (sometimes called “raw moment”) of (p + q)th order is defined as:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy$$

for p and $q \in \mathbb{N}$

Central moments

And the central moments are :

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x,y) dx dy$$

with $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$

Normalized central moments

The normalized central moments are:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\gamma}}$$

where $\gamma = 1 + \frac{I+j}{2}$ for $i + j \geq 2$

Definition of the Hu moments

On the base of those Moments, Hu in [10] introduced 7 Moments which are invariant for translation, rotation and resizing:

$$
\begin{aligned}
I_1 =& \eta_{20} + \eta_{02} \\
I_2 =& (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
I_3 =& (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
I_4 =& (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
I_5 =& (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
& + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
I_6 =& (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
I_7 =& (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
& - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}
$$

3.3 Bag-of-Words

3.3.1 Process

Bag-of-Words *BoW*, also called Bag of features, is a feature descriptor method inspired by information retrieval from textual documents.

As illustrated in Fig. 3.2, the main steps are:

- On each picture, keypoints are detected (in my case, I use a dense grid).
- For every keypoint, we describe it, SIFT (scale invariant feature transform).

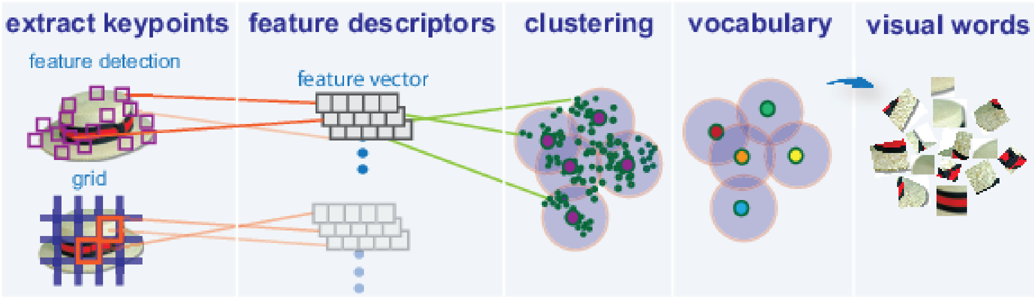

Figure 3.2: Illustration of the Bag-Of-Visual-Words model

- We generate the fix number of visual words that compose our codebook.
- We express each image as an histogram of these words' appearance.

3.3.2 SIFT and SURF

3.3.3 K-mean clustering

Given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k $(k \leq n)$ sets $S = S_1, S_2, \ldots, S_k$ so as to minimize the within-cluster sum of squares (sum of distance functions of each point in the cluster to the K center). In other words, its objective is to find:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{3.2}$$

Problem, the exact solution is a NP hard problem. That's why, we can use Lloyd's heuristic algorithm to compute an estimation.

It is an iterative method that find a local minima of the Eq. 3.2:

1. A set of k initial "means" is choosen randomly within the data domain $M = \{m_1, m_2, \ldots, m_k\}$

2. Then, k clusters are created by associating every observation with the nearest mean.

$$\forall i \in \{1,\ldots,k\},\ \ S_i^{(t)} = \big\{x_p : \big\|x_p - m_i^{(t)}\big\|^2 \le \big\|x_p - m_j^{(t)}\big\|^2\ \forall j \in \{1,\ldots,k\}\big\}$$

3. The centroid of each of the k clusters becomes the new mean.

$$\forall i \in \{1,\ldots,k\},\ \ m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

4. Repeats step 2 and 3 until M not longer changes.

The centroid results and number of iterations are highly dependant of the initial centroid.

As a result, the computation is often done several times, with different initializations of the centroids. One method to help address this issue is the k-means++ initialization scheme, which has been described in [11]. This initializes the centroids to be (generally) distant from each other, leading to provably better results than random initialization, as shown in the reference.

Chapter 4

Classifier

k-nearest neighborhood

Naive bayesian

SGD classifier + loss function + regularization term

4.1 Decision tree and random forest

Decision tree is a simple learning method that can be used for classification or regression. The implementation used of decision tree is based on the CART (Classification and Regression Tree) algorithm.

A decision tree is recursively partitioning the space in a left P_{left} and right P_{right} partitions such that the samples with the same labels are grouped together, i.e. the generated sets with the smallest impurity.

It continues to split until the impurity can't be reduced or some pre-set stopping rules are met. Alternatively, the data are split as much as possible and then the tree is later pruned.

Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as decision tree methods. The figure 4.1 illustrate a toy example of decision tree.

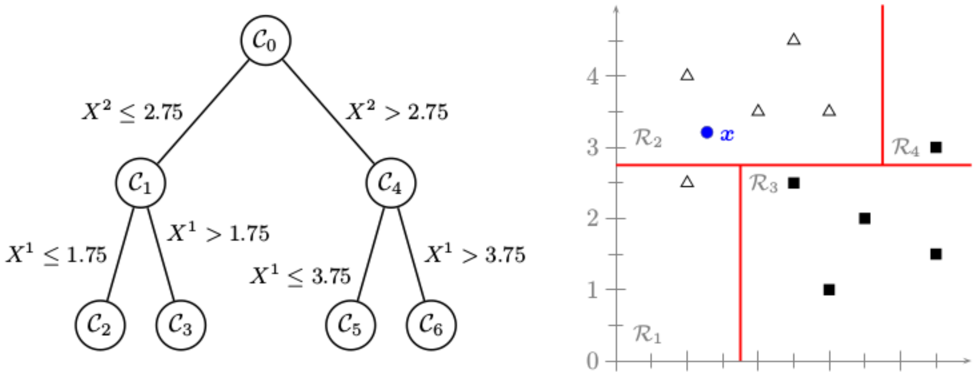

Figure 4.1: Decision tree of depth two for ten elements (X^1, X^2) belonging to the black square and white triangle classes

The most used impurity measure's functions are:

- **Gini**:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$$

- **Cross-entropy**:

$$H(X_m) = -\sum_k p_{mk} \log(p_{mk})$$

To avoid overfitting, keep the decision tree as simple as possible.

Random forest or Decision forest is build from a number of decision trees. The prediction of the ensemble is given as the averaged prediction of the individual classifiers. Each tree is trained on a random subsets of the training data.

When building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of n features. A typical value of m is $m \approx \sqrt{n}$.

4.2 Support Vector Machine

(binary case) + kernel trick + multi-class (one-versus-one or one-versus-all)

Support Vector Machine *SVM* is a method used for classification and regression.

4.2.1 Linear SVM

Hard margin

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

For a 2 classes (value represented as -1 and 1), the hyperplane must verify:

$$\vec{x}_i \cdot \vec{w} + b \geq +1 \text{ for } y_i = +1 \tag{4.1}$$

$$\vec{x}_i \cdot \vec{w} + b \leq -1 \text{ for } y_i = -1 \tag{4.2}$$

where $\vec{w}$ is the normal to the hyperplane

Combining equation 4.1 and 4.2, we obtain:

$$\forall i \in 0, \ldots, n, \;\; y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0$$

where $y_i = f(\vec{x}_i) = -1, 1$

Gemetrically, the distance between the two hyperplane from 4.1 and 4.2 is $\frac{2}{w}$ (equal width to each side).

Thus, to obtain the hyperplane with the highest margin, we want to maximize:

$$\arg\max_{\vec{w},\vec{b}} \frac{2}{\|\vec{w}\|^2}$$

which is equivalent to minimize:

$$\arg\min_{\vec{w},\vec{b}} \frac{1}{2}\|\vec{w}\|^2$$

Thus, we obtain a constrained optimization problem.

Soft Margin

For the case of non-separable training sets, we introduce a penality parameter C, $C \leq 0$ and obtain:

$$\arg\min_{\vec{w},\vec{b}\zeta} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n} \zeta_i \text{ subject to } y_i(\vec{x}_i \cdot \vec{w} + b) \geq 1 - \zeta_i, \zeta_i \geq 0, \forall i \in [1, \ldots, n]$$

The decision function for new example is:

$$f(\vec{x}) = \text{sign}(\sum_{s_i \in \text{ support vectors}} w_i \vec{s}_i \cdot \vec{x} + b)$$

where the support vectors selected sub-set of the training examples that define the boundary of the hyperplane separation and hence the classification boundary.

To generalize SVM to the case of multi-class, multiple approaches are possible:

- "one-versus-one": train a separate classifier for each different pair of labels. This leads to $\frac{N(N-1)}{2}$ classifiers
- "one-versus-all": train a single classifier per class, with the samples of that class as

positive samples and all other samples as negatives

Non-linear SVM and kernel trick

The idea of the kernel trick is to transform the initial space to a higher dimensional space where a hyperplane can separate this data. Kernel trick: use kernel function to implicitly transform datasets to a higher-dimensional using no extra memory, and with a minimal effect on computation time: realise just a dot product.

To use the linear SVM for non-linear data: project the data in a new feature H space thanks to an application and then reserch for maximum margin hyperplan in H to make sure that the new problem has a unique solution, must satisfy the Mercer's condition or simply it must be a positiv-definit matrix

- **Linear** : $k(x,y) = \langle \vec{x}, \vec{y} \rangle + C = x^T y + C$
- **Polynomial**: $k(x,y) = (\gamma \cdot \langle \vec{x}, \vec{y} \rangle + C)^d = (\gamma \times x^T y + C)^d$
- **Radial Basis Function (RBF)**: $k(x,y) = \exp\left(-\gamma \|x-y\|^2\right)$
- **Chi-Square**: $k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$

  A modified version presented in [12] of this kernel is the **Additive Chi-Square** kernel : $k(x,y) = \sum_{i=1}^{n} \frac{2(x_i - y_i)}{x_i + y_i}$

The adjustable parameters of these kernels are d, γ, C and must be choosen according to the problem.

For food classification, the chi square kernel is the most used kernel as it is often combined with histograms. !!CITE!!

4.3 CNN

inspired by the neural system composed of different layers and communication shemes recent years: use of the adjectiv "deep" to qualify NN: many layers

Different types of layers: - convolutional (give the name of the type of NN) : The Convolution layer convolves the input image with a set of learnable filters, each producing one feature map in the output image. - max pooling - normalization layer - sigmoid - ReLu

Chapter 5

Dataset

Why do we use a dataset? - learning - some research make them freely available to test

Describe how it was build ?

5.1 Choice of the datatset

Numerous datasets are already existing and have been made freely available. I could create my own dataset but it would have been very time consuming and I wouldn't be able to compare my results with previous scientific papers. To choose, a couple of criteria were defined:

- Preferably, it should be a recent dataset
- It must have a decent number of pictures (a few thousand pictures)
- It must be composed of a general kind of food such as worldwide, Western or Asian
- It must contain pictures with multi-food items

As we can see in the table 5.1, UEC FOOD 256 is the dataset that best match our expectations.

Name	Release date	Number of pictures	Type of food	Number of classes	Multiple food items
PFID [13]	2009	4545	American fast-food	101	No
UEC FOOD 100 [14]	2012	14361	Japanese	100	Yes
FIDS 30 [15]	2013	971	Fruit	30	No
ETHZ Food-101 [16]	2014	101 000	European	100	No
FooDD [17]	2015	3000	Fruit	23	Yes
UEC FOOD 256 [18]	**2015**	**31395**	**World**	**256**	**Yes**

Table 5.1: Summary of some available food datasets according to the criteria

+ ETHZ FOOD 101 + UPMC Food 101 ? NTU-FOOD ? UNICT-FD889 ? CAS dataset

5.2 UEC FOOD-100 and UEC FOOD-256

UEC FOOD-100 and **UEC FOOD-256** are datasets used for food localization and recognition.

The UEC FOOD-100 dataset can be found in [1]. It was created in 2012 and presented in [14].

It contains 100 types of food, mainly Japanese food. Each kind is represented by at least 100 samples.

As presented in figure 5.1, a photo can contain more than one food items. The dataset contains files to indicate bounding boxes marking the location of a food items.

UEC FOOD-256 can be found in [2]. It was presented in [18] in 2015. It contains the 100 types of food from UEC FOOD-100 plus 156 new ones. The newly introduced food kinds are more international dishes with food from various countries such as France, Italy,

[1]Dataset can be found at `http://foodcam.mobi/dataset100.html`
[2]Dataset can be found at `http://foodcam.mobi/dataset256.html`

the USA, China, Thailand, Vietnam, Japan and Indonesia. As for FOOD 100, every food photo has a bounding box indicating the location of the food item.

The most represented category is miso soup with 728 and rice with 620 pictures.

Figure 5.1: Pictures with multiple food items from UEC FOOD 256

Chapter 6

Implementation

6.1 Classification

6.1.1 Color histogram

For each picture:

1. extract the sub-image delimited by the bounding box
2. resize this sub-image to 224×224 pixels
3. extract the histogram of local binary pattern
4. extract the joint color histogram for the channel H and s of the HSV (hue, saturation and value) representation
5. extract the 7 hu-moment: invariant feature for translation, rotation and scale change (as stated in [10])

Normalized the data to have all features centered around zero (mean of 0) and have unit variance(variance equal to 1).

Then, apply multiple famous classifiers:

- decision tree
- random forest
- AdaBoost with decision tree
- k-nearest neighborhood
- SVM
- SGD Classifier

hyperparameter optimization: using a grid Try to optimize the accuracy for each classifier Separate the dataset in 3, 10 % for validation, 10 cross validation to select the best parameters Then 10 cross validations to train and test the classifier

Talk in result: show the best amelioration with hyperparemeter (but in general it only improve it by one or two percents)

6.1.2 Bag of words

For each picture:

1. extract the sub-image delimited by the bounding box
2. resize this sub-image to 224×224 pixels
3. detection of keypoints: use of a dense grid
4. descriptors: Root SIFT. Root SIFT is a simple variant of SIFT, presented in [19]. When the SIFT descriptors as been computed for each keypoints, we apply an element wise square root of the L1 normalized SIFT vectors

clustering: using the k-means algorithm to obtain a 2500-word codebook.

For each picture: compute the histogram of occurence counts of visual words

Kernel trick: use of a variant of the χ^2 kernel named additive χ-squared kernel presented in [12]

Then we apply the SVM classifier.

6.1.3 CNN

A pre-trained CNN used for image recognition on ImageNet Challenge 2014.

[20]

it is available [1].

The model is an improved version of the 19-layer model used by the VGG team in the ILSVRC-2014 competition.

6.2 Segmentation

A pre-trained CNN used for saliency detection.

[21]

it is available [2].

It is the same model as GoogleNet model. It is composed of 19 layers.

6.3 Code

The code is public [3].

[1] `https://gist.github.com/ksimonyan/3785162f95cd2d5fee77/`
[2] `https://gist.github.com/jimmie33/339fd0a938ed026692267a60b44c0c58`
[3] `https://github.com/bnogaret/food_log`

Using python 3.5.2 and its scientific stack (numpy, scipy, matplotlib) For the data structure: pandas [22] For the image processing: scikit-image [23] For most of the machine learning: package: sklearn [24] For the CNN framework: caffe framework [25] (using the pyhton layer) SIFT implementation: opencv 3.1 [26]

Documentation is generated from the python file using sphinx.

Chapter 7

Evaluation

7.0.1 Environment

All the code has been run on the "Astral" high performance computer of Cranfield's university. The operating system is SUSE Linux Enterprise Server 11 (64 bits architecture), with a Linux 3 kernel.

The system is separated in login nodes and compute nodes. There are two "front-end" login nodes and they contain two Intel E5-2660 (Sandy Bridge - 8 cores) CPUs giving 16 CPU cores and have a total of 192 GB of shared memory. The login nodes enable the user to connect to the system and compile one's program. There are 80 compute nodes, each node having two Intel E5-2660 (Sandy Bridge - 8 cores) CPUs. This is giving a total of 1280 available cores. Each compute node have at least accessed to 64 GB shared memory. Nodes are connected with Infiniband™ low-latency interconnect.

7.0.2 Segmentation metrics

[1]

[1] Information on the evaluation system can be found at `http://host.robots.ox.ac.uk/pascal/VOC/voc2012/devkit_doc.pdf`

To measure the precision of the localization / segmentation algorithm, we use the metrics as defined in [27].

Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the **Intersection over Union** IoU between the predicted bounding box B_p and ground truth bounding box B_{gt} must exceed 50% by the formula:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$$

To simplify the calculation, this formula can be rewritten as:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p) + area(B_{gt}) - area(B_p \cap B_{gt})}$$

Using this metric, we can compute the precision P, the recall R and the accuracy A given by:

$$P = \frac{T_p}{T_p + F_p}$$

$$R = \frac{T_p}{T_p + F_n}$$

$$A = \frac{T_p}{T_p + F_n + F_p}$$

with:

- T_p the number of true positives (the bounding boxes correctly localized)
- F_p the number of false positives (the predicted bounding boxes incorrectly localized)
- F_n the number of false negative (the ground truth bounding boxes not localized)

Note that given the convention from [27], if more than one predicted bounding box overlaps the same ground truth bounding box, only one will be considered as T_P, the rest will be F_Ps.

7.0.3 Classification metrics

cross validation accuracy confusion matrix

7.1 Results

7.1.1 Food segmentation

For the three metrics: Accuracy: 0.73 % Precision: 0.74 % Recall: 0.79 %

In [28], the authors use fine-tuned pre-trained Deep Neural Network and obtain around: Accuracy: 60 % Precision: 80 % Recall: 70 %

7.1.2 Classification

For using 10 fold cross validation without parameters optimization

using LBP (98 bins) + HS (30 * 30 bins) + mean and variance of each RGB channel + Hu-moments

- random forest: 21 % (250 trees, gini)
- decision tree: 6 % (gini)
- k-nearest neighborhood: (k=10, distance metric: minkowski, weights of each neighborhood point: uniform): 10 % and 16 % with hyperparameter optimization
- SGD classifier: 12 %

- Gaussian Naive Bayesian: 4 %
- Linear SVM: 9 % (no kernel trick)
- AdaBoost with decision tree: 4 % (SAMME.R algorithm)

using a 2500-word codebook, root-sift, k-mean, RF (500 trees): 10 %

using the CNN + Random forest (500 trees): 49 %

In [28], the authors use fine-tuned pre-trained Deep Neural Network and obtain 63 % accuracyon UEC FOOD-256.

In [29], the authors use fine-tuned pre-trained Deep Neural Network and obtain 67 % accuracyon UEC FOOD-256.

7.1.3 Segmentation followed by classification

CNN Segmenter + CNN feature descriptor + RF classifier

Result: 0.27 % (0.73 % accuracy for segmentation, 0.37 % for classifier)

Accuracy: 0.73328912 Precision: 0.74412334 Recall: 0.7963661

accuracy: 0.37 precision: 0.54 recall: 0.45 f1-score: 0.41

Top 5 : french fries 0.93006986503 beef bowl 0.951754344221 hamburger 0.954545415101 rice 0.989278742795 miso soup 0.989988865517

Least 5: meatloaf 0.0 grilled eggplant 0.0 mozuku 0.0 chicken cutlet 0.0 tanmen 0.00943396137415

134 35 clear soup || miso soup 0.830188600926 124 35 zoni || miso soup 0.745613969683 156 35 oshiruko or red bean soup || miso soup 0.71717164473 88 35 Japanese tofu and vegetable chowder || miso soup 0.591549254116 135 35 yudofu || miso soup 0.572727220661 89 35 pork miso soup || miso soup 0.568345282853 82 5 cutlet curry || beef curry 0.54411760705

23 22 beef noodle || ramen noodle 0.503703666392 238 11 kaya toast || toast 0.453488319362 153 86 Caesar salad || green salad 0.444444389575

[28] : accuracy 36.84 %, 54.44 % precision, Recall 50.86 %

Chapter 8

Conclusions / Future work / Improvement / Comment

Limitation: salient object detection

Improve object recogntion. using DCNN –> mainly a technical problem

Add the food estimation part

Appendix A

Appendix

A.1 RGB to HSV

Assuming the RGB values have been normalised to be in $[0,1]$, we have:

$$M = \max(R,G,B)$$

$$m = \min(R,G,B)$$

$$C = M - m$$

$$H = \begin{cases} 0 & \text{if } C = 0 \\ 60 \times \left[\frac{G-B}{C} \mod 6\right] & \text{if } M = R \\ 60 \times \left[\frac{B-R}{C} + 2\right] & \text{if } M = G \\ 60 \times \left[\frac{R-G}{C} + 4\right] & \text{if } M = B \end{cases}$$

$$S = \begin{cases} 0 & \text{if } M = 0 \\ \frac{C}{M} & \text{otherwise} \end{cases}$$

$$V = M$$

A.2 HSV to RGB

$$C = V \times S$$

$$X = C \times (1 - |\frac{H}{60} \mod 2 - 1|)$$

$$(R', G', B') = \begin{cases} (C, X, 0) & 0 \leq H \leq 60 \\ (X, C, 0) & 60 \leq H \leq 120 \\ (0, C, X) & 120 \leq H \leq 180 \\ (0, X, C) & 180 \leq H \leq 240 \\ (X, 0, C) & 240 \leq H \leq 300 \\ (C, 0, X) & 300 \leq H \leq 360 \end{cases}$$

$$m = V - C$$

$$(R, G, B) = (R' + m, G' + m, B' + m)$$

Bibliography

[1] Ali H Mokdad et al. "Prevalence of obesity, diabetes, and obesity-related health risk factors." In: *JAMA : the journal of the American Medical Association* 289.1 (2003), pp. 76–9. ISSN: 0098-7484. DOI: 10.1001/jama.289.1.76..

[2] Ping Zhang et al. "Global healthcare expenditure on diabetes for 2010 and 2030". In: *Diabetes Research and Clinical Practice* 87.3 (2010), pp. 293–301. ISSN: 01688227. DOI: 10.1016/j.diabres.2010.01.026. URL: http://dx.doi.org/10.1016/j.diabres.2010.01.026.

[3] Lora E. Burke, Jing Wang, and Mary Ann Sevick. "Self-Monitoring in Weight Loss: A Systematic Review of the Literature". In: *Journal of the American Dietetic Association* 111.1 (2011), pp. 92–102. ISSN: 00028223. DOI: 10.1016/j.jada.2010.10.008. URL: http://dx.doi.org/10.1016/j.jada.2010.10.008.

[4] S W Lichtman et al. "Discrepancy between self-reported and actual caloric intake and exercise in obese subjects." In: *The New England Journal of Medicine* 327.27 (1992), pp. 1893–1898. ISSN: 0028-4793. DOI: 10.1056/NEJM199212313272701. arXiv: arXiv:1011.1669v3.

[5] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 15731405. DOI: 10.1007/s11263-015-0816-y. arXiv: 1409.0575.

[6] Richard Hillestad et al. “Can electronic medical record systems transform health care? Potential health benefits, savings, and costs.” In: *Health affairs (Project Hope)* 24.5 (2005), pp. 1103–17. ISSN: 0278-2715. DOI: 10.1377/hlthaff.24.5.1103. URL: http://www.ncbi.nlm.nih.gov/pubmed/16162551.

[7] Nir Menachemi and Taleah H. Collum. “Benefits and drawbacks of electronic health record systems”. In: *Risk Management and Healthcare Policy* 4 (2011), pp. 47–55. ISSN: 11791594. DOI: 10.2147/RMHP.S12985. arXiv: 0710.4428v1.

[8] Joachim Dehais, Marios Anthimopoulos, and Stavroula Mougiakakou. “Dish Detection and Segmentation for Dietary Assessment on Smartphones”. In: *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops* 9281 (2015), pp. 433–440. ISSN: 16113349. DOI: 10.1007/978-3-319-23222-5. URL: http://link.springer.com/chapter/10.1007/978-3-319-23222-5%7B%5C_%7D53.

[9] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), pp. 971–987. ISSN: 01628828. DOI: 10.1109/TPAMI.2002.1017623.

[10] Ming-Kuei Hu. “Visual pattern recognition by moment invariants”. In: *IRE Transactions on Information Theory* 8 (1962), pp. 179–187. ISSN: 0096-1000. DOI: 10.1109/TIT.1962.1057692.

[11] David Arthur and Sergei Vassilvitskii. “k-means++: The Advantages of Careful Seeding”. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* 8 (2007), pp. 1027–1035. URL: http://portal.acm.org/citation.cfm?id=1283494.

[12] A Vedaldi and A Zisserman. “Efficient Additive Kernels via Explicit Feature Maps”. In: *{IEEE} Int. Conf. on Computer Vision and Pattern Recognition* XX.Xx (2010), pp. 3539–3546.

[13] Mei Chen et al. “PFID: Pittsburgh Fast-food Image Dataset”. In: *Proceedings - International Conference on Image Processing, ICIP* (2009), pp. 289–292. ISSN: 15224880. DOI: 10.1109/ICIP.2009.5413511.

[14] Yuji Matsuda, Hajime Hoashi, and Keiji Yanai. “Recognition of multiple-food images by detecting candidate regions”. In: *Proceedings - IEEE International Conference on Multimedia and Expo*. IEEE, July 2012, pp. 25–30. ISBN: 978-1-4673-1659-0. DOI: 10.1109/ICME.2012.157. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6298369.

[15] Škrjanec Marko. “Automatic fruit recognition using computer vision”. Mentor: Matej Kristan. Bsc thesis. Faculty of Computer and Information Science, University of Ljubljana, 2013.

[16] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. “Food-101 - Mining discriminative components with random forests”. In: *Lecture Notes in Computer Science*. Vol. 8694 LNCS. PART 6. 2014, pp. 446–461. ISBN: 9783319105987. DOI: 10.1007/978-3-319-10599-4_29. arXiv: 978-3-319-10599-4{_}29 [10.1007]. URL: http://link.springer.com/chapter/10.1007/978-3-319-10599-4%7B%5C_%7D29.

[17] Parisa Pouladzadeh Abdulsalam Yassine and Shervin Shirmohammadi. “FooDD: Food Detection Dataset for Calorie Measurement Using Food Images”. In: *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops* 9281 (2015), pp. 441–448. ISSN: 16113349. DOI: 10.1007/978-3-319-23222-5. URL:

http://link.springer.com/chapter/10.1007/978-3-319-23222-5%7B%5C_%7D54.

[18] Yoshiyuki Kawano and Keiji Yanai. "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation". In: *Lecture Notes in Computer Science* 8927 (2015), pp. 3–17. ISSN: 16113349. DOI: 10.1007/978-3-319-16199-0_1.

[19] Relja Arandjelovic and Andrew Zisserman. "Three things everyone should know to improve object retrieval c". In: *IEEE Conference on computer vision and Pattern Recognition* April (2012), pp. 2911–2918. ISSN: 9781467312288. DOI: 10.1109/CVPR.2012.6248018.

[20] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *ImageNet Challenge* (2014), pp. 1–10. ISSN: 09505849. DOI: 10.1016/j.infsof.2008.09.005. arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556.

[21] Jianming Zhang et al. "Unconstrained Salient Object Detection via Proposal Subset Optimization". In: *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)* (2016). URL: http://cs-people.bu.edu/jmzhang/sod.html.

[22] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference* (2010), pp. 51–56. URL: http://conference.scipy.org/proceedings/scipy2010/mckinney.html.

[23] Stéfan van der Walt et al. "Scikit-image: image processing in Python". In: *PeerJ* 2 (2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. arXiv: 1407.6245. URL: https://peerj.com/articles/453.

[24] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: … *of Machine Learning …* 12 (2012), pp. 2825–2830. ISSN: 15324435. DOI: 10.1007/s13398-014-0173-7.2. arXiv: 1201.0490. URL: http://scikit-learn.org/stable/.

[25] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *Proceedings of the ACM International Conference on Multimedia* (2014), pp. 675–678. ISSN: 10636919. DOI: 10.1145/2647868.2654889. arXiv: 1408.5093. URL: http://arxiv.org/abs/1408.5093.

[26] G Bradski. "The OpenCV Library". In: *Dr Dobbs Journal of Software Tools* 25 (2000), pp. 120–125. ISSN: 1044-789X. DOI: 10.1111/0023-8333.50.s1.10. URL: http://opencv.org/.

[27] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. URL: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[28] Marc Bolaños and Petia Radeva. "Simultaneous Food Localization and Recognition". In: (2016), pp. 2–7. arXiv: 1604.07953. URL: http://arxiv.org/abs/1604.07953.

[29] Keiji Yanai and Yoshiyuki Kawano. "Food image recognition using deep convolutional network with pre-training and fine-tuning". In: *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, June 2015, pp. 1–6. ISBN: 978-1-4799-7079-7. DOI: 10.1109/ICMEW.2015.7169816. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7169816.