UNIVERSITÀ
degli STUDI
di CATANIA

Dipartimento di Matematica e Informatica

Dottorato di ricerca in Informatica

IMAGE REPRESENTATION USING CONSENSUS VOCABULARY AND FOOD IMAGES CLASSIFICATION

MARCO MOLTISANTI

A dissertation submitted to the Department of Mathematics and Computer Science and the committee on graduate studies of University of Catania, in fulfillment of the requirements for the degree of doctorate in Computer Science.

ADVISOR
Prof. Sebastiano Battiato
CO-ADVISORS
Prof. Giovanni Maria Farinella
Dr. Arcangelo Ranieri Bruna

XXVIII Ciclo

Contents

List of publications

- **Proceedings** S. Battiato, M. Moltisanti, F. Ravì, A. R. Bruna, and F. Naccari, *Aestethic scoring of digital portraits for consumer applications* in Proceedings of SPIE 8660 - Digital Photography IX 866008 (February 4, 2013);

- **Proceedings** G.M. Farinella, M. Moltisanti, S. Battiato, *Classifying food images represented as Bag-of-Textons* in Proceedings of IEEE International Conference on Image Processing, pp.5212-5216, Paris, 2014;

- **Proceedings** S. Battiato, M. Moltisanti – *The future of consumer cameras.* In Proceedings of SPIE 9399, Image Processing: Algorithms and Systems XIII, 93990C (March 16, 2015).

- **Proceedings** M. Moltisanti, A. Paratore, S. Battiato, L. Saravo – *Image manipulation on Facebook for Forensics Evidence.* In Image Analysis and Processing — ICIAP 2015, V. Murino and E. Puppo, Eds., vol. 9280 of Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 506 – 517.

- **Proceedings** G. M. Farinella, M. Moltisanti, S. Battiato – *Food Recognition Using Consensus Vocabularies.* In New Trends in Image Analysis and Processing – ICIAP 2015 Workshops, V. Murino, E. Puppo, D.

Sona, M. Cristani, and C. Sansone, Eds., vol. 9281 of Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 384 –392.

- **Proceedings** M. Moltisanti, G. M. Farinella, S. Battiato *Semi-Naive Mixture Model for Consensus Clustering.* To appear in International Workshop on Machine learning, Optimization and big Data, Lecture Notes in Computer Science. Springer International Publishing, 2015.

- **Book Chapter** S. Battiato and M. Moltisanti, *Tecniche di steganografia su immagini digitali*, in IISFA Memberbook 2012 DIGITAL FORENSICS, G. Costabile and A. Attanasio, Eds. Experta, Italy, 2012.

- **Technical Report** A. Furnari, V. Giuffrida, D. Moltisanti and M. Moltisanti - *Reading Group Report* (winner of the Reading Group Competition Prize) - ICVSS 2013

- **Technical Report** M. Buffa, A. Furnari, O. Giudice, V. Giuffrida, M. Moltisanti, A. Ortis, A. Torrisi, *Reading Group Report - ICVSS 2014*

- **Accepted** M. Moltisanti, G. M. Farinella, S. Battiato, A. R. Bruna – *Exploiting Visual Saliency for Car Detection and Tracking.* Submitted to IS&T Electronic Imaging – Image Processing: Machine Vision Applications, San Francisco, 2016

- **Submitted** G. M. Farinella, D. Allegra, M. Moltisanti, F. Stanco and S. Battiato *Retrieval and Classification of Food Images.* Submitted to Computers in Biology and Medicine.

- **Other** G. M. Farinella, D. Allegra, M. Moltisanti, F. Stanco, S. Battiato – *Food understanding from digital images*, La Simulazione nel settore Food & Beverage, 2015.

List of Figures

List of Tables

Chapter 1

Introduction

Digital images are the result of many physical factors, such as illumination, point of view an thermal noise of the sensor. These elements may be irrelevant for a specific Computer Vision task; for instance, in the object detection task, the viewpoint and the color of the object should not be relevant in order to answer the question “Is the object present in the image?”. Nevertheless, an image depends crucially on all such parameters and it is simply not possible to ignore them in analysis [3]. Hence, finding a **representation** that, given a specific task, is able to keep the significant features of the image and discard the less useful ones is the first step to build a robust system in Computer Vision.

One of the most popular model to represent images is the **Bag-of-Visual-Words (BoW)** model. Derived from text analysis, this model is based on the generation of a codebook (also called vocabulary) which is subsequently used to provide the actual image representation. Considering a set of images, the typical pipeline, depicted in Fig. 1.1, consists in:

1. Select a subset of images to be the *training set* for the model;

2. Extract the desired features from the all the images;

3. Run a clustering algorithm on the features extracted from the training set: each cluster is a **codeword**, the set containing all the clusters is the **codebook**;

4. For each feature point, find the closest codeword according to a distance function or metric;

5. Build a normalized histogram of the occurrences of each word.

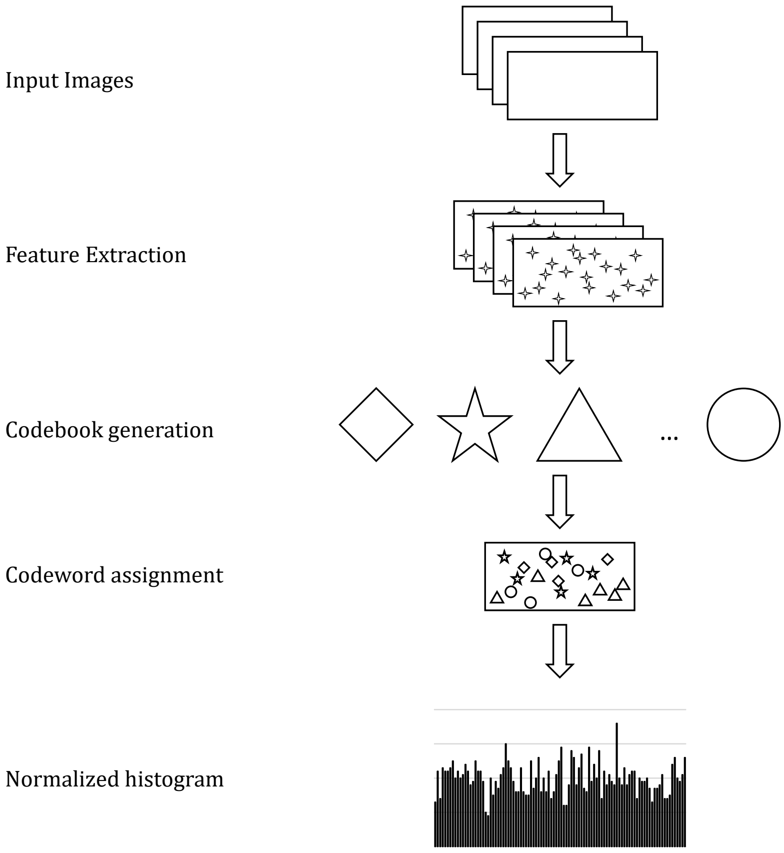


Figure 1.1: Typical Bag-of-Visual-Words pipeline.

The choices made in the design phase influence strongly the final outcome of the representation. For example, choosing a feature which is able to capture changes in the local gradient (e.g. SIFT [4]) will give different results than choosing a feature which is able to describe textures (e.g. Textons [5, 6]). In this work we will discuss how to aggregate different kind of features to obtain more powerful representations, presenting some state-of-the-art (Chapter 2) methods in Computer Vision community. We will focus on Clustering Ensemble techniques (Section 2.3), presenting the theoretical framework (Section 2.4) and a new approach (Section 2.5).

In the second part of this work, we discuss about food image analysis. Understanding food in everyday life (e.g., the recognition of dishes and the related ingredients, the estimation of quantity, etc.) is a problem which has been considered in different research areas due its important impact under the medical, social and anthropological aspects. For instance, an insane diet can cause problems in the general health of the people. Since health is strictly linked to the diet, advanced Computer Vision tools to recognize food images (e.g., acquired with mobile/wearable cameras), as well as their properties (e.g., calories, volume), can help the diet monitoring by providing useful information to the experts (e.g., nutritionists) to assess the food intake of patients (e.g., to combat obesity). On the other hand, the great diffusion of low cost image acquisition devices embedded in smartphones allows people to take pictures of food and share them on Internet (e.g., on social media); the automatic analysis of the posted images could provide information on the relationship between people and their meals and can be exploited by food retailer to better understand the preferences of a person for further recommendations of food and related products. Image representation plays a key role while trying to infer information about food items depicted in the image. We propose a deep review of the state-of-the-art Section 3.2 and two different novel representation techniques (Section 3.3, Section 3.4).

Chapter 2

Feature Aggregation

2.1 Visual Phrases

The Bag-of-Visual-Word (BoW) model relies on the generation of a vocabulary using a clustering method. This approach, first proposed in [7], is widely employed for its simplicity and flexibility, but, in its original formulation, it suffers from the limitation of being able to use only one kind of feature at a time. To overcome this restriction, several solutions have been proposed.

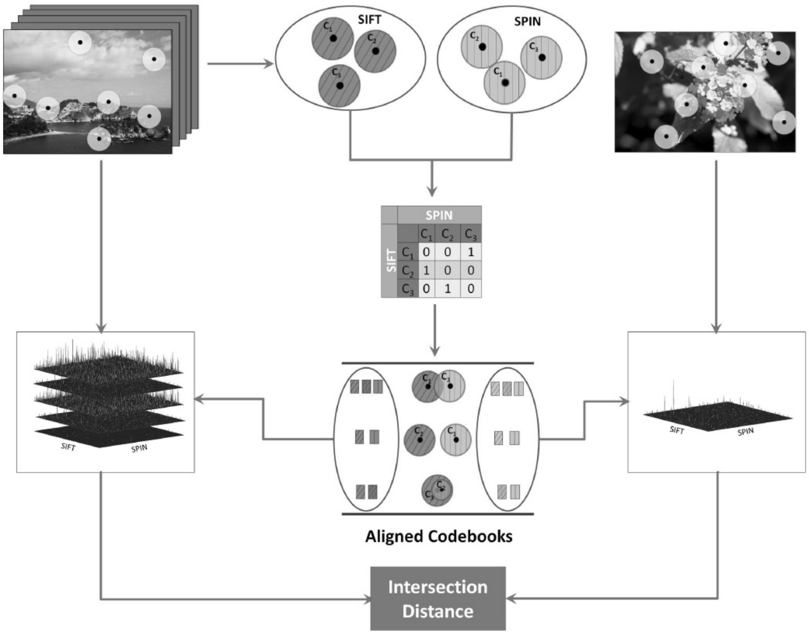
Hu *et al.* in [8] develop two methods based on the extraction of multiple features, in particular the Feature Coherent Phrase (FCP) model and the Spatial Coherent Phrase model. First, given an image,local regions are located by using some detector [9] or using dense sampling [10]. For each extracted region, K different descriptors ϕ_{ik} are computed, generating a descriptor $\phi_i = \{\phi_{i1}, \phi_{i2}, \ldots, \phi_{iK}\}$. The final descriptor is obtained mapping every ϕ_i to a $K-$tuple of visual words. This tuple is called a **visual phrase**. Hence, mapping all the ϕ_i extracted from an image to the corresponding visual phrases and computing the frequency histogram of the co-occurrence gives the final representation of the image. The authors of [8] propose two different coherent models:

a) The Feature Coherent Phrase (FCP) model;

b) The Spatial Coherent Phrase (SCP) model.

In the first case, the visual phrase is built out of different kind of features extracted on the same image region; specifically, they use SIFT [4] and SPIN [11] features. In the second case, the same kind of feature is extracted from the same region varying the scale at which the descriptor is computed.

An improvement to the Bags of Visual Phrases model is proposed by Battiato *et al.* [1]. In their work, the authors propose to exploit the coherence between feature spaces not only in the image representation, but also during the generation of codebooks. This is obtained by aligning the codebooks of different descriptors to produce a more significant quantization of the involved spaces of descriptors. The algorithm, depicted in Fig. 2.1, starts detecting a set of keypoints from a training dataset of images. Then, SIFT [4] and SPIN [11] are computed and vocabularies are built separately in these two feature spaces. The use of two different vocabularies poses a cluster correspondence problem. To deal with it, a similarity matrix is obtained counting the number of elements (i.e. local image regions) they share, and the Hungarian algorithm is used to find the pair the corresponding clusters. Using this correspondence information, a new vocabulary is created, taking into account both the common and the uncommon elements between aligned clusters. The representation of training images consists in a two-dimensional histogram of co-occurrence of visual words related to the generated codebook.



SIFT \ SPIN	C1	C2	C3
C1	0	0	1
C2	1	0	0
C3	0	1	0

Figure 2.1: Schema of the algorithm proposed in [1]. Image courtesy of the authors.

2.2 Capturing the shape and its layout: Pyramid of Histograms of Orientation Gradients (PHOG)

Bosch *et al.* [12] set as their goal to represent an image by its shape local shape and the spatial layout of the shape. The idea is illustrated in Fig. 2.2.

The descriptor consists of a histogram of orientation gradients over each image subregion at each resolution level – a Pyramid of Histograms of Orientation Gradients (PHOG). The distance between two PHOG image descriptors then reflects the extent to which the images contain similar shapes and correspond in their spatial layout. To encode the local shape, a classic

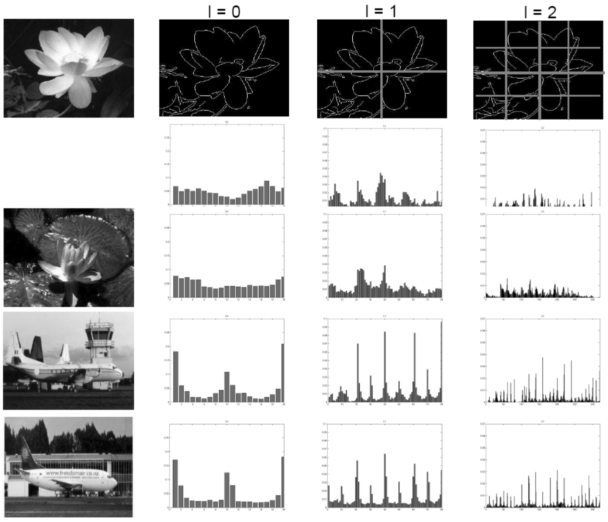


Figure 2.2: Shape spatial pyramid representation. Top row: an image and grids for levels l = 0 to l = 2; Below: histogram representations corresponding to each level. The final PHOG vector is a weighted concatenation of vectors (histograms) for all levels. Remaining rows: images from the same and from different categories, together with their histogram representations. Image courtesy of the authors.

BoW approach is employed, using SIFT features, while the spatial layout is captured following the well-known scheme of spatial pyramid matching proposed by Lazebnik *et al.* [10]. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction (like a quadtree). The number of points in each grid cell is then recorded. This is a pyramid representation because the number of points in a cell at one level is simply the sum over those contained in the four cells it is divided into at the next level. The Histograms of Oriented Gradients (HOG) [13] vectors are computed for each grid cell at every pyramid resolution level, and subsequently concatenated to form the PHOG descriptor. However, shapes and their layouts could not be sufficient to discriminate among different kind of images; then, the authors propose to consider also the appearance as a feature, in order to perform the image classification with a standard SVM. To deal with the combination of such aspects, they introduce two kernels, suitable for combining or selecting between shape representation and appearance, alongside with a kernel which encodes the similarity between PHOG descriptors. While the last kernel is a simple χ^2 distance matrix (see Eq. 2.1), the first and the second combine in different ways the shape representation and the appearance, as shown in Eq. 2.2 and Eq. 2.3.

$$K\left(S_I, S_J\right) = \sum_{l \in L} \alpha_l d_l \left(S_I, S_J\right) \tag{2.1}$$

In Eq. 2.1, the distance between the representation of two images $S_I, S_J,\ I \neq J$ is computed at each of the L levels of the spatial pyramid, and the value for each cell of the kernel is obtained summing over all the $l \in L$. The kernels in Eq. 2.2 and Eq. 2.3 show the combination of two kernels, one for appearance (K_A) and one for shapes (K_S), both derived from the χ^2 kernel in Eq. 2.1. Please note the change of notation: in facts, while in Eq. 2.1 the inputs are generic image representations of images S_I, S_J, the kernels proposed in Eq. 2.2 and Eq. 2.3 take as input the image indexes x, y, and two different representation are considered as input to K_A and K_S. The parameters α and β, in both Equations 2.1 and 2.2 are learned from data, and two different strategies have been tested: a global approach, where the weights are optimized over all the classes together, and a class-specific approach, where the weights are optimized for each class separately.

$$K\left(x, y\right) = \alpha K_A \left(x_{app}, y_{app}\right) + \beta K_S \left(x_{shape}, y_{shape}\right) \tag{2.2}$$

$$K\left(x,y\right)=\max\left[K_A\left(x_{app},y_{app}\right),K_S\left(x_{shape},y_{shape}\right)\right] \tag{2.3}$$

2.3 Clustering Analysis and Consensus Ensemble

The definition of clustering encloses a wide range of different techniques, all of them aiming to group similar objects according to a similarity or distance function. The factors in this definition, together with the choice of the cluster model (e.g. connectivity model, centroid model, distribution model, etc.) lead to the high variability in the clustering algorithms family [14]. Among the different employments of clustering algorithms, the Bag-of-Words model is one of the most popular, especially in Computer Vision community.

In [15], Kleinberg defines some desirable properties, proving that there is no clustering function able to satisfy them all together:

Scale-Invariance: insensitivity to changes in the units of distance measurements;

Richness: every partition of the data space S should be a possible output of the algorithm;

Consistency: changing the distance function to reduce intra-cluster distances and augment inter-cluster distances, the output partition should be the same.

Combining more partitions of the same space can be interpreted as a partitioning task itself. Typically, each partition in the combination is represented as a set of labels assigned by a clustering algorithm. The output partition (i.e. the combined one) is generated taking as inputs the labels of the contributing clustering algorithms, which are processed by another clustering algorithm. In general, the clustering ensemble framework can be pictured as shown in Fig. 2.3. *Classifying* and *clustering* ensembles, although similar, are still different problems. Designing a cluster ensemble framework is more difficult than designing a classifier since a correspondence problem arises, for the cluster labels are merely symbolic. Moreover, it is not possible to know in advance the number of clusters, and the high variability of the number and shapes of input clusterings must be taken into account. Ghaemy *et al.* [16] define the problem of clustering ensembles (or clustering combination) as follows:

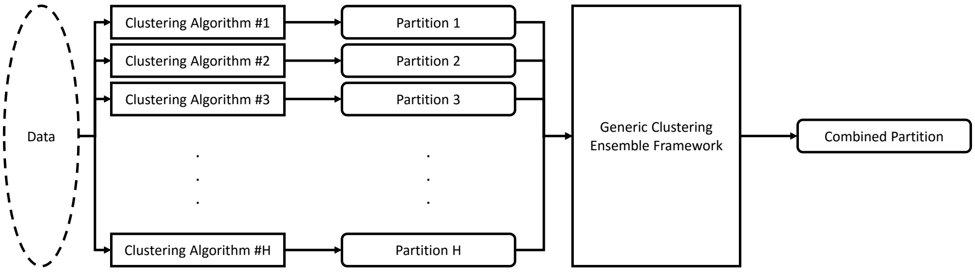


Figure 2.3: Generic representation of a clustering ensemble framework.

> Given multiple clusterings of the dataset, find a combined clustering with better quality.

The combination problem poses three specific problems [17, 18]:

1. The choice of the consensus function : combination of different clusterings outcomes, label correspondence problem, symmetry and equity with respect to all the input partitions;

2. The diversity of clustering: the generation of different partitions can be achieved applying various clustering algorithms [19], varying the initialization or the parameters on different runs of the same algorithm [20, 21, 22], projecting the data onto different subspaces [19, 23], choosing different subsets of features [19] or selecting different subset of data points [24, 25, 26];

3. The robustness of the components to be considered: use of weak partitions and minimal complexity [2].

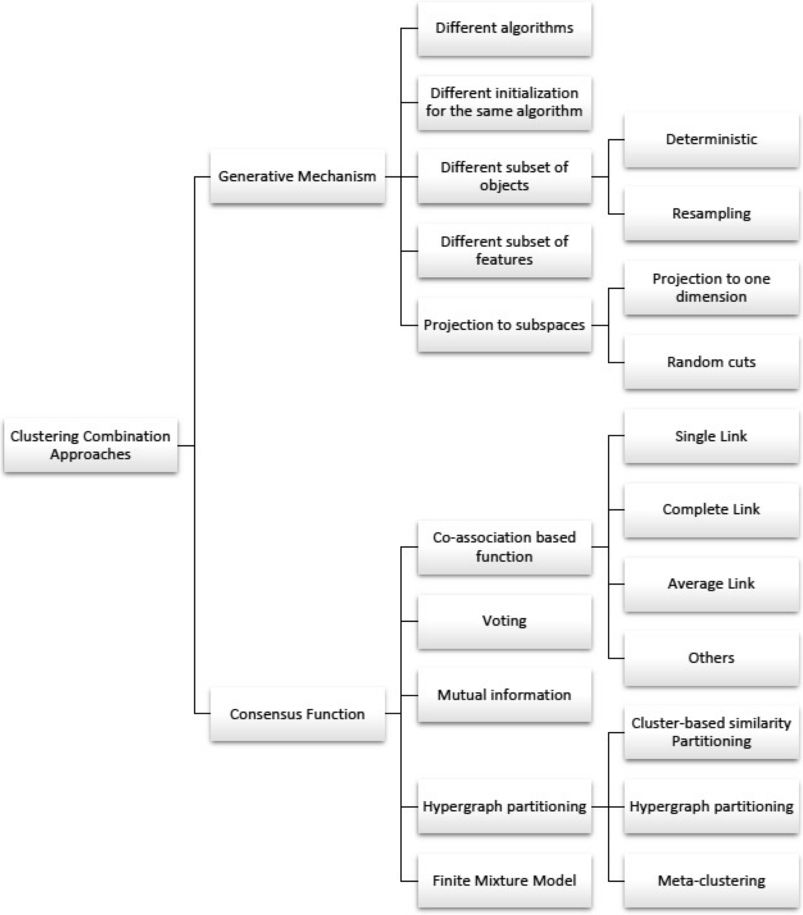


Figure 2.4: Visual taxonomy of different clustering combination techniques.

A visual taxonomy of different clustering ensembles techniques is reported in Fig. 2.4. This taxonomy is proposed in [16], together with a deep review of state-of-the-art methods.

2.4 Consensus Clustering via Expectation – Maximization

Topchy *et al.* [2] modeled the problem using a Gaussian Mixture Model (GMM) in order to find the consensus partition by solving a Maximum Likelihood optimization. Given N data points, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, they consider the outcomes of H different clustering algorithms, each of which establish a partition in the feature space. They refer to the partitions as $\mathbf{H} = \{\pi_1, \pi_2, \ldots, \pi_H\}$. It is straightforward that every clustering algorithm assigns each data point $\mathbf{x}_i$ to a partition:

$$\mathbf{x}_i \rightarrow \{\pi_1(\mathbf{x}_i), \pi_2(\mathbf{x}_i), \ldots, \pi_H(\mathbf{x}_i)\}, \qquad i = 1, \ldots, N$$

Therefore, each data point $\mathbf{x}_i$ has two representation: the first is a $d-$dimensional vector that lies in original the feature space, while the second is a vector with H elements that belongs to the labels space (Tab. 2.1). The vector composed by the labels for the $i-$th data point will be named $\mathbf{y}_i$. The whole labels set will be denoted as $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$. The rationale behind this approach is that the labels can be modeled as random variables, drawn from a GMM. Hence, the probability for each label $\mathbf{y}_i$ can be expressed as in Eq. 2.4, where α_m, , with $m = 1, \ldots, M$, are the mixture coefficients and θ_m are the parameters of each component of the mixture.

$$P(\mathbf{y}_i|\Theta) = \sum_{m=1}^{M} \alpha_m P_m(\mathbf{y}_i|\theta_m) \tag{2.4}$$

				π_1	$\cdots$	π_H
$\mathbf{x_1}$	x_{11}	$\cdots$	x_{1d}	$\pi_1(\mathbf{x}_1)$	$\cdots$	$\pi_H(x_1)$
$\mathbf{x_2}$	x_{21}	$\cdots$	x_{2d}	$\pi_1(\mathbf{x}_2)$	$\cdots$	$\pi_H(\mathbf{x}_2)$
$\vdots$						
$\mathbf{x_N}$	x_{N1}	$\cdots$	x_{Nd}	$\pi_1(\mathbf{x}_N)$	$\cdots$	$\pi_H(\mathbf{x}_N)$
	Original Features			Labels		

Table 2.1: Transformation of data representation, from feature space (left) to label space (right).

Using this model under the assumption that the data points are independent and identically distributed, the consensus partition can be found optimizing as the partition which maximize the probability, for each $\mathbf{y}_i$, of having been drawn from the $m-$th mixture. Hence, the problem can be formulated as finding the GMM's parameters that maximize the label-to-mixture assignment probability.

$$\Theta^* = \arg\max_{\Theta} \log L(\Theta|\mathbf{Y}_i). \tag{2.5}$$

where L is a likelihood function, as defined in Eq. 2.6

$$\log L(\Theta|\mathbf{Y}) = \log \prod_{m=1}^{M} P(\mathbf{y}_i|\theta_m) = \sum_{i=1}^{N} \log \sum_{m=1}^{M} \alpha_m P_m(\mathbf{y}_i|\theta_m) \tag{2.6}$$

To complete the definition of the model, it is needed to specify the conditional probabilities for the labels vector $\mathbf{y}_i$ (see Eq. 2.7) and the probability density for each component (see Eq. 2.8). In [2], the authors assume that the components of $\mathbf{y}_i$ are conditionally independent.

$$P_m(\mathbf{y}_i|\theta_m) = \prod_{j=1}^{H} P_m^{(j)}\left(y_{ij}|\theta_m^{(j)}\right) \tag{2.7}$$

$$P_m^{(j)}\left(y_{ij}|\theta_m^{(j)}\right) = \prod_{k=1}^{K(j)} \vartheta_{jm}(k)^{\delta(y_{ij},k)} \tag{2.8}$$

Note that the probabilities $\vartheta_{jm}(k)$ sum up to 1. In Eq. 2.8, the function δ is a classic Kronecker delta function and the index $k = 1, \ldots, K(j)$ is used to enumerate the labels in the $j-$th input mixture.

The solution to the consensus partition problem can be found optimizing Eq. 2.4, hypothesizing the existence of a set of hidden variables $\mathbf{Z}$ and estimating the values of each $\mathbf{z}_i$ using the Expectation-Maximization algorithm. For completeness sake, in Equations 2.9, 2.10, 2.11 we report the formulas to compute the parameters of the mixture with the EM algorithm.

$$E\left[z_{im}\right] = \frac{\alpha'_m \prod_{j=1}^{H} \prod_{k=1}^{K(j)} \left(\vartheta'_{jm}(k)\right)^{\delta(y_{ij},k)}}{\sum_{n=1}^{M} \alpha'_n \prod_{j=1}^{H} \prod_{k=1}^{K(j)} \left(\vartheta'_{jn}(k)\right)^{\delta(y_{ij},k)}} \tag{2.9}$$

$$\alpha_m = \frac{\sum_{i=1}^{N} E\left[z_{im}\right]}{\sum_{i=1}^{N} \sum_{m=1}^{M} E\left[z_{im}\right]} \tag{2.10}$$

$$\vartheta_{jm}(k) = \frac{\sum_{i=1}^{N} \delta\left(y_{ij}, k\right) E\left[z_{im}\right]}{\sum_{i=1}^{N} \sum_{k=1}^{K(j)} \delta\left(y_{ij}, k\right) E\left[z_{im}\right]} \tag{2.11}$$

2.4.1 Experimental Results

We applied this approach to the Near Duplicate Image Retrieval problem. We used the UKBench [27] dataset, which contains a total of 10200 images of 2550 different objects with four near duplicate images (photometric and/or geometric variations) for each object [27]. We proceeded extracting SIFT [4] and SPIN [11] features using a dense sampling approach. For comparison purposes, we computed the bi-dimensional histogram of co-occurrences as in [8]. Initially, we used a subset of the dataset restricted to the first 1000 images, in order to perform a parameter tuning phase and looking forward to extend the approach to the whole dataset. The parameters involved in the tuning step are:

- The number of clusters in the feature spaces: K;
- The number of output clusters after the consensus aggregation: M;
- The number of different partitions used as input to the consensus procedure: H.

The pipeline is depicted in Fig. 2.5. After the dense sampling and the feature description, H vocabularies are built using the BoW approach ($\frac{H}{2}$ using the SIFT descriptors and $\frac{H}{2}$ using the SPIN descriptors). Then, each point is labeled using the cluster index to which it has been assigned. In other words, we are moving from the feature spaces to the label spaces. The E-M Consensus Clustering algorithm takes as input these labels and produces the final consensus partition, selecting the most likely among the M clusters for the given data point. The Consensus Vocabulary is used to build a representation in the BoW fashion, and then using a standard SVM with a χ^2 kernel to perform classification. The accuracy is referred to this task, while the retrieval performances are estimated using the mean Average Precision (mAP) as shown in Eq. 2.12

$$mAP = \frac{\sum\limits_{q=1}^{Q} AveP\left(q\right)}{Q}, \quad \text{where } Q = \text{number of queries} \tag{2.12}$$

We performed a first bank of tests varying the dimension of input vocabular-

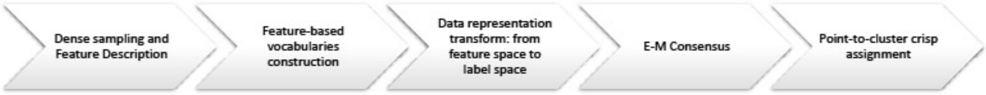
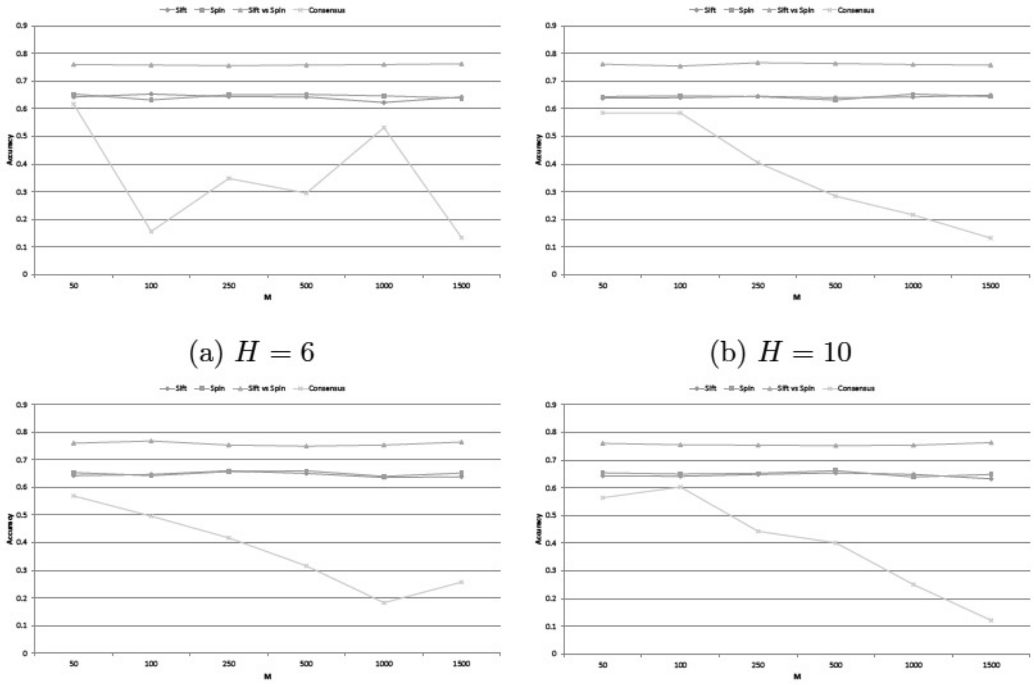

Dense sampling and Feature Description
Feature-based vocabularies construction
Data representation transform: from feature space to label space
E-M Consensus
Point-to-cluster crisp assignment

Figure 2.5: The pipeline implemented to represent images using Consensus Vocabularies and crisp point-to-cluster assignment.

ies K, the number of different partitions H and the number of final clusters M. To whom it concerns K, we used a value of $K = 50$ after a tuning parameter stage. Fig. 2.6



(a) $H = 6$

(b) $H = 10$

(c) $H = 16$

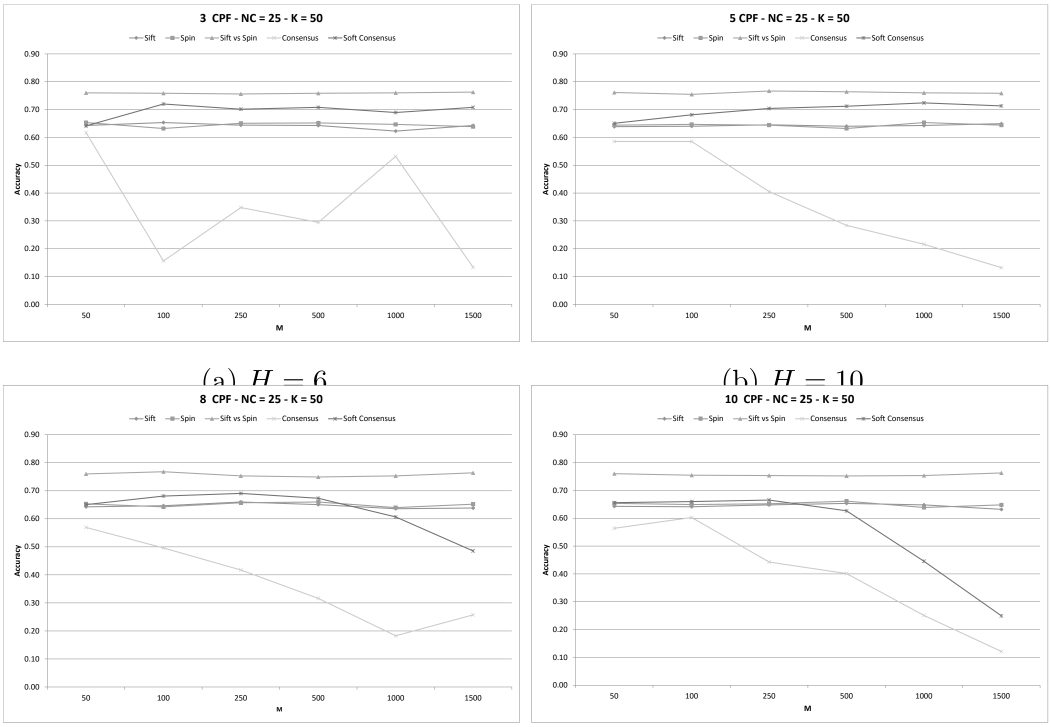
(d) $H = 20$

Figure 2.6: Comparison of the classification accuracies.

It is evident how the performances were not satisfying, considering that the bi-dimensional co-occurrences histogram (SIFT vs SPIN in the charts) considerably outperformed all the other representations. Hence, we modified the framework introducing a soft assignment instead of the crisp one. The image representation is then a normalized histogram (one bin per cluster) of the probabilities that the represented image belongs to the considered bin (see Eq. 2.13).

$$h_{SC}(I) = (p_1(I), p_2(I), \ldots, p_M(I)) \tag{2.13}$$

Fig. 2.7 and Fig. 2.8 shows the accuracy of the method adding the soft assignment.



(a) $H = 6$

(b) $H = 10$

(c) $H = 16$

(d) $H = 20$

Figure 2.7: Comparison of the classification accuracies.

Despite the improvement is considerable, the method proposed in [8] was still better. Looking at the histograms representing the images, we noticed that many of the bins after the normalization had very very low values, but

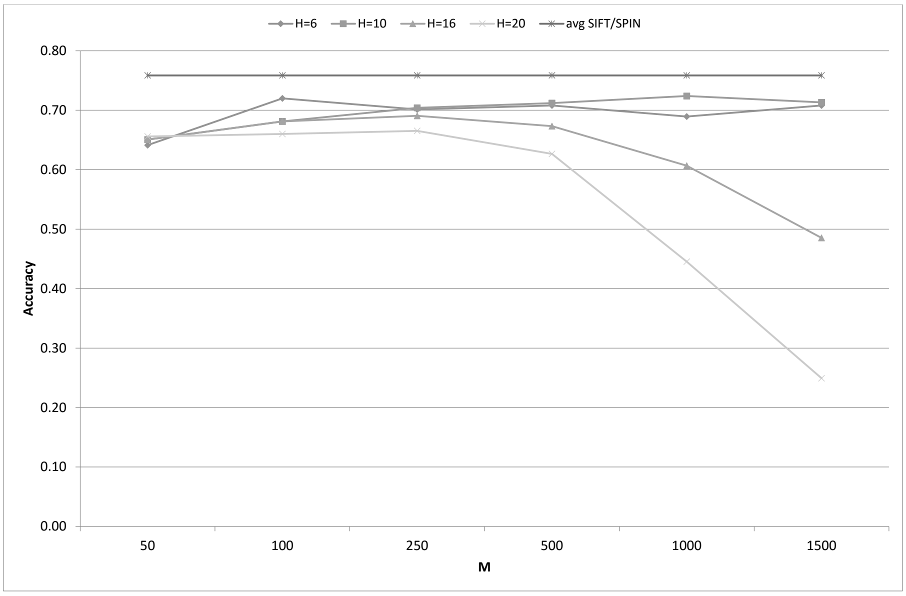


Figure 2.8: Classification accuracy of the E-M based Consensus Vocabulary approach with soft assignment.

different than zero. We considered this as a sort of noise in the representation of the image, so we simply applied a threshold followed by a re-normalization step, in order to make the distinctive bins of the histograms stronger. The results, shown in Fig. 2.9, achieved almost the 77% in terms of accuracy, outperforming the bi-dimensional co-occurrences histogram. For the evaluation of the threshold, we selected the combination of parameters that had the best performances in the previous step of the tests. Thus, we used $M = 500$ and $H = 6$. It is important to point out that, while the representation proposed by [8] had a cardinality of $K \times K = 2500$ bins, ours is more efficient in terms of space requirements, using only $M = 500$ bins.

Once we found the configuration, we ran the algorithm with the tuned

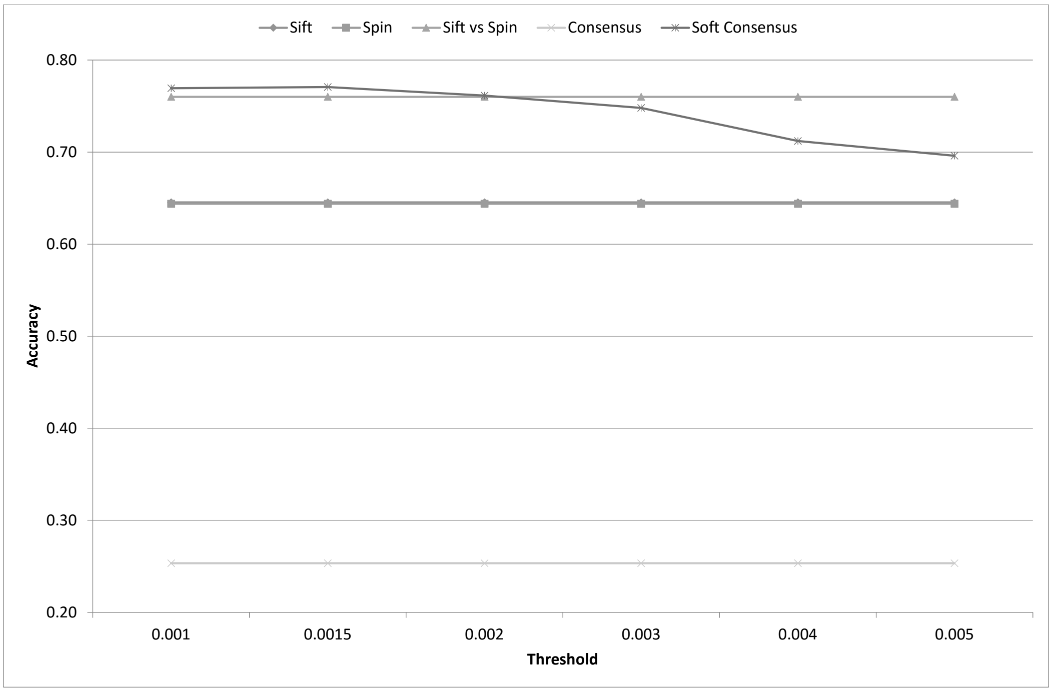


Figure 2.9: Comparison of classification accuracy varying the threshold for soft consensus representation.

parameters using the whole dataset. Unfortunately, the chosen parameters did not fit the scale of the problem, so a new tuning step should have been performed, requiring a lot of time because of the complexity of the approach which, because of its intrinsically sequential nature, could not even be parallelized. We then decided to try the approach on a different dataset (see Section 3.4 and to improve the framework by considering a semi-naive approach instead of the full naive one (see next Section).

2.5 Semi-Naive Consensus Clustering

In order to model the problem in a Semi-Naive way, we relax the Bayesian Naive assumption [28, 29], by grouping the labels and imposing that the labels belonging to the same group are drawn from a probability distribution, while the groups are conditionally independent. In [2], it is assumed that the labels $\mathbf{y}_i$ are conditionally independent, as modeled in Eq. 2.7, in order to make the problem tractable. Thus, given the H labels in $\mathbf{y}_i$, we create S partitions of size $D = \frac{H}{S}$ mutually conditionally independent. Thus, the probability becomes:

$$P_m\left(\mathbf{y}_i|\theta_m\right) = \prod_{s=1}^{S} P_m^{(i)}\left(F_{is}|\theta_m^{(i)}\right). \tag{2.14}$$

In Eq. 2.14, $F_{is} = \left\{y_{i\sigma(s,1)}, y_{i\sigma(s,2)}, \ldots, y_{i\sigma(s,D)}\right\}$ are the labels belonging to the $s-$th group, while $\sigma\left(s,j\right)$, $j = 1, \ldots, D$ is a random permutation function within the range $1, \ldots, H$. The labels F_{is} are dependent, thus the probability $P_m^{(i)}\left(F_{is}|\theta_m^{(i)}\right), s = 1, \ldots, D$ has to be expressed as a joint probability over the elements of $\mathbf{y}_i$ (see Eq. 2.15).

$$P_m^{(i)}\left(F_{is}|\theta_m^{(i)}\right) = P_m^{(i)}\left(y_{i\sigma(s,1)}, y_{i\sigma(s,2)}, \ldots, y_{i\sigma(s,D)}|\theta_m^{(i)}\right) \tag{2.15}$$

We define now an enumeration function T to assign a unique numerical label to each of the elements in F_{is}, $i = 1, \ldots, N$, $s = 1, \ldots, S$. The values of T lie in the range $\{1, \ldots, K(1) \times K(2) \times K(S)\}$. As shown in Section 2.4, $k = 1, \ldots, K(j)$ is an index referring to the labels in the $j-$th clustering.

$$T\left(F_{is}\right) : \{1, \ldots, K(1)\} \times \{1, \ldots, K(2)\} \times \ldots \{1, \ldots, K(D)\} \longrightarrow \mathcal{N} \tag{2.16}$$

We can now formulate the probability density (see Eq. 2.17) for each group F_{is} in the form of a multinomial trial, as in [2].

$$P_m^{(s)}\left(F_{is}|\theta_m^{(s)}\right) = \prod_{k=1}^{K(1)\times K(2)\times K(S)} \vartheta_{sm}(k)^{\delta(T(F_{is})),k)} \tag{2.17}$$

The consensus partition can still be found using the EM algorithm using the new equations formulated above. Thus, the expected values for each component of the hidden variables vectors $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ can be computed from Eq. 2.18 using Eq. 2.17 as the component probability, together with the mixture weights α (Eq. 2.19) and the mixture parameters ϑ (Eq. 2.20).

$$E\left[z_{im}\right] = \frac{\alpha'_m \prod\limits_{s=1}^{S} \prod\limits_{k=1}^{K(1)\times\ldots\times K(S)} \left(\vartheta_{sm}(k)\right)^{\delta(T(F_{is}),k)}}{\sum\limits_{n=1}^{M} \alpha'_n \prod\limits_{s=1}^{S} \prod\limits_{k=1}^{K(1)\times\ldots\times K(S)} \left(\vartheta_{sn}(k)\right)^{\delta(T(F_{is}),k)}} \tag{2.18}$$

$$\alpha_m = \frac{\sum\limits_{i=1}^{N} E\left[z_{im}\right]}{\sum\limits_{i=1}^{N} \sum\limits_{m=1}^{M} E\left[z_{im}\right]} \tag{2.19}$$

$$\vartheta_{sm}(k) = \frac{\sum\limits_{i=1}^{N} \delta\left(T_{is}, k\right) E\left[z_{im}\right]}{\sum\limits_{i=1}^{N} \sum\limits_{k=1}^{K(1)\times\ldots\times K(S)} \delta\left(T_{is}, k\right) E\left[z_{im}\right]} \tag{2.20}$$

2.5.1 Experimental Results

To test our approach, we used the well-known Two Spirals dataset, proposed by Alexis Wieland[1]. The key feature of this dataset is that the points form two spirals as shown in Fig. 2.10. For our experiments, we chose to use 1000 data points.

The experiments have been performed varying the parameters of both the original Naive [2] and the proposed Semi-Naive algorithms. In the first case, the parameters are the number of input clusterings H and the number of clusters K to be generated by the runs of the input clusterings. H takes values in the range $\{5, \ldots, 50\}$, while K varies in the range $\{2, \ldots, 20\}$. In addition to these parameters, the number of groups S has been taken into account, considering the range $\{2, \ldots, 10\}$.

[1] http://www.cs.cmu.edu/Groups/AI/areas/neural/bench/cmu/

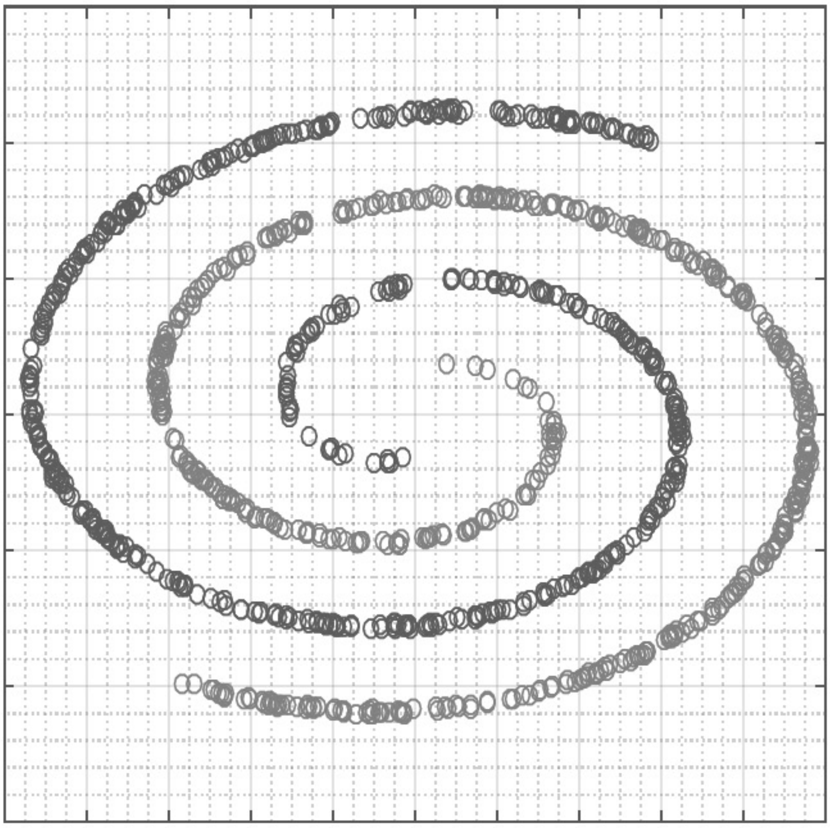


Figure 2.10: Plot of the Two Spirals dataset with 1000 data points.

The results, obtained over 10 different runs of the experiments, are presented in Fig. 2.11 and Fig. 2.12. We computed the accuracy as the ratio between the number of elements correctly classified over the total number of elements to be classified (see Eq. 2.21).

$$\text{Accuracy} = \frac{\#\text{ of elements correctly classified}}{\#\text{ of elements to be classified}} \tag{2.21}$$

As shown in Fig. 2.11, the mean values are not very significant, both for the Naive and Semi-Naive approaches. This is because the algorithms need a fine parameters tuning step, in order to find the combination that best fits the problem. Hence, we considered the best results in terms of accuracy over all the runs and over all the parameters, as shown in Fig. 2.12. The best accuracy obtained for the Naive Bayesian method is 0.634, while for the Semi-Naive approach, the best result is 0.695.

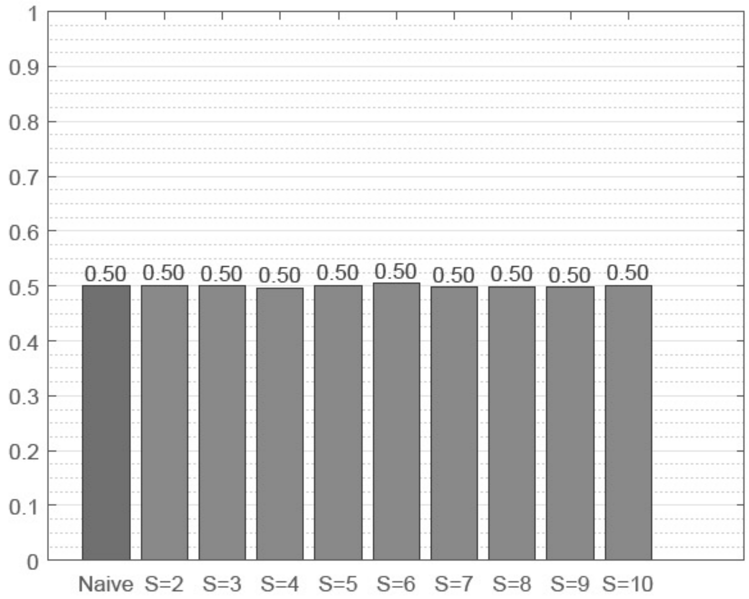


Figure 2.11: Mean accuracies over 10 different runs, averaging over the parameters K and H. The first bar on the left represents the accuracy value obtained with the original Naive method [2], the next bars represent the accuracies obtained using the proposing method varying the number of groups S.

It is interesting to visualize how the two methods partition the plane. The Naive consensus (Fig. 2.13b) splits the plane in two, as a linear classifier would do, while the labeling produced by the Semi-Naive consensus (Fig. 2.13c) has a different behavior.

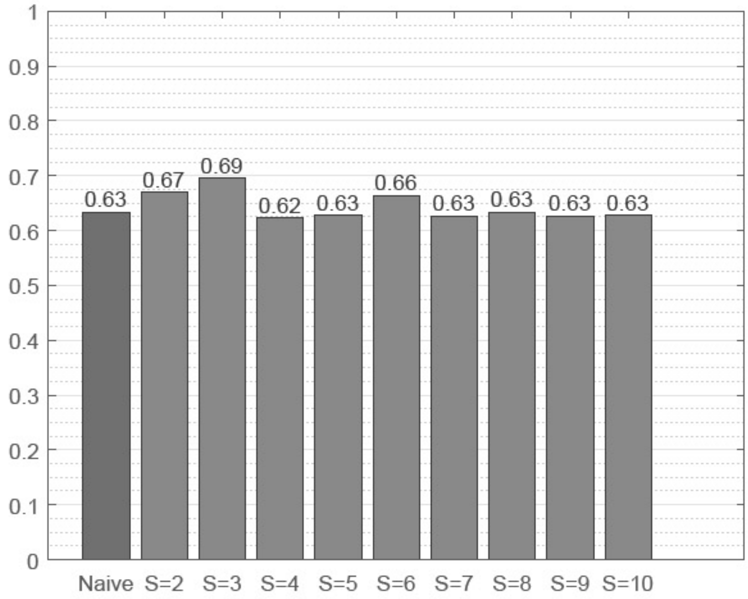
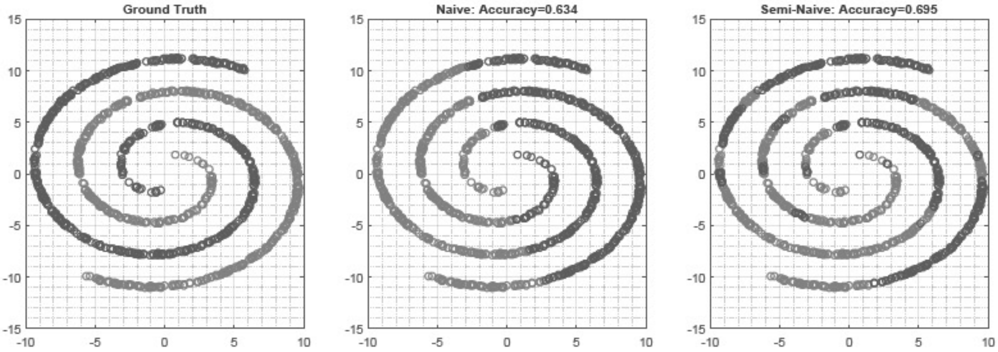


Figure 2.12: Max accuracies over 10 different runs, averaging over the parameters K and H. The first bar on the left represents the accuracy value obtained with the original Naive method [2], the next bars represent the accuracies obtained using the proposing method varying the number of groups S.



(a) (b) (c)

Figure 2.13: Visual Comparison of the results.

Chapter 3

Food Classification

3.1 Introduction and Motivations

It is well-known that a non healthy diet can cause health problems such as obesity and diabetes, as well as risks for people with food allergy. The current mobile imaging technologies (e.g., smartphones and wearable cameras) give the opportunity of building advanced systems for food intake monitoring in order to assess the patients' diet [30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. Related assistive technologies can also be useful to increase the awareness of the society with respect to the quality of life. In this context the ability to automatically recognize images of food acquired with a mobile camera is fundamental to assist patients during their daily meals. Automatic food image retrieval and classification could replace the traditional dietary assessment based on self-reporting that is often inaccurate. As pointed out in different works [40, 41, 42, 40, 43, 44, 45, 46, 47, 48], food understanding engines embedded in mobile or wearable cameras can create food-logs of the daily intake of a patient; these information help the experts (e.g., nutritionists, psychologists) to understand the behavior, habits and/or eating disorders of a patient.

However, food has a high variability in appearance and it is intrinsically deformable. This makes classification and retrieval of food images difficult tasks for current state-of-the-art methods [49, 50, 51], and hence an interesting challenge for Computer Vision researchers. The image representation used to automatically understand food images plays the most important role. Despite many approaches have been published, it is difficult to find works where different representation techniques are compared on the same dataset. This makes difficult to figure out peculiarities of the different techniques, as well as to understand which is the best representation method for food retrieval and classification.

To find a suitable representation of food images it is important to have representative datasets with a high variety of dishes. Although different retrieval and classification methods have been proposed in literature, most of the datasets used so far have not been designed having in mind the study of a proper image representation for food images. Many food datasets are composed by images collected through the Internet (e.g., downloaded from Social Networks), where a specific plate is present just once; there is no way to understand if a specific type of image representation is useful for the classification and retrieval of a specific dish acquired under different points of view, scales or rotation angles. Also the food images collected through the Internet have usually a low resolution and have been processed by the users with artistic or enhancement filters.

The automatic analysis of food images has a long history. The article by Parrish et al. [52], which is probably the first using Computer Vision techniques for a food analysis tasks, dates back to 1977. Looking at the literature in this context, it is quite evident that between 1980s and 2000s the interest on food image understanding was mainly for engineering applications related to the production chain and the assessment of the quality of the marketed food. From the beginning of the new century, with the proliferation of high performance mobile devices, the research has focused more and more on aspects which are strictly related to everyday life, and hence on problems and applications for food intake monitoring.

3.2 State of the art

Food image analysis has a long history. With the aim of giving a survey of the main works in the literature, we have identified four application areas:

- Detection and recognition of food for automatic harvesting;
- Quality assessment of meals produced by industry;
- Food logging, dietary management and food intake monitoring;
- Food classification and retrieval.

Despite most of the “ingredients” involved in the solutions proposed in the different application areas overlap, the main aims of the final systems are different. For instance, if a certain accuracy obtained by a system for the detection and recognition of food for automatic harvesting could be acceptable by a robotic industry, the same accuracy could be not sufficient in systems dedicated to the diet monitoring for patients with diabetes or food allergy. This motivated us in grouping the works in the literature by considering the four aforementioned areas. In Fig. 3.1 is shown a timeline which identifies the periods on which the different areas become of interest and have got highest popularity by taking into account the published papers in literature over the years.

Automatic detection and recognition of fruits and vegetables is useful to enhance robots affordable and reliable vision systems in order to improve the harvesting procedures both in terms of quality and speed. In the late 80s, industrial meals production knew a large scale expansion, so the evaluation of the quality of the produced food with vision systems became an interesting and valuable challenge. From late 90s, the growth of the number of people affected by diseases caused by a non healthy diet, moved the focus to the usage of Computer Vision techniques to help experts (e.g., nutritionists) for the monitoring and understanding the relationships between patients and their meals. This particular researches can take advantage of the huge diffusion on low-cost imaging devices, such as the current smartphones and wearable devices. The large and fast growth of mobile cameras, together with the birth and diffusion of social network services - such as Facebook, Instagram, Pinterest - opened the possibility to upload and share pictures of food. For these reasons, in the past few years, classification and retrieval of food images become more and more popular.

In the following section we will review the state-of-the-art in the field in order to give to the reader an overview of what have been done in the four application domains mentioned above.

3.2.1 Detection and recognition for automatic harvesting

Among the several techniques used for the harvesting of fruits, the more desirable are the ones which do not cause damages to the fruit and/or to the tree. Thus, accurate systems for fruits detection and recognition from images

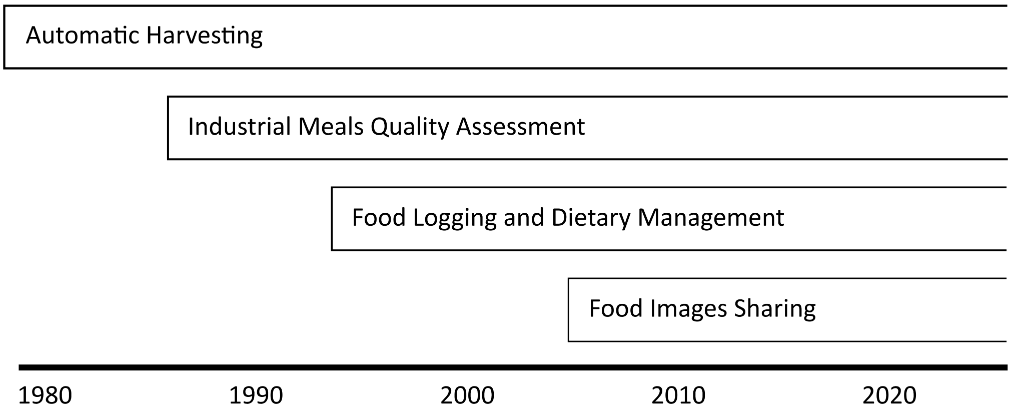


Figure 3.1: Food image analysis task employed during the years.

are needed in order to perform the task correctly. One of the first Computer Vision approach has been designed by Parrish et al. [52], and focuses on apples detection task. The vision system is composed by a B/W camera and an optical red filter. The image is binarized through thresholding operation, then smoothed to suppress noise and artifacts and finally, for each of the segments, the difference between the lengths of the horizontal and vertical extrema are computed, in order to estimate the roundness of the region. Then, the density of the region is computed by placing a window, whose size is determined by the mean value of the extrema, on the centroid. If the density of the region is found to be greater than a preset threshold, the region is accepted as an apple.

In [53], a robot vision system called AID is implemented for oranges recognition. A pseudo-grey image is obtained by means of an electronic filter used to enhance the image. During digitization, 6 bits are used to codify the pixel value which is proportional to the closeness of the actual pixel hue to a preset reference hue. Subsequently, the image is filter using the Sobel operator in order to get a gradient image and a direction image. Finally, the scene interpretation is done through searching for a match with an object model previously stored. This gradient direction template is moved step by step throughout the direction image. Approximately 70% of the visually recognizable fruits were detected.

An orange recognition method, based on color images, is proposed in [54]. Here, Hue and Saturation components of each pixel are employed to form a two-dimensional feature space. Then, two thresholds based on the maximum and minimum values for each component are used as linear classifiers in order to define a square region in the feature plane. Approximately 75% of the pixels were correctly classified. In [55], the same authors extend their earlier study employing a traditional Bayesian classifier, using the RGB values instead of the Hue and Saturation components, with the goal of segmenting the fruit pixels from the background pixels. The tests show that 75% of the pixels are correctly classified.

The Purdue University (USA) and The Volcani Center (Israel) developed a vision system for melon harvesting [56]. A B/W image is analized to locate the melon and estimate its size; this first stage performs an image enhancement, a thresholding, a parameter extraction and hypothesis generation. Shape and texture parameters in the neighborhood of the hypothesized position are computed to obtain the final candidates. Then, a knowledge-based evaluation using rules which allow to avoid noisy detections and to eliminate multiple occurrences is performed. If the second step is not employed, approximately 89% of success and relatively high rates of false detections are found, but when using the knowledge-based rules, 84% and 10% rates are obtained, respectively

A robotic system for greenhouse operation, AGROBOT, was developed at CIRAA in Italy [57]. The vision system used for this project is based on a color camera that supplies the HSI color components. Hue and Saturation

histograms are employed to perform a thresholding to segment the image. The 3-dimensional information is obtained by a stereo-matching of two different images of the same scene. About 90% of the ripe tomatoes are detected and the most frequent errors are due to occlusions.
Jiménez *et al.* [58] built a system for automatic harvesting of spherical fruits. Using a 3D laser scanner, they acquire the whole natural scene. This specific sensor provides the spherical coordinates of each scene point as well as a value indicating the attenuation of the laser energy due mainly to the distance, the surface type and orientation of the sensed surface. So, for each scan, four images are produced as output, representing respectively the azimuth and elevation angles, the distance from the sensor and the attenuation values. Exploiting the sensor model, these images are processed, and taking advantage of the information retrieved by the scanner, four images are produced in output. Of these four, three are actually used for the orange recognition: one is an enhancement of the previous image representing the distance from the sensor, the others encode respectively the apparent reflectance and the reflectance of the surfaces. The image analysis focuses on the last two images. The apparent reflectance image is thresholded to separate the background from the foreground and then the remaining pixels are clustered using the Euclidean distance. The detected clusters without a minimum number of pixels belonging to it are rejected as valid fruit in order to eliminate the possibility of random small areas of a highly reflective non-fruit object. This method, though, is not able to detect fruits whose reflectance is under 0.3. To cope with these kind of items, the Circular Hough Transform is employed on the distance image to detect fruits.

Many other methods have been developed over the years: for an accurate review of this techniques, the reader should refer to [58].

3.2.2 Quality assessment of meals produced by industry

The assessment of the food quality produced by an industry is a crucial task needed to guarantee a good experience to the final customer. Alongside with human control of the product chain, Computer Vision systems can be used to perform the quality assessment through the automatic inspection of images.

In [59, 60, 61], a review of methods for food quality assessment is presented. The authors considers different acquisition systems, the features that can be employed in different tasks, as well as the machine learning algorithms used to perform the decision among the quality of the food items.

In a typical computer vision based pipeline for quality assessment, an image preprocessing, a feature extraction process, and a classification are performed.

Munkevik et al. [62] propose a method to check the validity of industrial cooked meals. In particular, fish burger with peas and smashed potatoes have been considered. As first step, the images of the food are segmented. Then 18 features are extracted from the segmented image, in order to capture different aspects. Specifically, the features are related to the size of the food items on the plate, to the overlapping between different food items, to the shape of the food and to the colors. Eventually, the extracted features are used to train a Self Organizing Feature Map [63], which is employed to learn the model of a meal that fits the standards, and therefore to detect the deviations from this standard model in order to accept or discard a dish.In [64] the approach is refined and extended by considering more food items and employing an Artificial Neural Network (ANN) for classification purposes.

A beans classification system was proposed by Kilic et al. [65] in 2007. For testing purposes, they considered a dataset of images with variable number of beans. The main aim was the assignment of a quality label to each bean. After a segmentation stage using morphological operators, the 1st to 4th order statistics on the RGB channels of the image are computed. Three quality levels for both color and integrity of the sample were defined, but only 5 out of the 9 possible combinations were used to better separate top quality beans from medium and low quality ones. In other words, given a rating from A to C for both colors and integrity, the considered classes are AA, BB, BC, CB, CC. The classification was performed using an ANN, using 69 samples for training and 71 for validation, while the testing set is composed by 371 beans images.

The quality of pizza production has been explored by different researchers. In [66, 67] methods for inspecting shapes, toppings and sauce spread in pizza production are proposed. Different features were computed for the shape, sauce and topping inspection. Specifically, to assess the quality with respect to the shape, the area ratio, aspect ratio, eccentricity, roundness have been considered. For sauce and topping the Principal Component Analysis (PCA) on the histograms computed in the HSV color space have been exploited. The food items are classified considering 5 quality levels concerning the sauce spread and topping, and in 4 quality levels with respect to the shape. The quality classification task was performed using a set of binary Support Vector Machine (SVM) classifier (one-vs-all) organized in a Directed Acyclic Graph (DAG). A food item follows a classification path from the root down to the leaves and then the quality level is assigned. The system is trained using 120 images for the shape, 120 images for the sauce and 120 images for the topping.

Despite the quality assessment and inspection of food is not strictly related to the application domain of dietary food monitoring, we have decided to include information on this application domains such that the reader can have a better overview of what has been done in the context of food image analysis. The inspection of the food quality is usually performed in constrained settings to analyze small number of food classes and variabilities. Usually, simple approaches (e.g., very simple features such as shape measurement) are enough to address the problem and the results claimed by the authors are very good. This scenario is very different from the one where images of food are acquired during meals of a patient or they are downloaded from a social network. The systems for generic food intake monitoring have to deal with a higher number of food classes, mixed food, and a number of image variabilities, such as different environment illumination, different point of view in the acquisition, and different acquisition devices (i.e., different resolution, compression factors, etc). Moreover, usually these systems have to be able to work without prior knowledge. For instance, differently than an industrial production chain where the different ingredients (e.g., to make a pizza) are known in advance, in a generic food image understanding problem there are not a priori assumption by making the task more challenging.

3.2.3 Food logging, dietary management and food intake monitoring

Diet monitoring has a key role for the human health and can help to reduce disease risks such as diabetes. For this reason, since the '70, the computers have been employed to help the medical teams for dietary assessment of the patients. However, the primordial systems for food logging and intake monitoring did not use the Computer Vision; they were calculators for nutrition factors from a predefined food list [68, 69].

During the last century, despite the great steps forward in the knowledge of nutrition, there has been a dramatic increase of food-related illnesses [70]. It has been proved that food diaries are efficient instrument to boost self-awareness of eating habits, and augmenting written diaries with photographs have a more effective impact on the patients. Hence, Computer Vision researchers have put effort in order to provide reliable tools to make the automatic detection and recognition of meals images more accurate.

Among these systems, FoodLog[1] [42, 40, 43, 44, 45] is a multimedia Internet application that enables easy capture and archival of information regarding daily meals. The goal of this framework is to assist the user to keep note of their meals and balance the nutritional values coming from different kinds of food (e.g. carbohydrates, fats, etc.). The user upload the pictures on a remote folder, where the archive is maintained. In [42], the images containing food items are identified by exploiting features related to the HSV and RGB color domain, as well as the shape of the plate. A SVM classifier is trained to detect food images. More specifically, the images are divided in 300 blocks and each block is classified as one of the five nutritional groups defined in the "My Pyramid" model[2] (grains, vegetables, meat & beans, fruits, milk) or as "non-food". In [40] more local features are considered. Color statistics were coupled with SIFT descriptor [4] obtained with three different keypoint selection methods (Difference of Gaussians, centers of grid, centers of circles). In [44] the approach has been extended, adding also a pre-classification step and the personalization of the food image estimator. In [45] the Support Vector Machine is replace by a Naive Bayesian Classifier.

The goal of the approach proposed in [71] is to help people affected by diabetes in following their dietary prescriptions. The authors used object-

[1] `http://www.foodlog.jp`

[2] `http://www.mypyramid.gov/`

related features (color, size, texture and shape) and context-related features (time of the day and user preference). Using an ANN as a classifier, the authors proved that the context information can be exploited to improve the accuracy of the monitoring system.

Food recognition and 3D volume estimation is the goal of the work by Puri et al. [72]. The images, taken under different lighting conditions and poses, are normalized by color and scale, by means of dedicated calibration patterns placed besides the food items. They use an Adaboost-based feature selection method to combine color (RGB and LAB neighborhood) and texture (Maximum Response filters) information, in order to perform a segmentation by classification of the different food items in a dish. The final classifier is obtained as a linear combination of many weak SVM classifiers, one for each feature. Moreover, they reconstruct the 3D shape of the meal using dense stereo matching, after a pose estimation step performed using RANSAC [73].

Chen et al. [74] aim to categorize food from video sequences taken in a laboratory setting. The dishes are placed on a turntable covered with a black tablecloth. They consider an elliptical Region-of-Interest (ROI), inside which they first extracted MSER [75], SURF [76] and STAR [77] features. Since these detectors work on monochrome images, a color histogram in the HSV color space is computed inside the ROI, in addition to the aforementioned detectors, in order to capture the richness of food images in terms of colors. The images are then represented using the Bag of Words paradigm; they create a vocabulary with 10000 visual words using k-means clustering and subsequently each data point is associated with the closest cluster using the Approximated Nearest Neighbor algorithm. For each image, hence, a Bag of Word representation and the color histogram in the HSV color space are provided. The goal is to classify the dish in a specific frame. The authors propose to do that comparing the frame under examination with a frame already classified, in a retrieval-like fashion. To do so, a similarity score is computed separately for the Bag of Words representation and for the color histograms. For the first representation, the term frequency- inverse document frequency (tf-idf) technique is employed; for the second, the correlation coefficient between the $|L_1|$-norm of the histograms is computed. The two scores are then combined with different weights to obtain the global score for the considered frame. Since the calories for the reference dish are known, the similarity is able to coarsely quantify the difference of food in the two frames.

3D reconstruction is used in [78] for volume computation. A disparity map is computed from stereo pairs, and hence a dense 3D points cloud is computed and aligned with respect to the estimated table plane using a specific designed marker. The different food items present in the image are assumed to be already segmented. Each food segment is then projected on the 3D model, in order to compute its volume, which can be defined as the integral of the distance between the surface of each segment and either the plate (identified by its rim and reconstructed shape), or the table (identified by the reference pattern).

Food consumption estimation is also the goal in [37]. The authors propose a wearable system equipped with a camera and a microphone. When the microphone detects a chewing sounds, the Computer Vision part of the framework is activated. The algorithm tries to identify keyframes containing food by using very simple features such as ellipse detection and color histograms. The first step is to perform an ellipse detection. When the ellipse is found, it is split in four quadrants and, for each quadrant, the color his-

Table 3.1: Food Image Datasets. C = Classification, R = Retrieval, CE = Calorie Estimation

Dataset	Related Works	Classes	Images per Class	Total # of Images	Task	Link
UEC FOOD 100	[82, 83, 84, 85, 86, 87, 39]	100	≈ 100	9060	C	`http://foodcam.mobi/dataset.html`
PFID	[49, 50, 88, 89, 90, 51, 91]	101	18	1818	C/R	`http://pfid.intel-research.net/`
FRIDa	[92]	8	ND	877	CE	`http://foodcast.sissa.it/neuroscience/`
NTU-FOOD	[93]	50	100	5000	C	`http://www.cmlab.csie.ntu.edu.tw/project/food/`
Food-101	[94]	101	1000	101000	C	`http://www.vision.ee.ethz.ch/datasets/food-101/`
UNICT-FD899	[46, 48, 95]	899	3/4	3583	R	`http://www.iplab.dmi.unict.it/UNICT-FD889/`
FooDD	[96]	23	ND	3000	CE	`http://www.eecs.uottawa.ca/~shervin/food/`

togram is computed in the C-color space [79]. Then, the difference between the histograms computed over subsequent frames are computed to evaluate the food consumption.

3.2.4 Food classification and retrieval

The approaches we have reviewed so far aim to solve specific food-related task, such as fruit recognition, quality assessment or food logging for dietary management. All of these application domains share a key component related to the recognition of the food. In last years, this aspect has been considered by many computer vision researchers thanks to the increasing availability of large quantity of image data in Internet and the explosion of posts portraying food in social media. This led to the proliferation of datasets with a consistently increasing number of classes and samples. In Table 3.1 we summarize the main features of the publicly available datasets which have been used in the state-of-the-art works in the last years.[3]

In order to recognize food depicted in images, two type of techniques can

[3] Some other datasets have proposed in literature [80, 78, 81, 38]. However these datasets have been not included in Table 3.1 because they are not publicly available. More information on these datasets can be found at URLs `http://www.tadaproject.org` and `http://gocarb.eu`.

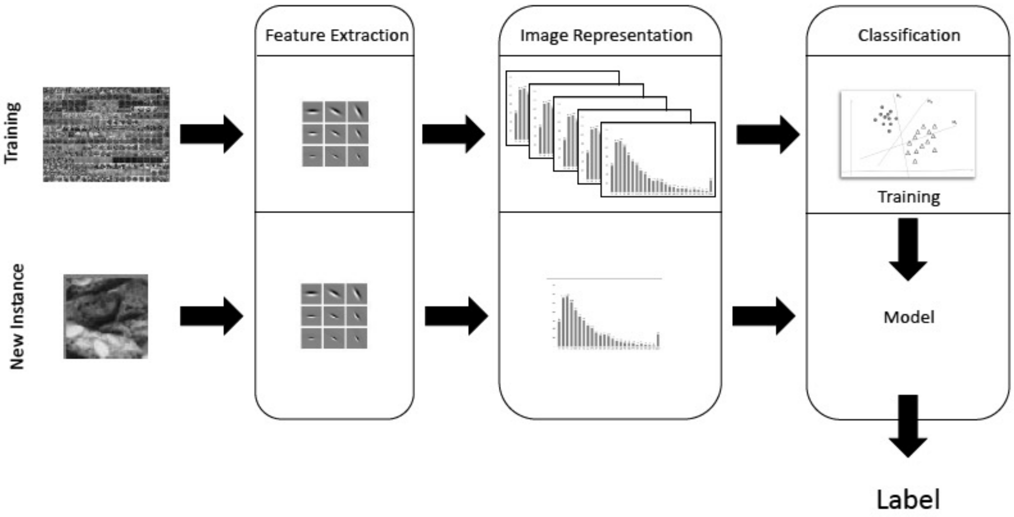


Figure 3.2: Generic food image classification pipeline.

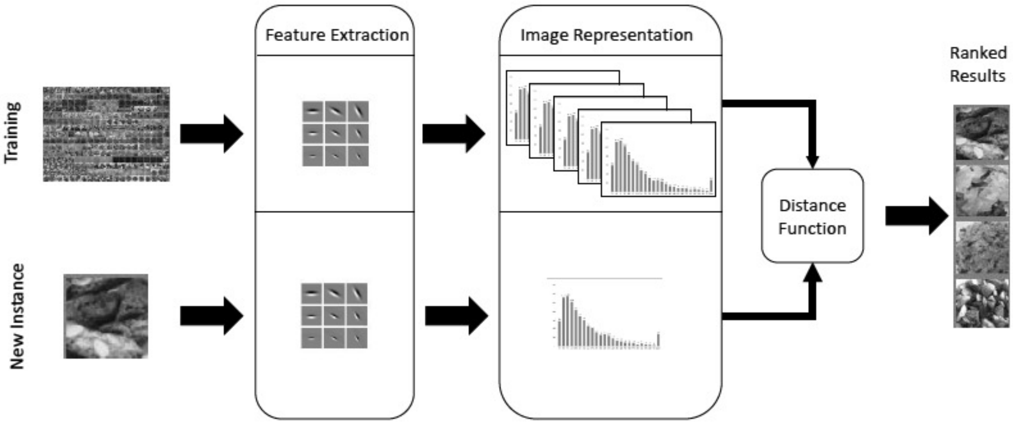


Figure 3.3: Generic food image retrieval pipeline.

be considered: classification and retrieval. In both cases the task is to identify the category of a new food image observation on the basis of a training set of data. The main difference between the two approaches stay in the mechanism used to perform the task. In case of classification the training set is used just to learn the decision function by considering the representation space of the images. Hence, the training images are represented as vectors in a feature space through a transformation function (e.g., Bag of Visual Word approach by considering SIFT or Textons features [11, 97]) and a learning mechanism is used to train a classifier (e.g., a Support Vector Machine) to discriminate food images belonging to different classes. After this phase, the training dataset is discarded and a new observation can be classified by considering the feature space used during the training phase and the trained classification model. In case of retrieval, the training set is maintained and the identification is performed comparing the images through similarity measures (e.g. Bhattacharyya distance [98] after their representation in the feature space. In Fig. 3.2 and Fig 3.3 the generic pipeline for food classification and retrieval are shown.

In [82], a framework for food classification of japanese food is proposed. The approach is trained and tested on a dataset with 50 classes. Three kinds of features are extracted and used: a) Bag of SIFT; b) Color Histograms c) Gabor Filters [99]. The keypoint sampling strategy on which the SIFT descriptor have been computed is implemented with three different ways: using the DoG approach, by random sampling and using a regular grid. To compute Color Histograms, the images are first divided in 2×2 regions, and for each region a 64-bin RGB histogram is calculated. The region-based histograms are then concatenated into a 256-bin. In a similar way, the images are split in 3×3 and 4×4 blocks to compute Gabor Filters responses. The employed Gabor filters take into account four different scales and six orientation, so for the whole image a 216 or 384-dimensional vector arises as result of the extraction step. While Color Histograms and Gabor Filters provide a representation of the images by themselves, SIFT keypoints are clustered generating two different vocabularies with 1000 and 2000 codewords and the images are represented using the Bag of Words paradigm. Summing up, for each image 9 different representation are provided, one coming from the Color Histograms, two from the Gabor Filters with different blocking schemes and six from the combination of sampling strategies and vocabulary size for SIFT features.Classification is performed using a Multiple Kernel Learning SVM (MKL-SVM) [100]. In [83] the dataset is extended up to 85 classes, and 8 variants of Histogram of Oriented Gradients (HOG) [13] are introduced as new features. Moreover, the χ^2 kernel is employed as a kernel function in the MKL-SVM. has been used in [85] where candidate regions are identified using different methods (whole image, Deformable Part Model (DPM) [101], a circle detector and the segmentation method proposed in [102]). The final segmentation arises by integration of the results of the aforementioned techniques.For each candidate region, four sets of features are computed: Bag of SIFT and Bag of CSIFT [103], Spatial Pyramid Representation [10], HOG and Gabor Filters. Then a MKL-SVM is trained for each category, and a score is assigned to every candidate region. The experiments are conducted on images containing both single and multiple food-item. In successive work [84] the same approach is used, but the scores assigned by the classification algorithm are re-arranged applying a manifold learning technique to the candidate regions. The dataset used in [85, 84] is called UEC FOOD 100 and is an extension of the dataset presented in [82, 83]. On this dataset, other approaches have been tested. For instance, pre-trained Convolutional

Neural Networks (CNN) [104] are used in [86] for feature extraction. The CNN features are coded using the Fisher Vectors technique [105], and then the classification is performed by means of SVM. Raví et al. [39] exploited jointly different features in a hierarchy to obtain real-time food intake classification. The hierarchy of features encodes, in some way, the complexity of the images: on simple classes, the classification will rely on the features at the first level, while on more complex classes more features will be used. To represent the images, the Fisher Vector [106] technique is employed, and PCA is applied as in [107]. To perform classification, a linear SVM is trained using the one-vs-rest strategy. The UEC FOOD 100 has been extended to 256 categories in [87] using a so-called "foodness classifier" and transfer learning on images coming from crowd sourcing.

Another dataset used in literature is the Pittsburgh Food Image Dataset (PFID) [49]. This dataset is composed by 4545 still images, 606 stereo pairs, 303 360° videos for structure from motion, and 27 privacy-preserving videos of eating events of volunteers. The images portrays 3 instances of 101 food items, bought in 11 different fast food chains. In [49], a baseline for future experiments is provided. The authors use color histograms and Bag of SIFT features to train a multi-class SVM. In [50], an ingredient based segmentation is performed using a Semantic Texton Forest [108]. Hence, pairwise statistics of local features are computed on the segment connecting two points, and specifically: a) orientation; b) midpoint; c) between-pair; d) distance. Moreover, two joint features are considered (Distance + Orientation and Orientation + Midpoint). A SVM with a χ^2 kernel is employed for classification purpose. The PFID is also used for calories estimation in [88]. SIFT are extracted and a cosine-based distance function is used for matching. Rankings on food categories can be obtained in two ways: 1) a ranking based matching, based on top T items of each frame-based rankings; 2) a count-based matching based on sum of keypoint matching counts over all video frames. Zong et al. [89] locate the keypoints using the SIFT detector, applying the Local Binary Pattern (LBP) [109]. Then they employ a BoW model, using a codeword filtering function to select the most discriminative words in the vocabulary. Dictionary creation is performed in a class-based manner. To provide spatiality, the shape context descriptor [110] is calculated on the image space, considering the words as keypoints. The images are classified by means a cost function which takes into account the Bhattacharyya distance and the shape context matching cost. Nguyen et al. extended the previous mentioned approach introducing the Non-Redundant Local Binary Pattern (NRLBP) [90] and propose two strategies to classify the images: the first makes use of a SVM, the second is based on a cost function. Farinella et al. propose two different approaches on the PFID: one [51] is based on the representation of food images as Bag of Textons. Textons are computed using the responses of MR4 filters, then clustered in a class-based fashion obtaining a visual vocabulary. In the other approach [47] SIFT and SPIN [11] features are computed over a dense grid, and multiple runs of the k-means algorithm are performed separately for SIFT and SPIN. The vocabularies obtained in output are used as input for an Expectation-Maximization based consensus clustering technique [2]. In both approaches, a SVM is used for classification. These two approaches will be further explained in Section 3.3 and Section 3.4. The method proposed in [91] combines different descriptors and statistics calculated on patched centered on the keypoints detected by the Harris-Laplace detector. For each feature, a visual codebook with 1000 words is built, and for each set a gaussian kernel is computed. The resulting kernels are used as input to train a Sequential Minimal Optimization (SMO) MKL-SVM.

Bosch et al. propose a method for food identification based on global and local features [111]. As global features, they use: 1) 1^{st} and $2^{n}d$ moment statistics computed on the color channels of the image; 2) entropy statistics; 3) predominant color statistics. As local features, they consider small patches, on which they calculate the following features: 1) local color statistics; 2) local entropy color; 3) Tamura perceptual features; 4) Gabor filters; 5) SIFT descriptor; 6) Haar wavelets; 7) Steerable filters; 8) DAISY descriptor [112]. While the global features are used as input for a SVM with a RBF kernel, the Bag of Words approach is used with local features. Classification, in this case, is done using a Nearest Neighbor algorithm. This approach was tested on a subset of the dataset created at Purdue University [80]. The Purdue Food Dataset is an extension of the USDA Food and Nutrient Database for Dietary Studies (FNDDS), created having in mind the goal of augmenting *"an existing critical food database with the types of information needed for dietary assessment from the analysis of food images and other metadata"*.

Rahmana et al. in [113] present a dataset with 209 acquired using a iPhone3, to be used for retrieval purposes. They propose, as a baseline, Gabor filter variants to ensure scale and rotation invariance to their algorithm. However, they perform also a classification task, grouping the categories in 5 groups (Bread, Cereal, Veg, Fruit, Fast).

Another system for mobile food recognition is proposed in [114]. Here, color histograms on the RGB space are computed on 3×3 blocks and a dictionary with 500 visual words is built on SURF descriptors, to enclose local features in the general description of the image. To classify the images, a linear SVM with explicit embedding [115] is employed. It is interesting to note that the authors propose a system able to suggest the direction to which the camera should be moved, in order to improve classifier accuracy. Also, a dataset with 50 categories containing 100 images each is presented.

A Computer Vision system for Chinese food identification is presented in [93]. The authors work on a database composed by 50 categories of ready-to-eat Chinese meals, with 100 images per category. On each image, the following features are extracted: 1) SIFT with sparse coding; 2) LBP with multi-resolution sparse coding; 3) color histograms; 4) Gabor textures. A SVM is trained for each feature using 5-fold cross validation; the fusion is done using the Multi-Class AdaBoost algorithm. Marginally, the authors propose also a quantity estimation technique using Microsoft Kinect, but this approach has been tested only on a single item of “hot & sour soup”.

A food recognition system integrated on a chopping board is the topic of the work by Pham et al. [116]. In this work, an imaging system composed by a matrix of optical fibers is placed under an appropriately prepared chopping board. The sensor acquires the image and afterwards a 64-dimensional color histogram and a 64-dimensional vector of Bag of SURF features are computed. The algorithms used to classify the images are kNN and SVM. The training and testing phases make use of a dataset composed by 1800 pictures of 12 food ingredients.

Random Forest (RF) [117] are used in [94] for mining discriminative regions. Superpixels are generated from the images and dense SURF and color histograms are computed and encoded using Fisher Vectors [105]. These descriptors are supplied to the RF for training. Once the RF has been trained, the leaves constitute the set of candidates for the components. Using a probability-based distinctiveness function, the most discriminative leaves are selected. Hence, a linear binary SVM is trained for each class, using the samples lying in the most discriminative leaves as positive samples and hard negative samples to speedup the learning process. Alongside with the algorithm, the authors present a novel dataset, called Food-101, composed by 1000 images for each one of the 101 most popular dishes on `foodspotting.com`.

The UNICT-FD899 [46] has been acquired by users with a smartphone in the last four years during meals (i.e., iPhone 3GS or iPhone 4) in unconstrained settings (e.g., different backgrounds and light environmental conditions). Each dish has been acquired with a smartphone multiple times to introduce photometric (e.g., flash vs no flash) and geometric variability (rotation, scale, point of view changes). The overall dataset contains 3583 images acquired with smartphones. The dataset is designed to push research in this application domain with the aim of finding a good way to represent food images for recognition purposes. The first question the authors try to answer is the following: are we able to perform a near duplicate image retrieval (NDIR) in case of food images? Note that there is no agreement on the technical definition of near-duplicates. The definition of near duplicate depends on the degree of variability (photometric and geometric) that is considered acceptable for each particular application. Some approaches consider as near duplicate images the ones obtained by slightly modifying the original ones through common transformations such as changing contrast or saturation, scaling, cropping, etc. Other techniques (e.g. [118]) consider as near duplicate the images of the same scene but with different viewpoint and illumination. In [46], the authors consider this last definition of near duplicate food images to test different image representations on the proposed dataset. Then, they benchmark the proposed dataset in the context of NDIR by using three standard state-of-the-art image descriptors: Bag of Textons [119], PRI-CoLBP [120] and SIFT [4]. Results confirm that both textures and colors are fundamental properties. The experiments performed point out that the Bag of Textons representation is more accurate than the other two approaches for NDIR.

3.3 Classification using texture-based features

In Chapter 1 the Bag-of-Visual-Word model has been presented as one of the most used paradigms to represent images. Recalling the steps involved in the approach (i.e. feature detection, feature description, codebook generation and proper image representation), it is noteworthy to underline that each of these four steps introduces a variability on the final model used to represent the images, and influences the overall pipeline as well as the results of the classification. Different local feature descriptors can be exploited to generate the codebook. For instance, in [49] SIFT has been used to test BoW paradigm on the PFID dataset. Among the other descriptors, Textons [5] have been employed when the content of the images is rich of textures [119, 97, 121]. Since textures are one of the most important aspects of food images, here we treat the classification of food as a texture classification problem. In the learning stage, training images are convolved with a filter bank to compute filter responses. This feature space is quantized via K-Means clustering and the obtained clusters prototypes (i.e., the visual vocabulary) are used to label each filter response (i.e., each pixel) of the training images. The distribution of Textons is then used to feed the SVM classifier and hence to build the model to be used for classification purpose. During classification phase, test images are represented as distribution on the pre-learned Textons vocabulary after filter bank processing. Each test image, represented as Bag of Textons, is then classified accordingly with the previous learned SVM model. In our experiments we use the Maximum Response filter bank [119] which is composed by filters (Gaussian, first and second derivative of Gaussian and Laplacian of Gaussian) computed at multiple orientation and scales. To achieve rotational and scale invariance, the responses of the anisotropic filters are recorded at the maximum response on both scales and orientations (MRS4 filters). In this way, a very compact 4-dimensional vector for each color channel is associated to every pixel of the food images. As suggested in [119], filters are L_1 normalized so that the filter responses lie approximately in the same range. To achieve invariance to the global affine transformation of the illumination, the intensity of the images is normalized (i.e., zero mean and unit standard deviation on each color channel) before the convolution with the MRS4 filter bank. Finally, the filter response $\mathbf{r}$ at each pixel is contrast normalized as formalized in the following:

$$\mathbf{r}_{final} = \frac{\mathbf{r}\left[\log\left(1 + \frac{\|\mathbf{r}\|_2}{0.03}\right)\right]}{\|\mathbf{r}\|_2} \tag{3.1}$$

Regarding the Textons vocabulary generation, differently than the classic procedure where the feature descriptors extracted from all training images of the different classes are quantized all together, here we consider a class-based quantization [119]. First, a small codebook D_c with K_c Textons is built for each food class c. Then, the learned class-based Textons vocabularies are collected in a single visual dictionary $D = \bigcup_c D_c$ of cardinality $K = \sum_c K_c$, and the food images are represented as visual words distributions considering the vocabulary D. The rationale beyond this codebook generation is similar to the one presented in [14]. Each class-based Textons vocabulary is considered suitable to encode textures of a specific class of food and not suitable to encode the textures of the other classes; this is reflected in the image representation in which all the class-based vocabularies are collected in a single codebook D. Intuitively, when an image of class c is encoded as Textons distribution considering the final vocabulary D, the bins of the sub-vocabulary D_c are more expressed than the bins related to the other sub-vocabularies $D_{c'}, c' \neq c$, making the representation more discriminative. The experiments reported in Subsec. 3.3.1 show that, considering the PFID dataset, the class-based Textons representation achieve better results than the one learned without considering the different food classes during the codebook generation. For classification purpose, we use a multiclass SVM with a pre-computed kernel by considering the cosine distance. Given two Bag of Textons signatures S_{I_i}, S_{I_j} , the cosine distance d_{cos} is calculated as following:

$$d_{cos}\left(S_{I_i}, S_{I_j}\right) = 1 - \frac{S_{I_i} S'_{I_j}}{\sqrt{\left(S_{I_i} S'_{I_j}\right)\left(S_{I_j} S'_{I_i}\right)}} \tag{3.2}$$

The kernel is defined as:

$$k_{\cos}\left(\mathbf{S}_{I_i}, \mathbf{S}_{I_j}\right) = e^{-d_{\cos}\left(\mathbf{S}_{I_i}, \mathbf{S}_{I_j}\right)}. \tag{3.3}$$

3.3.1 Experimental settings and results

This method have been compared against the techniques reported in [49, 122] on the PFID dataset [49]. As in [49, 122], we follow the experimental protocol defined for the PFID dataset: 3-fold cross-validation using 12 images from

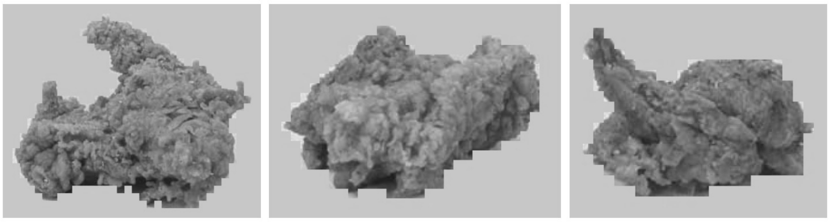


Figure 3.4: Three different classes of the PFID dataset. Left: Crispy Chicken Breasts. Middle: Crispy Chicken Thighs. Right: Crispy Whole Chicken Wing.

two instances of each class for training, and the 6 remaining images of the third instance of each class for testing. We employed the libSVM library [123] to assess the class-based Bag of Textons representation described in the previous section.

Vocabulary Size	**610**	**1220**	**1830**	**2440**
Class-based Textons	27.9%	29.1%	29.4%	31.3%
Global Textons	23.1%	25.3%	26.0%	26.2%

Table 3.2: Class-based vs Global Textons Vocabularies. In all settings class-based vocabulary achieve better results.

As in [122], we have also performed tests by re-organizing the 61 PFID food categories into seven major groups: Sandwiches, Salads & Sides, Chicken, Breads & Pastries, Donuts, Bagels, and Tacos. As first test, we have compared the class-based Textons vocabulary with respect to the global one, i.e., the one obtained considering all the feature descriptors of the different classes all together during quantization. Tab. 3.2 reports the results in terms of accuracy at varying of the vocabulary size for the classification of the 61 classes of the PFID dataset. The size of the vocabulary has been fixed by considering the number of class-based Textons K_c to be learned for each food class. We have considered $K_c \in \{10, 20, 30, 40\}$ Textons for each class c, corresponding to a final vocabulary size of $K \in \{610, 1220, 1830, 2440\}$. As expected, increasing the number of Textons, the classification accuracy improve. Nevertheless, we do not have further improvements by considering more than 40 Textons per class. Note that the class-based vocabulary achieve better results in all cases. The comparison of the class-based Bag of Textons representation (with $K_c = 40$) against to the others state-of-the-art methods [122, 49] is shown in Fig. 3.5 and Fig. 3.6 for both the 61 classes and the 7 major classes respectively.

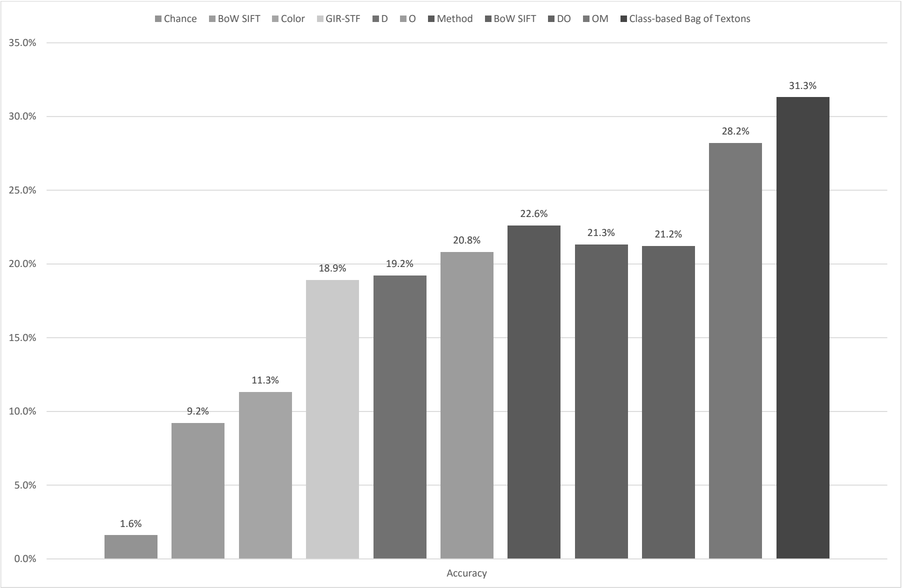


Figure 3.5: Classification accuracy (%) – 61 classes.

The names of the different methods are related to the original name used by the authors in their papers. The chance recognition rate is also indicated.

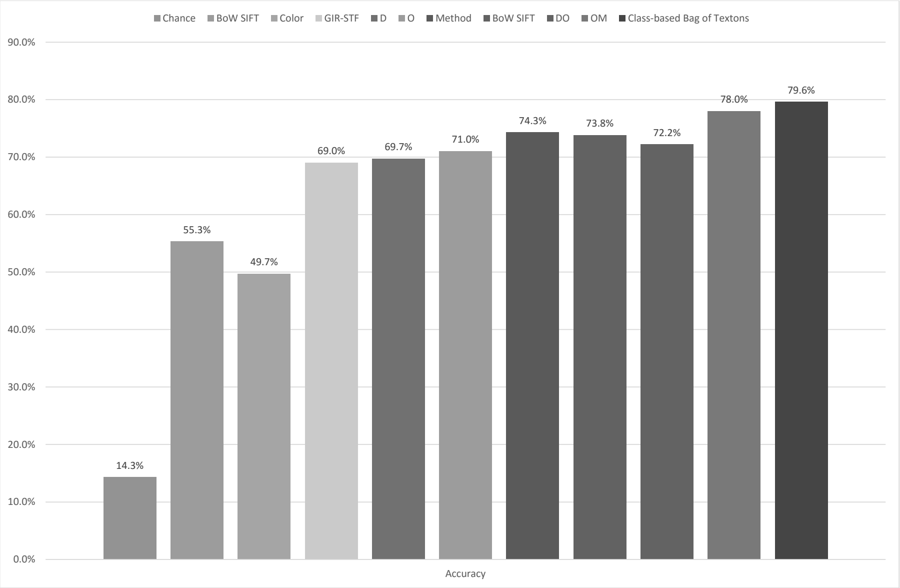


Figure 3.6: Classification accuracy (%) – 7 classes.

The classification accuracy of the class-based Bag of Textons representation was 31.3% for the 61 classes and 79.6% for the 7 major classes. Although its simplicity, the class-based Bag of Textons representation achieve much better results (¿ 20%) than the global BoW considering SIFT descriptor. It also outperforms the method proposed in [122] where OM features encoding spatial information are used after a semantic segmentation performed trough STF [108]. It is important to note that, differently than [122], Textons based representation does not require any manual labeling of the different ingredients composing the food items to be employed. Although the labeling of the different food ingredients is possible for a small set of plates, the up-scaling to a huge number of categories (composed by many ingredients) became not feasible, making the approach described in [122] difficult to be applied. The

Table 3.3: Per-Class accuracy of the different methods on the 7 Major Classes of the PFID dataset. In each row, the two highest values are underlined, while the maximum is reported in **bold**.

Class	**Per-Class Accuracy % (# of images)**				
	Color [49]	BoW SIFT [49]	GIR-STF [122, 108]	OM [122]	Class-based Bag of Textons
Sandwich	69.0 (157.3)	75.0 (171)	79.0 (180.1)	86.0 (196.1)	**87.6 (199.7)**
Salad & Sides	16.0 (5.8)	45.0 (16.2)	79.0 (28.4)	**93.0 (33.5)**	84.3 (30.3)
Bagel	13.0 (3.1)	15.0 (3.6)	33.0 (7.9)	40.0 (9.6)	**70.8 (17)**
Donut	0.0 (0)	18.0 (4.3)	14.0 (3.4)	17.0 (4.1)	**43.1 (10.3)**
Chicken	49.0 (11.8)	36.0 (8.6)	73.0 (17.5)	**82.0 (19.7)**	66.7 (16)
Taco	39.0 (4.7)	24.0 (2.9)	40.0 (4.8)	65.0 (7.8)	**69.4 (8.3)**
Bread & Pastry	8.0 (1.4)	3.0 (0.5)	47.0 (8.5)	**67.0 (12.1)**	53.7 (9.7)
Average	27.7 (26.3)	30.9 (29.6)	52.1 (35.8)	64.3 (40.4)	**67.9 (41.6)**

experiments point out that a proper encoding of textures play an important role for food classification. Note that, even considering only a few Textons per class (i.e., 10 Textons for a total of 610 visual word – see Tab. 3.2 and Fig. 3.5) the accuracy obtained by the proposed method on the 61 classes (27.9%) outperforms the ones achieved by other methods and is very close to a more complex food classification pipeline described in [122] (28.2%).The proposed representation outperform all the others methods with a number of class-based Textons $K_c \geq 30$. In Tab. 3.3 are reported the accuracies of the different methods on the 7 major classes of the PFID dataset. Since the number of images belonging to the different classes are not balanced, for a better understanding of the results, the number of images is reported together with the per-class accuracy. Also in the case of 7 major classes the average per-class accuracy is in favor of the Textons based representation.

3.4 Classifications using consensus vocabularies

As an application of the study presented in Chapter 2, we applied the consensus vocabulary representation model to food images for classification purposes. Recalling the theoretical framework, we define

$$\mathbf{v}_n^{(i)} = \left(V_1\left(x_n^{(i)}\right), \ldots, V_H\left(x_n^{(i)}\right)\right)$$

as the vector that contains all the *ids* labels for the interesting point x_n^i. Considering the set of all vectors $\mathbf{v}_n^{(i)}$, the consensus clustering algorithm is used to find a consensus partition V_c called the *Consensus Vocabulary.*
The original formulation of the consensus clustering assigns each vector $\mathbf{v}_n^{(i)}$ to the most likely cluster of the consensus partition in a hard way. Taking into account possible visual words ambiguities [124, 125], we use a soft assignment. Specifically, we employ the probability vector $\mathbf{z}_n^{(i)}$ given by the consensus algorithm to establish the membership degree of each vector $\mathbf{v}_n^{(i)}$ to the different consensus clusters. Every image I_i is hence represented as the normalized sum of all the $\mathbf{z}_n^{(i)}$:

$$\mathbf{S}_{I_i} = \frac{1}{N_i}\sum_{n=1}^{N_i}\mathbf{z}_n^{(i)} \tag{3.4}$$

To represent test images $\overline{I}_i$, we first project their interesting points in the set of vocabulary $\mathbf{V}$, and then the consensus vocabulary is used to compute the final signature in the same way as for the training images (Eq. 3.4).

To perform the classification, a multiclass SVM with a pre-computed kernel and cosine distance is used, as shown in Equations 3.2 and 3.3.

To assess the proposed approach we have used the PFID dataset [49]. Our method has been compared against the two baseline methods reported in [49] as well as with respect to the different methods proposed in [122]. As in [122], we followed the experimental protocol of [49] by performing a 3-fold cross-validation for our experiments. We used 12 images from two instances of each class for training and the 6 remaining images of the third instance for testing.

A dense sampling procedure to extract the descriptors has been considered by using a spatial grid with steps of 8 pixels in both horizontal and vertical directions. The descriptors are computed on patches of 24×24 pixels centered on each point of the spatial grid. The visual vocabularies to be used as input for the consensus clustering have been obtained considering three different runs of the K-means clustering for each descriptor. We have used $K = 200$ on each run with a random initialization. So, each point into the spacial grid of the dense sampling has been projected into the 6-dimensional feature space of the computed visual vocabularies (3 on the SIFT features and 3 on the SPIN features). For the final consensus vocabulary, we chose a size of 300 consensus words. This means that the final food image is represented with a very small vector. After representing images as described in Section 3.4, we trained the SVM classifier, using the training images and pre-computed kernel with cosine distance. The trained classifier has been then employed on the test images. The classification accuracy achieved employing consensus vocabularies on the 61 classes is reported in Fig. 3.7a, along with the accuracies of the compared state-of-the-art approaches. The low accuracy in discriminating among the 61 different classes is mainly due to foods items of the PFID dataset have very similar appearances (and similar ingredients) despite they belong to different classes [51].

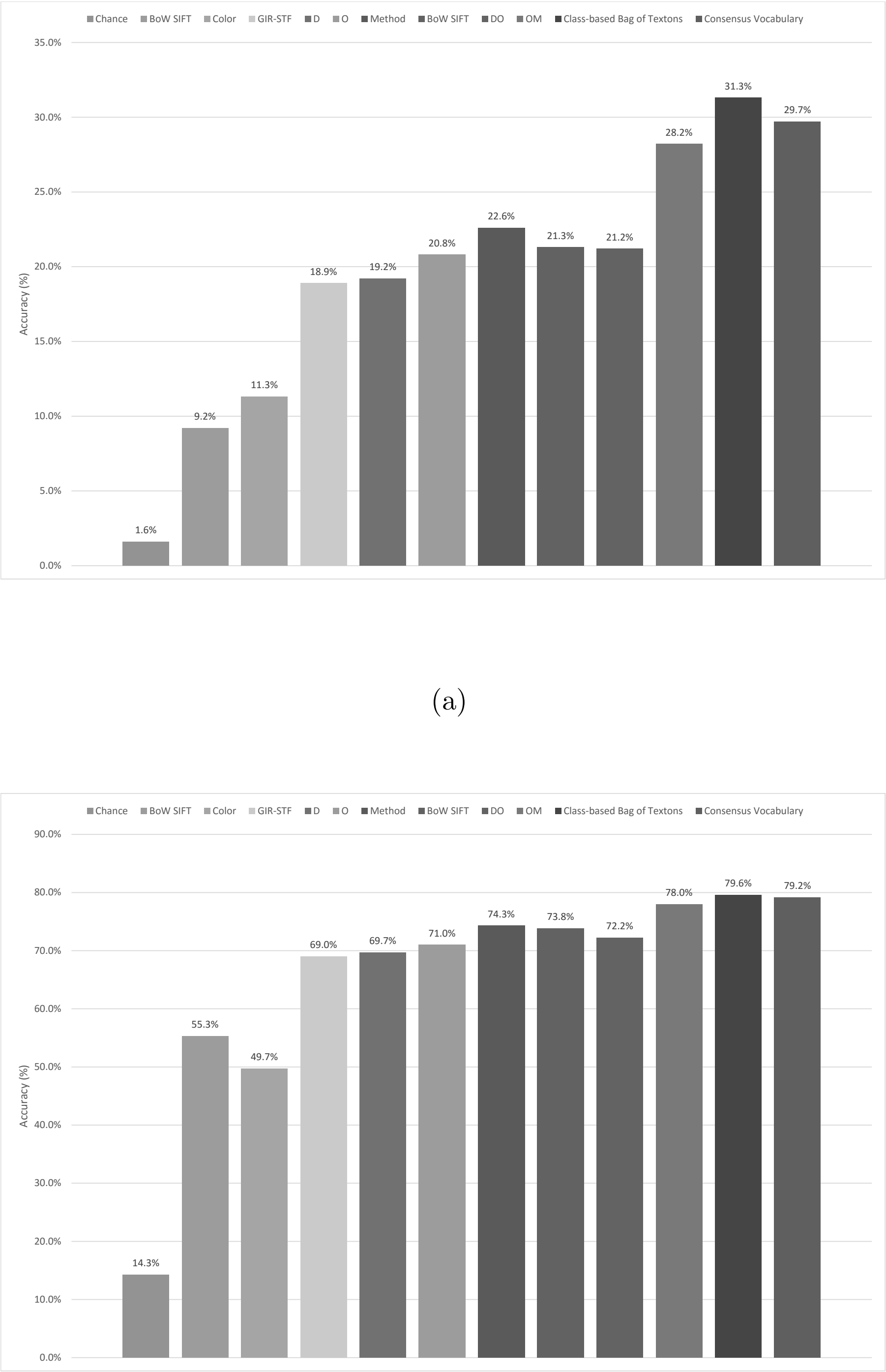
It is important to note that our method, differently than [122], does not need any manual labeling of the different ingredients composing the food items to be employed to produce the representation. Although the labeling of the different food ingredients is possible for a small set of plates, the up-scaling to a huge number of categories (composed by many ingredients) became not feasible, making the approach described in [122] difficult to be applied.

As in [122], we have also performed tests re-organizing the 61 PFID food categories into seven major groups (e.g. sandwiches, salads and sides, chicken, breads and pastries, donuts, bagels, tacos). Results obtained by the different approaches are reported in Fig. 3.7b. In Tab. 3.4 the per-class accuracies of the results of the different methods on the seven major classes of the PFID dataset are reported. Since the number of images belonging to the different classes is not balanced, for a better understanding of the results, the number of images is reported together with the per-class accuracy. Also in this case, our approach obtains better performances with respect to the best performing one proposed in [122].

We want also to underline that, despite the approach in [51] has better results in terms of accuracy, the proposed method is valuable under a theoretical perspective. In fact, it shows that the results obtained using a combination of different features are almost as good as standard techniques, but it captures different aspects of the image, such as local gradient and textures. Note that the representation proposed in [51] can be exploited together with SIFT and SPIN to build a novel Consensus vocabulary which takes into account the power of Textons in representing patterns. Moreover, the final vocabulary size used in this approach is much lower than the one used in [51].

Table 3.4: Per-Class accuracy of the different methods on the 7 Major Classes of the PFID dataset. In each row, the two highest values are underlined, while the maximum is reported in **bold**.

Class	**Per-Class Accuracy % (# of images)**					
	Color [49]	BoW SIFT [49]	GIR-STF [122, 108]	OM [122]	Bag of Textons [51]	Consensus Vocabularies
Sandwich	69.0 (157.3)	75.0 (171)	79.0 (180.1)	86.0 (196.1)	87.6 (199.7)	**89.0 (203)**
Salad & Sides	16.0 (5.8)	45.0 (16.2)	79.0 (28.4)	**93.0 (33.5)**	84.3 (30.3)	69.4 (25)
Bagel	13.0 (3.1)	15.0 (3.6)	33.0 (7.9)	40.0 (9.6)	**70.8 (17)**	62.5 (15)
Donut	0.0 (0)	18.0 (4.3)	14.0 (3.4)	17.0 (4.1)	**43.1 (10.3)**	29.2 (7)
Chicken	49.0 (11.8)	36.0 (8.6)	73.0 (17.5)	82.0 (19.7)	66.7 (16)	**91.7 (22)**
Taco	39.0 (4.7)	24.0 (2.9)	40.0 (4.8)	65.0 (7.8)	**69.4 (8.3)**	50.0 (6)
Bread & Pastry	8.0 (1.4)	3.0 (0.5)	47.0 (8.5)	**67.0 (12.1)**	53.7 (9.7)	66.7 (12)
Average	27.7 (26.3)	30.9 (29.6)	52.1 (35.8)	64.3 (40.4)	**67.9 (41.6)**	65.5 (41.4)



(a)

(b)

Figure 3.7: Classification accuracy on the 61 categories (3.7a) and on the 7 major classes (3.7b) of the PFID dataset.

Appendices

Appendix A

Image Forensics

A.1 Introduction

One of the most common problems in the image forensics field is the reconstruction of the history of an image or a video [126]. The data related to the characteristics of the camera that carried out the shooting, together with the reconstruction of the (possible) further processing, allow us to have some useful hints about the originality of the visual document under analysis. For example, if an image has been subjected to more than one JPEG compression, we can state that the considered image is not the exact bitstream generated by the camera at the time of shooting. In a digital investigation that includes JPEG images (the most widely used format on the network [127] and employed by most of cameras [128, 129]) as evidences, the classes of problems that we have to deal with, are essentially related to the authenticity of the visual document under analysis and to the retrieval of the device that generated the image under analysis. About the possibility to discover image manipulations in JPEG images, many approaches can be found in literature, as summarized in [130] and [131]. A first group of works (JPEG blocking artifacts analysis [132, 133], hash functions [134], JPEG headers analysis [129], thumbnails analysis [135], Exif analysis [136], etc.) proposes methods that seek the traces of the forgeries in the structure of the image or in its metadata. In [93] some methods based on PRNU (Photo Response Non-Uniformity) are exposed and tested. This kind of pattern characterizes, and allows to distinguish, every single camera sensor. Other approaches, as described in [137, 138, 139], take care of analyzing the statistical distribution of the values assumed by the DCT coefficients. The explosion in the usage of Social Network Services (SNSs) enlarges the variability of such data and presents new scenarios and challenges.

A.2 Motivation and Scenarios

Investigators nowadays make extensive use of social networks activities in order to solve crimes[1][2]. A typical case involves the need to identify a subject: in such a scenario, the information provided by the naming conventions of Facebook[3], jointly with the possible availability of devices, can help the investigators in order to confirm the identity of a suspect person. More about Social Network Forensic can be read in [140]. Another interesting scenario consider the detection of possible forgeries, in order to prove the authenticity of a picture. Kee and Farid in [129] propose to model the parameters used in the creation of the JPEG thumbnail[4] in order to estimate possible forgeries, while Battiato *et al.* in [134] use a voting approach for the same purpose. For

[1] http://edition.cnn.com/2012/08/30/tech/social-media/fighting-crime-social-media/

[2] http://www.usatoday.com/story/news/nation/2015/03/20/facebook-cracks-murder-suspect/25069899/

[3] http://facebook.com

[4] http://www.w3.org/Graphics/JPEG/

this task, the information inferred from this study can provide some priors to exclude or enforce such hypotheses.

Our analysis will focus on Facebook, because its pervasive diffusion[5] makes it the most obvious place to start for such a study.

A.3 Dataset

As previously stated, we refer in this phase to the Facebook environment, taking into account capabilities, data and related mobile applications available during the experimental phase.

In order to exploit how Facebook manages the images uploaded by the users, we decided to build a dataset, introducing three types of variability: the acquisition device, the input quality (in terms of resolution and compression rate) and the kind of scene depicted. Specifically we used the following imaging devices (see Fig. A.1), which are respectively a reflex camera, a wearable camera, a camera-equipped phone and a compact camera:

- Canon EOS 650D with 18-55 mm interchangeable lens - Fig. A.1a;
- QUMOX SJ-4000 - Fig. A.1b;
- Samsung Galaxy Note 3 Neo - Fig. A.1c;
- Canon Powershot A2300 - Fig. A.1d.

The considered scenes are 3 (i.e. indoor, natural outdoor, artificial outdoor); for each scene we choose 10 frames, keeping the same point of view when changing the camera. Moreover, we took each frame 2 times, changing the camera resolution (see Tab. A.1). The whole dataset is composed by 240 pictures.

Camera	Low Resolution (LR)	High Resolution (HR)
Canon EOS 650D	720 × 480	5184 × 3456
QUMOX SJ4000	640 × 480	4032 × 3024
Samsung Galaxy Note 3 Neo	640 × 480	3264 × 2448
Canon Powershot A2300	640 × 480	4608 × 3456

Table A.1: Resolution settings for the different devices (in pixels).

(a) (b) (c) (d)

Figure A.1: The cameras used to build the dataset.

Facebook actually provides two uploading options: the user can choose between low quality (LQ) and high quality (HQ). We uploaded each picture twice, using both options, and subsequently we downloaded them.
The whole dataset with both original pictures and their downloaded versions is available at http://iplab.dmi.unict.it/UNICT-SNIM/index.html. A subset is shown in Fig. A.2.

A.4 Social Network Image Analysis

A.4.1 Facebook resizing algorithm

Our first evaluation focus on if and how Facebook rescales the uploaded images. We implemented a tool to ease the upload/download process of

[5] http://newsroom.fb.com/company-info/

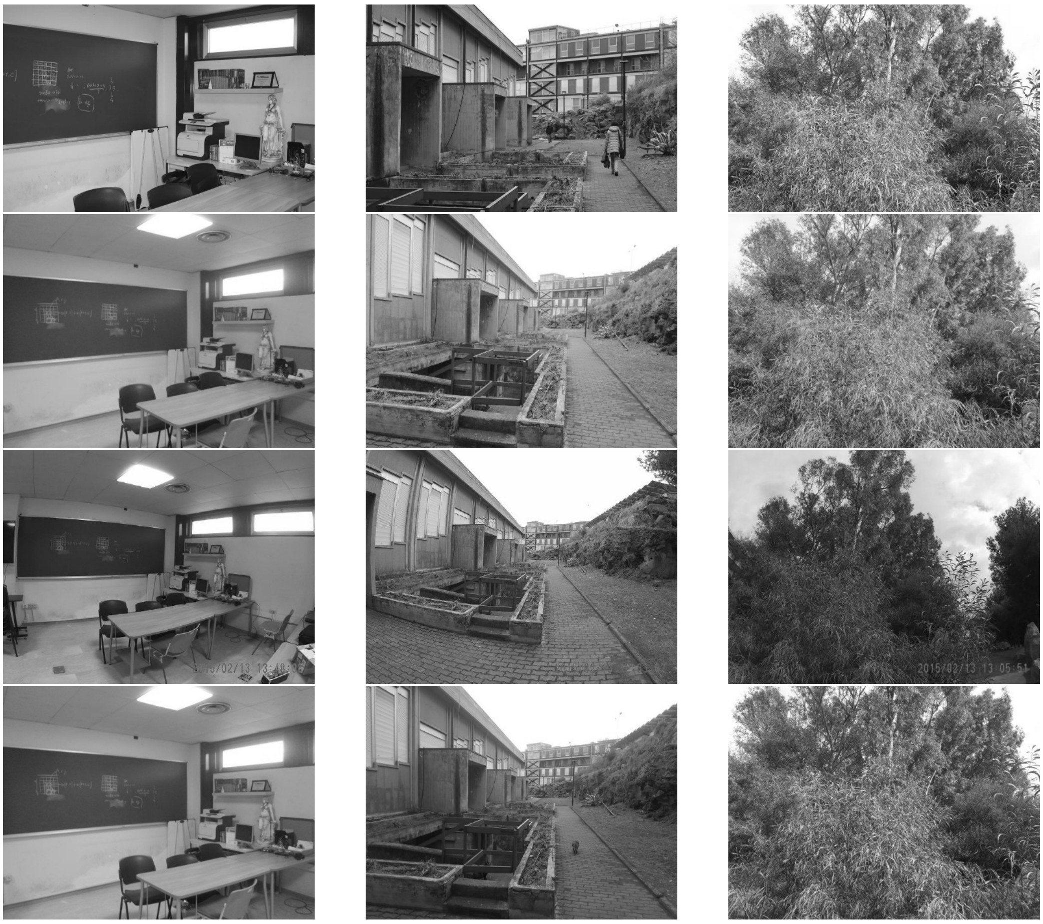


Figure A.2: Column 1: indoor, column 2: outdoor artificial, column 3: outdoor natural. Row 1: Canon EOS 650D, Row 2: QUMOX SJ4000, Row 3: Samsung Galaxy Note 3 Neo, Row 4: Canon Powershot A2300

the images. The different resolutions, related to the devices, are shown in Tab. A.1. Performing a fine-grained tuning using synthetic images, we found out that the resizing algorithm is driven by the length in pixels of the longest side of the uploaded image coupled with the high quality option (on/off). Figure A.3 report the overall flow of the resizing pipeline. Let I be a picture of size $M \times N$. If $max\,(M, N) \leq 960$, I will not be resized; if $960 \leq max\,(M, N) \leq 2048$ and the user selected the HQ upload option, I

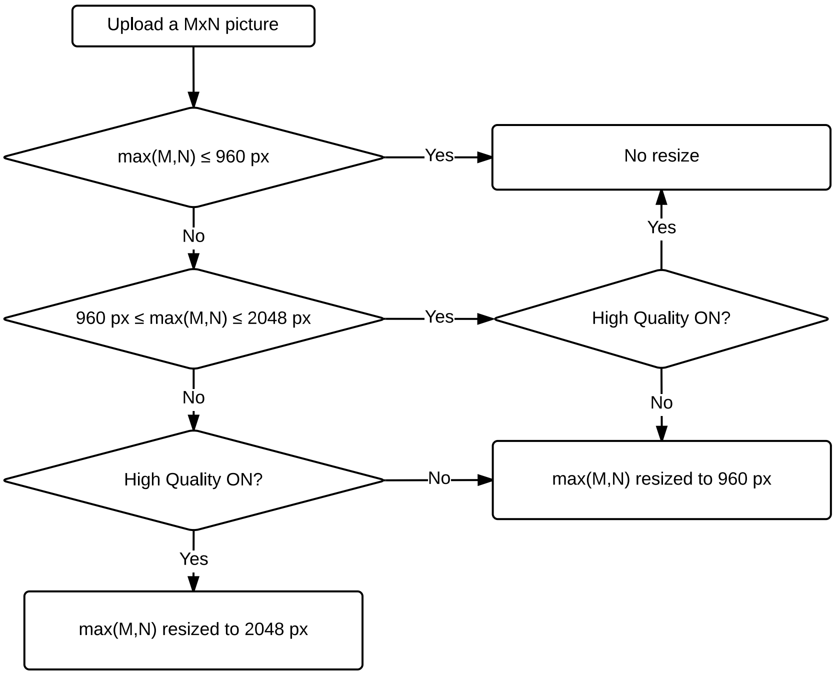


Figure A.3: Work-flow of Facebook resizing algorithm for JPEG images.

will not be resized; if the user did not select the HQ option, then I will be scaled in such a way that the resulting image I' will have its longest side equal to $max\,(M', N') = 960$ pixels. If $max\,(M, N) > 2048$ Facebook scales I both in the case the HQ option is switched on or not. In the first case, the scaled image I' will have its longest side equal to 2048 pixels; in the second case, the longest side will be scaled down to 960 pixels.

Naming of the files

Facebook renames the image files after the upload. Nevertheless, it is still interesting to do a brief analysis on how this renaming is performed, in order to discover patterns in the name of the file and potential relationships among the different elements involved in the upload process: the user, the image

itself, the options.
We found that the generated name is composed by three numeric parts: the first e and the third ones are random generated IDs, while the second part corresponds to the photo ID (see Fig. A.4).

$$\underbrace{10996172}_{\text{Random}}\_\underbrace{745317175583308}_{\text{Photo ID}}\_\underbrace{2711057934783350229}_{\text{Random}}\_(\mathrm{n}|\mathrm{o})].\mathrm{jpg}$$

Figure A.4: The filename generated for an uploaded picture.

The photo ID can be used to retrieve several information about the picture, using for instance the Facebook OpenGraph tool[6]. Just using a common browser and concatenating the photo ID to the OpenGraph URL, it is possible to discover:

- The direct links to the picture;
- The description of the picture;
- The URL of the server where the picture is hosted;
- The date and time of the creation;
- The date and time of last modification;
- The name and the ID of the user (both personal profile or page) who posted the photo;
- The name(s) and ID(s) of the user(s) tagged in the picture;
- Likes and comments (if any).

Moreover, OpenGraph shows the locations of all the copies at different resolutions of the picture, created by Facebook algorithms to be used as

[6] http://graph.facebook.com

thumbnails to optimize the loading time.
It is also interesting to note that the resizing algorithm adds a suffix to the name of the file, depending on the original dimensions and on the upload quality option. Specifically, if the dimensions are beyond the thresholds set in the resizing algorithm and the high quality option is selected, the suffix "_o" will be added; otherwise the added suffix will be "_n".

A.4.2 Quantitative measures

In this Section, we show how the processing done after the upload modify the Bits Per Pixel and the Compression Ratio for the images in the dataset. BPP are calculated as the ratio between the number of bits divided by the number of pixels (Eq. A.1); CR, instead, is computed as the number of bits in the final image divided by the number of bits in the original image (Eq. A.2). It is possible to compute the CR of a single image simply considering the uncompressed 24-bit RGB bitmap version.

$$BPP = \frac{\#\text{ bits in the final image}}{\#\text{ pixels}} \tag{A.1}$$

$$CR = \frac{\#\text{ bits in the final image}}{\#\text{ bit in the original image}} \tag{A.2}$$

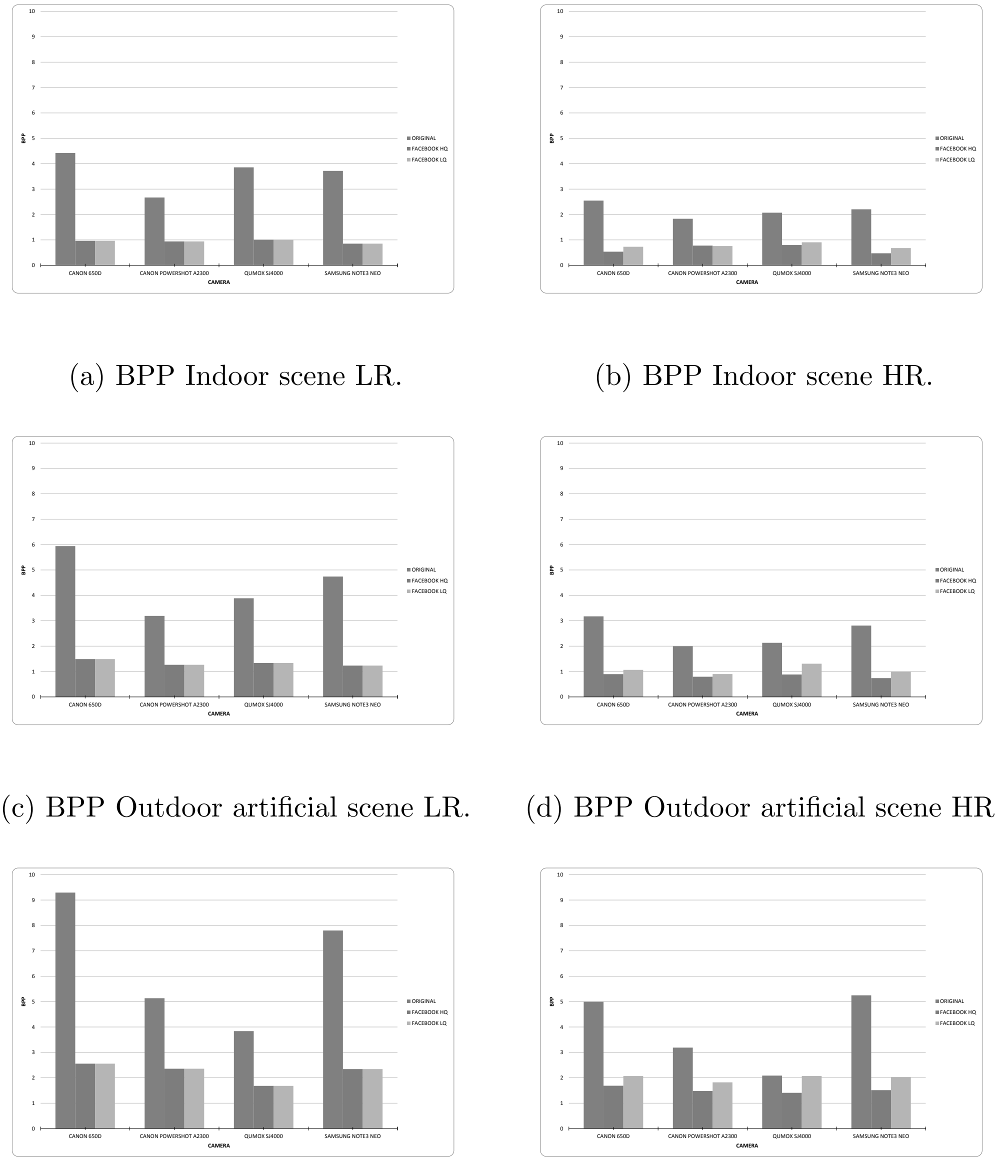
Eq. A.3 is a trivial proof that BPP and CR are proportional.

$$BPP \cdot \#\text{ pixels} = CR \cdot \#\text{ bits in the original image} =$$

$$= \#\text{ bits in the final image}$$

$$BPP = CR \cdot \frac{\#\text{ bits in the original image}}{\#\text{ pixels}} \tag{A.3}$$

The charts in Fig. A.5 report the average BPPs for the images, grouped by scene, which have been taken with the same camera, distinguished depending on the acquisition resolution. Since BPP and Compression Rate are proportional, we refer the reader to the supplementary material [7] for the charts related to CR.



(a) BPP Indoor scene LR.

(b) BPP Indoor scene HR.

(c) BPP Outdoor artificial scene LR.

(d) BPP Outdoor artificial scene HR.

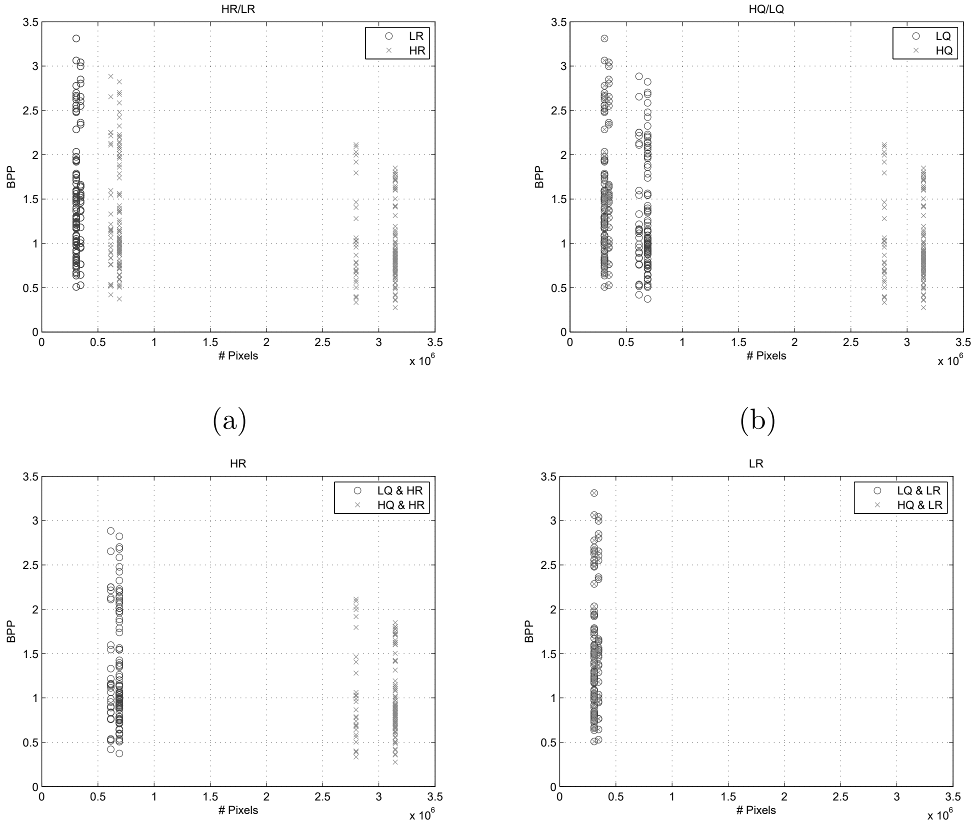
(e) BPP Outdoor natural scene LR.

(f) BPP Outdoor natural scene HR.

Figure A.5: BPP comparison with respect to scene and original resolution.

[7] http://iplab.dmi.unict.it/UNICT-SNIM/index.html

In Fig. A.6 and A.7 we reported the relation of the number of pixels respectively with the BPP and the Quality Factor (QF) as estimated by JPEG Snoop[8]. Observing the graph in Fig. A.6, it emerges a relation of inverse proportionality between the number of pixels and the maximum BPP; this would support the hypothesis of a maximum allowed size for the uploaded images.



(a)

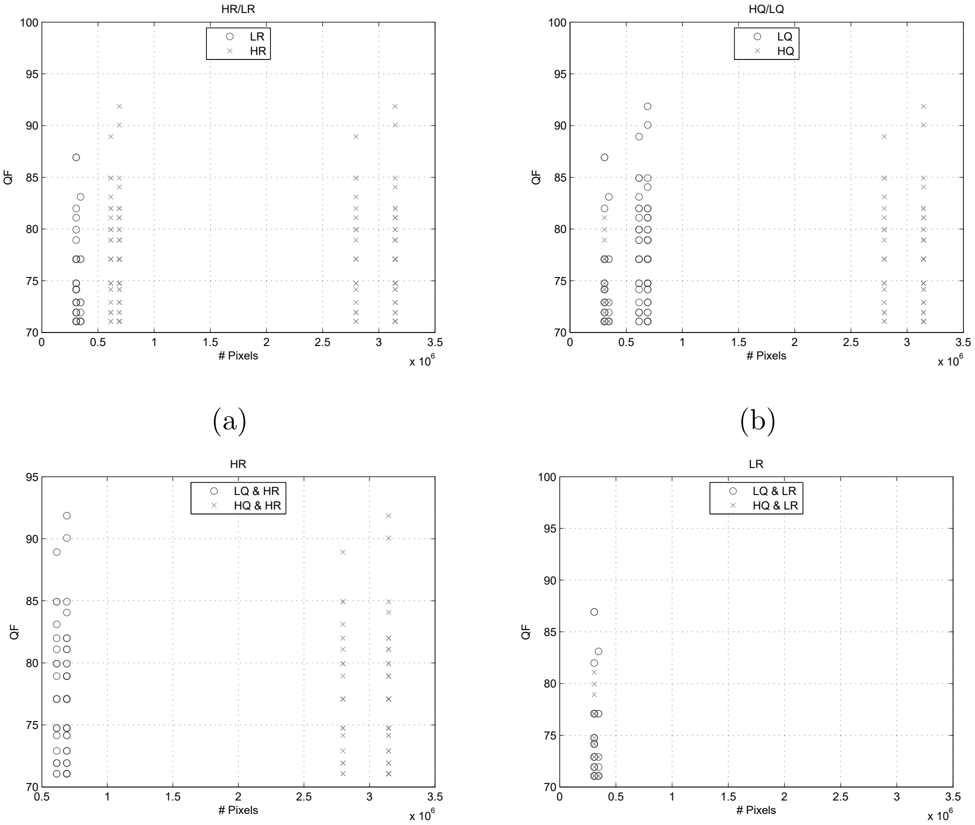
(b)

(c)

(d)

Figure A.6: Number of pixels in the images VS BPP. A.6a: images grouped by input resolution (HR/LR); A.6b: images group by upload quality (HQ/LQ); A.6c: HR input images grouped by upload quality; A.6d: LR input images grouped by upload quality.

[8] http://www.impulseadventure.com/photo/jpeg-snoop.html



(a)

(b)

(c)

(d)

Figure A.7: Number of pixels in the images VS Quality Factor. A.7a: images grouped by input resolution (HR/LR); A.7b: images group by upload quality (HQ/LQ); A.7c: HR input images grouped by upload quality; A.7d: LR input images grouped by upload quality.

A more interesting observation can be deducted from Fig. A.7: trivially, we observe the same six vertical lines corresponding to the different sizes of the images, but all the points are vertically distributed in 17 discrete positions, corresponding to the quality factors reported in Tab. A.2. Thus, we suppose there should be 17 different Quantization Table used in the upload process of the pictures belonging to the proposed dataset. A further discussion about the quantization tables follows in Subsec. A.4.3.

Quality Factor			
1	71.07	**10**	81.99
2	71.93	**11**	83.11
3	72.91	**12**	84.06
4	74.16	**13**	84.93
5	74.75	**14**	86.93
6	77.09	**15**	88.93
7	78.93	**16**	90.06
8	79.94	**17**	91.86
9	81.09		

Table A.2: Quality Factors of the JPEG Compression applied by Facebook (estimated by JPEG Snoop)

A.4.3 Quantization Tables

The images considered in our dataset are all in JPEG format, both the original versions and the downloaded ones. Thus, we want to find out how the JPEG compression affects the pictures, focusing on the Discrete Quantization tables used for that purpose. In fact, the Discrete Quantization Tables (DQT) can, in some way, certify that an image has been processed by some specific tool ([129]). We extracted the tables using JPEGSnoop. In Tab. A.3 and Tab. A.4 we report the DQTs for Luminance and Chrominance relative to the lowest and the highest quality factor.

Moreover, we performed the same operation on some pictures belonging to the authors that were uploaded previously, to check if the tables changed over the years.

DQT Luminance								DQT Chrominance							
9	6	6	9	14	23	30	35	10	10	14	27	57	57	57	57
7	7	8	11	15	34	35	32	10	12	15	38	57	57	57	57
8	8	9	14	23	33	40	32	14	15	32	57	57	57	57	57
8	10	13	17	30	50	46	36	27	38	57	57	57	57	57	57
10	13	21	32	39	63	60	45	57	57	57	57	57	57	57	57
14	20	32	37	47	60	66	53	57	57	57	57	57	57	57	57
28	37	45	50	60	70	70	59	57	57	57	57	57	57	57	57
42	53	55	57	65	58	60	57	57	57	57	57	57	57	57	57

Table A.3: DQT corresponding to QF = 71.07

DQT Luminance								DQT Chrominance							
3	2	2	3	4	6	8	10	3	3	4	8	16	16	16	16
2	2	2	3	4	9	10	9	3	3	4	11	16	16	16	16
2	2	3	4	6	9	11	9	4	4	9	16	16	16	16	16
2	3	4	5	8	14	13	10	8	11	16	16	16	16	16	16
3	4	6	9	11	17	16	12	16	16	16	16	16	16	16	16
4	6	9	10	13	17	18	15	16	16	16	16	16	16	16	16
8	10	12	14	16	19	19	16	16	16	16	16	16	16	16	16
12	15	15	16	18	16	16	16	16	16	16	16	16	16	16	16

Table A.4: DQT corresponding to QF = 91.86

A.4.4 Metadata

Among others, Exif data[141] contain some additional information about the picture, such as camera settings, date, time and generic descriptions. Moreover, a thumbnail of the picture is included. These kind of data has been used for forensic purposes, because it can provide evidences of possible forgeries (e.g. the thumbnail is different from the actual photo). Often, if the camera is equipped with a geo-tagging system, it is possible to find the GPS coordinates of the location where the photo has been captured.

Using JPEGSnoop, we extracted the Exif data from the downloaded images, and we found that Facebook completely removes them. Since no specification is available, our best guess is that, since removing the Exif data reduces the size in byte of the image, this procedure allows to save space on the storing servers, given the huge amount of pictures uploaded in the social network.

Appendix B

Saliency-based feature selection for car detection

B.1 Introduction

Car detection and tracking is a challenging problem Computer Vision, and it is of great interest because the benefits that such a system brings when mounted on surveillance cameras and autonomous vehicles. Among the information used to deal with the tracking problem are the one extracted with the well-known optical flow algorithm. The motion vectors extracted can be used to estimate the global motion of the scene or of the Region Of Interest (ROI) to be tracked with the popular RANSAC [73] algorithm. Several variants of RANSAC already exists, each of which introduces different elements in the filtering step, in the estimation step or in both. Among these methods, MLESAC [142] is a generalization of RANSAC, which provides a more robust estimation by minimizing a negative log likelihood function. We propose to improve MLESAC considering the approach proposed in [143], and in particular using Visual Saliency as a prior in the computation of the likelihood function. Visual Saliency is an important cue related to human perception of what is important in the scene. In the next Sections we will present a study of some saliency computation methods and the tests we have performed to verify the applicability of this approach to the car detection problem.

B.2 Contrast-based saliency computation

B.2.1 Global approach

Cheng et al. [144] propose an approach based on pixelwise differences in the L*a*b* color space. Thus, for every pixel $I_k \in I$, Eq. B.1 is computed.

$$S\left(I_k\right) = \sum_{\forall I_i \in I} D\left(I_i, I_k\right)) \tag{B.1}$$

where D is a metric in the L*a*b* color space.
Since the number of possible pairs can be very large, the authors reason about how to reduce the computational complexity of the method.

Switch to the color space Pixels with the same color have the same saliency, according to Eq. B.1, which becomes

$$S\left(I_k\right) = S\left(c_l\right) = \sum_{j=1}^{n} D\left(c_j, c_l\right)) \tag{B.2}$$

where c_l is the color value of pixel I_k, n is the number of distinct pixel colors, and f_j is the probability of pixel color c_j in image I.

Reduce the number of colors In the RGB color space we have 256^3 possible colors It is clear that such a number it too large to deal with; thus, the authors perform a quantization of the original color space using just 12 colors per channel, but 12^3 is still a big number. Nevertheless, in natural images only a fraction of the color space is present. The authors keep only the colors needed to cover the 95% of the image, and replace the remaining 5% with the closest color in the histogram. This gives an average amount of 85 colors per image. The result is shown in Fig. B.1, taken from [144].

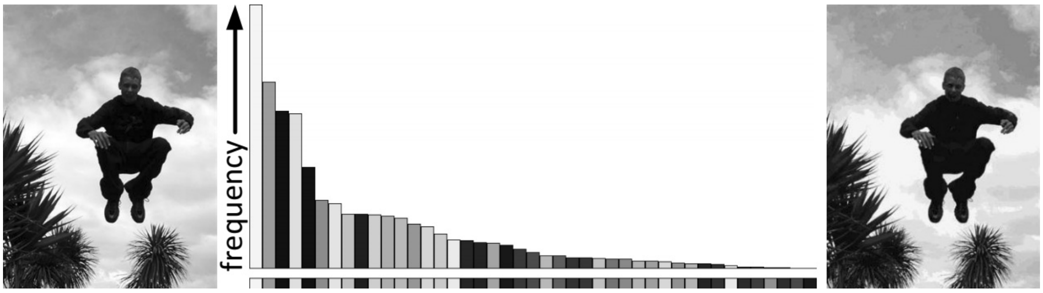


Figure B.1: Given an input image (left), we compute its color histogram (middle). Corresponding histogram bin colors are shown in the lower bar. The quantized image (right) uses only 43 histogram bin colors and still retains sufficient visual quality for saliency detection.

Smoothing Colour quantization can introduce artifacts in the final saliency maps, for example when similar non frequent colors are assigned to different bins in the histogram. To face this problem, the authors apply a linear smoothing function (Eq. B.3) to the saliency values histogram, considering m neighbours of the saliency value. Saliency maps before and after smoothing are shown in Fig. B.2, from [144]

$$S'(c) = \frac{1}{(m-1)\,T} \sum_{i=1}^{m} \left(T - D\left(c, c_i\right)\right) S\left(c_i\right) \tag{B.3}$$

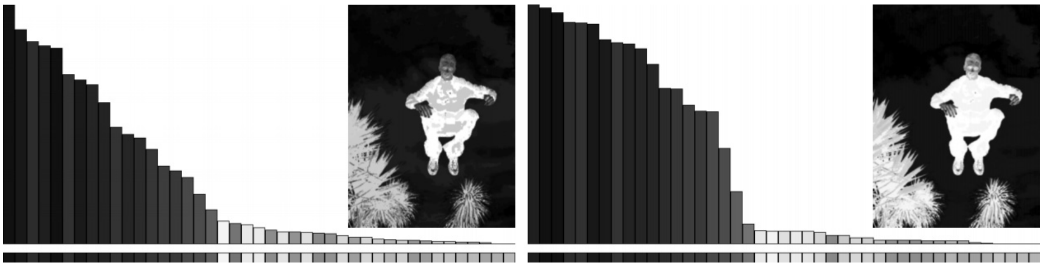


Figure B.2: Saliency of each color, normalized to the range [0, 1], before (left) and after (right) color space smoothing. Corresponding saliency maps are shown in the respective insets.

B.2.2 Region-based approach

The authors also propose to perform segmentation before computing the saliency maps. They use the algorithm by Felzenswalb et al. [145], and then they apply the algorithm described above to regions, instead of pixels. In this way, pixels belonging to the same region will have the same saliency value. Eq. B.4 shows the formula for saliency computation.

$$S\left(r_k\right) = \sum_{r_k \neq r_i} w\left(r_i\right) D_r\left(r_k, r_i\right) \tag{B.4}$$

where $w\left(r_i\right)$ is the size of the region r_i and D (see Eq. B.5)is a color metric.

$$D\left(r_1, r_2\right) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f\left(c_{1,i}\right) f\left(c_{2,i}\right) D\left(c_{1,i}, c_{2,j}\right) \tag{B.5}$$

Spatial information Including spatial relationships can help to emphasize the effect of the differences between close regions and decrease those between far regions. This is done applying Eq. B.6

$$S\left(r_k\right) = w_s\left(r_k\right) \sum_{r_k \neq r_i} e^{\frac{D_s\left(r_k, r_i\right)}{-\sigma^2}} w\left(r_i\right) D_r\left(r_k, r_i\right) \tag{B.6}$$

where D_s is the spatial distance between the regions, σ_s is a spatial weighting term, w_s is a spatial prior related to center bias.

Border refinement To refine the saliency maps, the authors put border regions saliency values to 0. They define a 15-pixels wide border; if a region lies for more than a fixed threshold, its pixels will be put to 0. After this, the saliency is recomputed.

B.3 Graph based visual saliency

This approach, proposed by Harel et al. [146], aims to estimate human fixation points, while [144] has the goal to detect salient objects. This approach is based on biological considerations. The authors define three standard steps for each saliency map computation:

1. Feature maps computation;
2. Activation maps creation;
3. Normalization and combination.

In [146], the focus is on step 2) and 3).

They start from feature maps $M : [n]^2 \rightarrow \mathbb{R}$ and try to find points somehow "unusual" that attract beholders' attention. The first choice would be to define "unusual" as unlikely, but this has been already done in the state of the art. Thus, they define a **dissimilarity measure** between points of a given map. So, given two points $M(i,j)$ and $M(p,q)$, let the dissimilarity d be

$$d\left((i,j) \parallel (p,q)\right) \triangleq \left\| \log \frac{M(i,j)}{M(p,q)} \right\| \tag{B.7}$$

Eq. B.7 induces a graph over the map, as shown in Fig. B.3. Let's call this graph G_A. The weights of the edges are proportional to the dissimilarity between the connected nodes and their closeness in the the space of M,

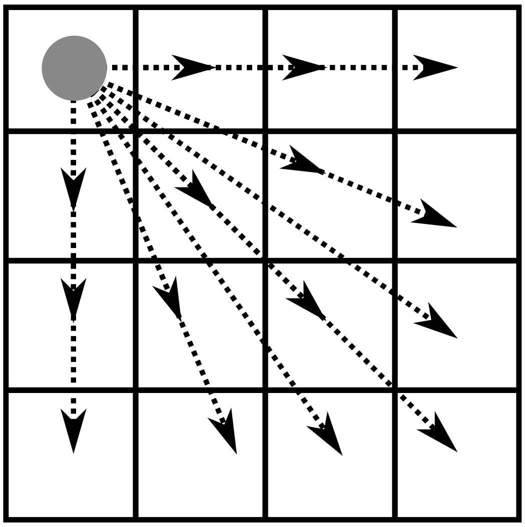


Figure B.3: Induced graph over a feature map M.

according to Eq. B.9.

$$w_1\left(\left(i,j\right),\left(p,q\right)\right) \triangleq d\left(\left(i,j\right) \parallel \left(p,q\right)\right) \cdot F\left(i-p, j-q\right), \text{ where} \quad \text{(B.8)}$$

$$F\left(a,b\right) \triangleq \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right) \quad \text{(B.9)}$$

Such a graph can be directly translated into a Markov Chain by:

1. Normalizing the weights of outbound edges to 1 for each node;

2. Mapping nodes to states;

3. Mapping edges to transitions.

The resulting Markov Chain is:

irreducible - each state can be reached from each other;

aperiodic - starting from a state i, it is possible to come back at i within a any number of steps;

recurrent - the probability of coming back at any state is always greater than 0.

Thus, the Markov Chain is **ergodic** and it has an equilibrium distribution π, defined as follows. Let S be the set of all the states and be $P = \{p_{ij} | s_i, s_j \in S\}$ be the transition matrix. The distribution π has the following properties

1. $0 \leq \pi_j \leq 1, \qquad \forall j \in \{1, \ldots, \#S\}$;
2. $\sum_{j \in S} \pi_j = 1$;
3. $\pi_j = \sum_{i \in S} \pi_i p_{ij}$;
4. $\pi_j = \frac{C}{M_j}$, where C is a normalization constant and M_j is the expected coming-back time;
5. $\lim\limits_{n \to \infty} p_{ij}^{(n)} = \frac{C}{M_j}$.

The computation ends giving out the amount of time a random walker would spend in the node. Such a value is taken as an activation pixel-wise measure, that forms an **activation map**.

The whole process is then iterated over the activation map, giving the saliency as a result.

B.4 Frequency Tuned

Achanta et al. [147] analyse the relationship between saliency and frequencies, considering different approaches in the state-of-the-art. Then, they define some requirements for a saliency detector:

- Emphasize the largest salient objects.
- Uniformly highlight whole salient regions.
- Establish well-defined boundaries of salient objects.

- Disregard high frequencies arising from texture, noise and blocking artefacts
- Efficiently output full resolution saliency maps.

They claim that, since a saliency map should contain a wide range of frequencies, using a Difference of Gaussian filter can be a good choice to build such a map, by setting in the proper manner the involved frequencies (i.e. the low-cut frequency σ_1 and the high-cut frequency σ_2, $\sigma_1 \geq \sigma_2$). The implementation of this filtering process is somehow unusual, since they compute saliency using B.10

$$S\left(x,y\right) = \left\|\mathbf{I}_{\mu} - \mathbf{I}_{\omega_{hc}}\left(x,y\right)\right\| \tag{B.10}$$

where $\mathbf{I}_{\mu}$ is the arithmetic mean of the values of all the pixels in the image I, and $\mathbf{I}_{\omega_{hc}}$ is the gaussian blurred version of I. The image is represented in the L*a*b* color space.

B.5 Spectral Residual

Hou et al. [148] investigated the visual saliency in natural images for object detection purposes. They observed that the Log Spectrum curve of the Fourier Transform of natural images tend to have the same trend, no matter the subject. They define the visual saliency as the singularities in the spectrum. The approach can be summarized as follows. Given an image I:

- Compute the Fourier Transform $\mathfrak{F}\left(I\right)$;
  $\mathcal{A} = \Re\left(\mathfrak{F}\left(I\right)\right)$;
  $\mathcal{P} = \Im\left(\mathfrak{F}\left(I\right)\right)$;
- Compute the logarithm of the real part of the transform: $\mathcal{L} = \log\left(\mathcal{A}\right)$;

- Compute the residual $\mathcal{R} = \mathcal{L} - h_n * \mathcal{L}$, where h_n is a box filter of size $n \times n$;
- Compute the saliency map $\mathcal{S} = g(x) * \mathfrak{F}^{-1} \left[\exp\left(\mathcal{R} + \mathcal{P}\right)\right]^2$.

B.6 Experiments and evaluation

We performed four different evaluations to assess the impact of the saliency methods described above in a car detection scenario. We used the Toyota Motor Europe (TME) Motorway Dataset [149], which

> *is composed by 28 clips for a total of approximately 27 minutes (30000+ frames) with vehicle annotation. Annotation was semi-automatically generated using laser-scanner data. Image sequences were selected from acquisition made in North Italian motorways in December 2011. This selection includes variable traffic situations, number of lanes, road curvature, and lighting, covering most of the conditions present in the complete acquisition.*
>
> The test have been run on the images from the right camera, as suggested by the authors of the dataset.

B.6.1 Ground-truth images

The dataset provides bounding boxes annotations for each frame. Given an image I_i, and its bounding boxes annotations, we define the union of all the boxes as

$$\mathcal{B}^{(i)} = \bigcup_{k=1}^{n_i} b_k^{(i)}$$

where $b_k = \left(x_1^k, x_2^k, y_1^k, y_2^k\right)$. Then we generate a ground truth image G such that

$$G\left(x,y\right) = \begin{cases} true & \text{if } \left(x,y\right) \in \mathcal{B}^{(i)} \\ false & \text{otherwise} \end{cases}$$

We will use these images for our experiments.

B.6.2 Test 1: Saliency reliability

We investigated the reliability of the different saliency methods with respect to the car detection task. For each image, we compute the ratio between the sum of the saliency values that lie inside the bounding boxes and the bounding boxes theirselves, posing $false = 0$ and $true = 255$. Formally, given a saliency map S computed over an image I, the reliability is

$$\mathcal{R} = \frac{\sum_{(x,y)\in\mathcal{B}^{(i)}} S\left(x,y\right)}{\sum_{\forall(x,y)} G\left(x,y\right)} \tag{B.11}$$

where $S\left(x,y\right) \in \left(0, 255\right)$, and therefore $0 \leq \mathcal{R} \leq 1$. The rationale behind this measurement is to catch how much of the ground truth is correctly highlighted by the saliency methods. Fig. B.4 show the distribution of the reliability values, highlighting the maximum value for each saliency method.

B.6.3 Test 2: Image Coverage

This test is meant to evaluate the percentage of the image covered by binarized saliency maps varying the threshold τ from 0 to 255.

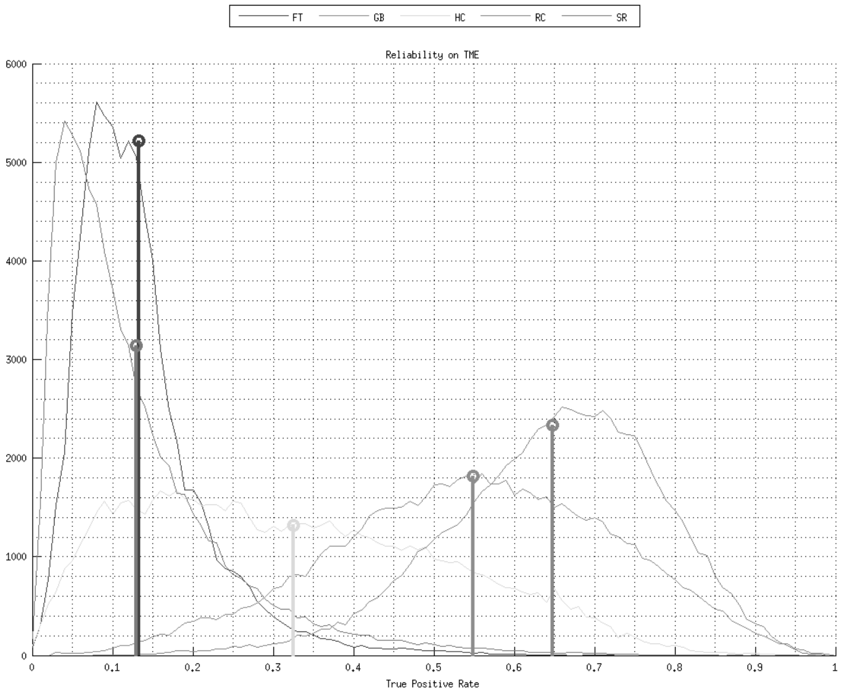


Figure B.4: Reliability on the TME dataset

So, given an image I and the corresponding saliency map S, we first performed a binarization obtaining a map

$$B_\tau(x,y) = \begin{cases} 1 & \text{if } I(x,y) > \tau \\ 0 & \text{if } I(x,y) \leq \tau \end{cases} \tag{B.12}$$

the coverage C_τ is computed as

$$C_\tau = \frac{1}{N} \sum_{(x,y)} B_\tau(x,y) \tag{B.13}$$

where N is the number of pixels in B.

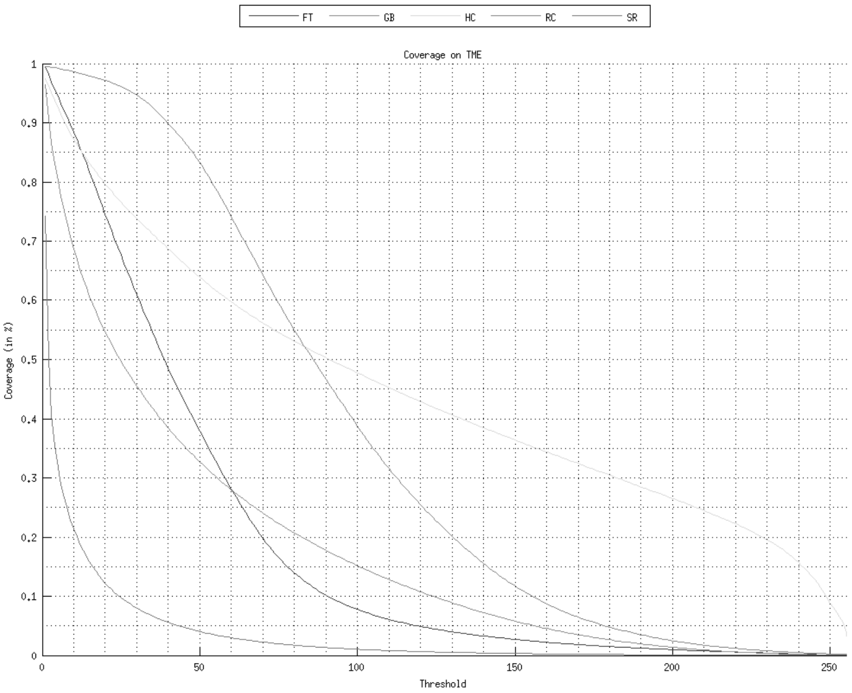


Figure B.5: Coverage on the TME dataset

B.6.4 Test 3: True Positive Rate VS Coverage

True Positive Rate (TPR) and False Positive Rate (FPR) are two measures of the goodness of a binary classification algorithm. Given a set of predictions P and the corresponding real labels L, we define the *confusion matrix* as shown in Tab. B.1:

		Prediction outcome		
		p	**n**	**total**
	$\mathbf{p}'$	True Positive	False Negative	P'
Actual Value	$\mathbf{n}'$	False Positive	True Negative	N'
	total	P	N	

Table B.1: Confusion Matrix for a binary classifier.

TPR, also known as **sensitivity** or **recall**, is the ratio between the true positives and the positives in the ground truth, i.e.

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{B.14}$$

In this test, we evaluate the trend of TPR with respect to the image coverage (see Subsec. B.6.3) varying the binarization threshold in the range $[0, 255]$. Therefore, for each threshold τ we obtain a confusion matrix, and more precisely we calculate the TPR and the Coverage. It is important to note that the Coverage can be calculated as shown in Eq. B.15.

$$\text{Coverage} \ = \frac{TP + FP}{TP + FN + FP + TN} \tag{B.15}$$

This measurement is useful in order to find out how much of the true positives (i.e. the interesting pixels that have been correctly detected) still remain in the pixels when the non-salient parts are discarded.

The results, shown in Fig. B.6 and Tab. B.2, demonstrate that a high TPR (over 99 %) can be reached using only the % of the image with the GBVS saliency method [146].

Saliency Method	**TPR**	**Coverage**
FT	0.9917	0.9953
GBVS	0.9914	**0.6363**
HC*	0.9720	0.9755
RC	0.9901	0.9109
SR*	0.9631	0.7406

Table B.2: Lowest Coverage when TPRs > 0.99

* This method never reaches TPR > 0.99

B.6.5 Test 4: ROC Curve

In the last test, which confirmed the previous one, we compute the Receiver Operator Characteristic (ROC) Curve for the aforementioned saliency methods. The ROC Curve show the trend of TPR (see Eq. B.14) with respect to FPR as the binarization threshold changes. The chart in Fig. B.7 can be summarized calculating the Area Under the Curve (AUC) for each line in the chart. The bigger the area, the more accurate is the classification. AUCs are reported in Fig. B.8.

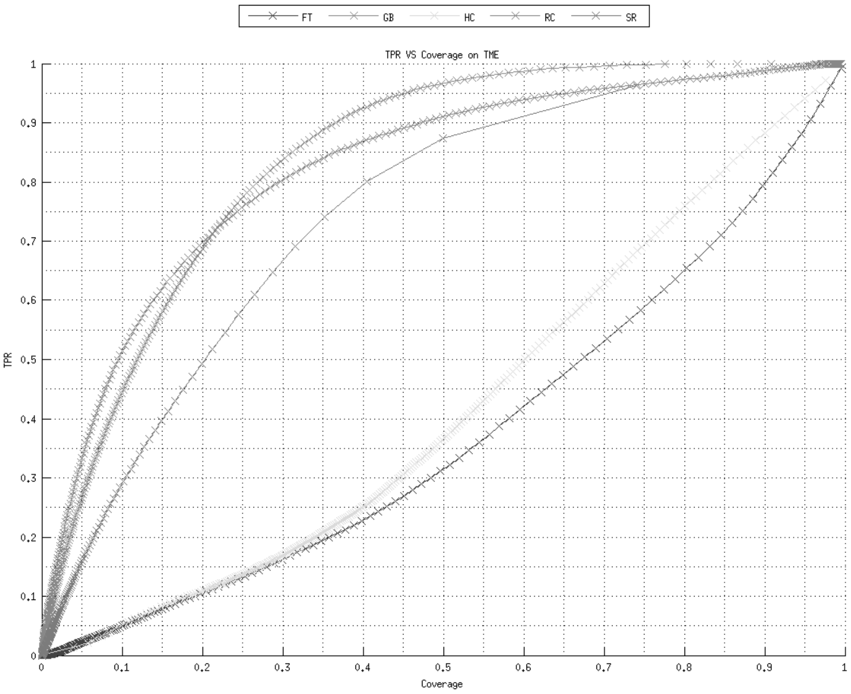


Figure B.6: TPR VS Coverage on the TME dataset

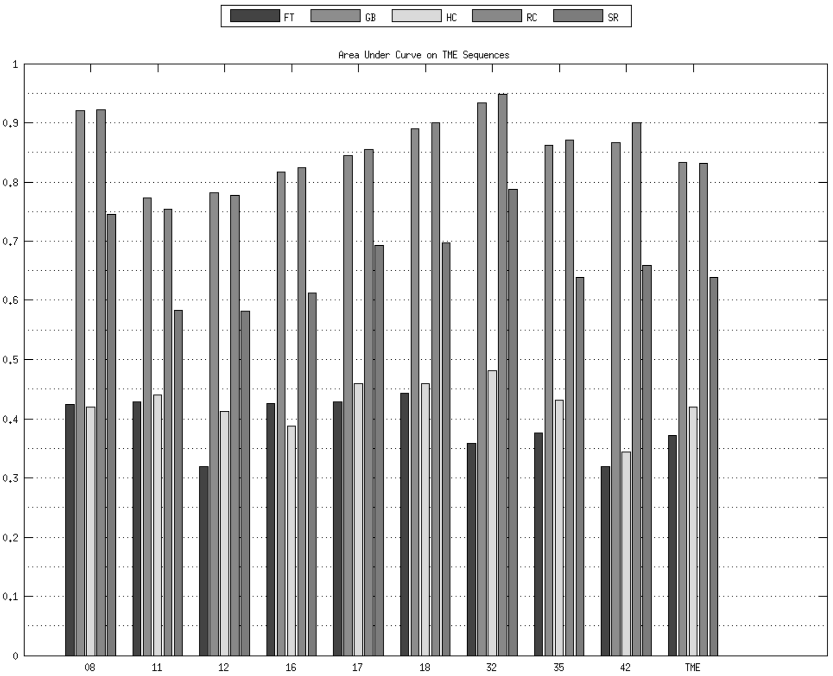


Figure B.8: Area Under the Curve

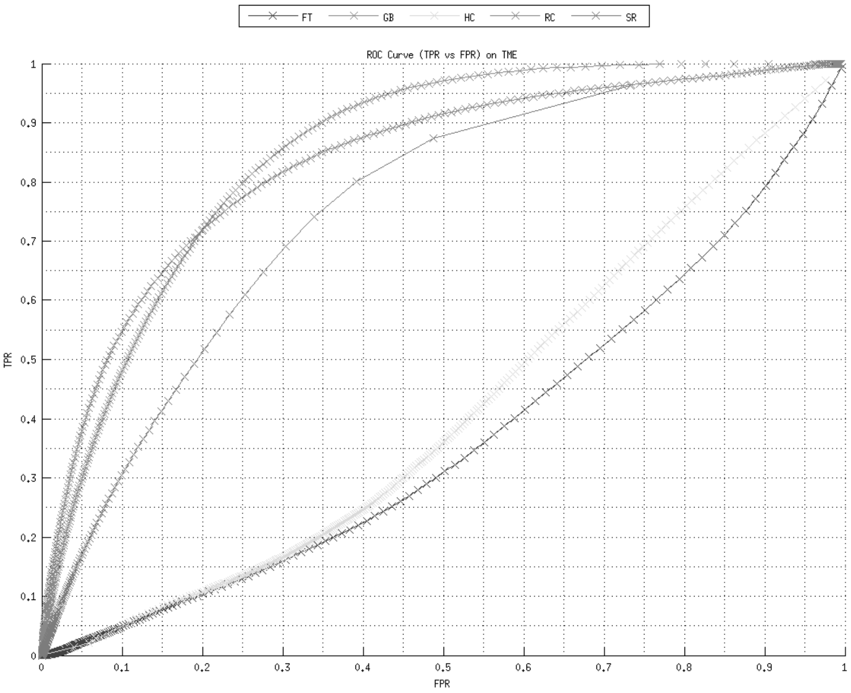
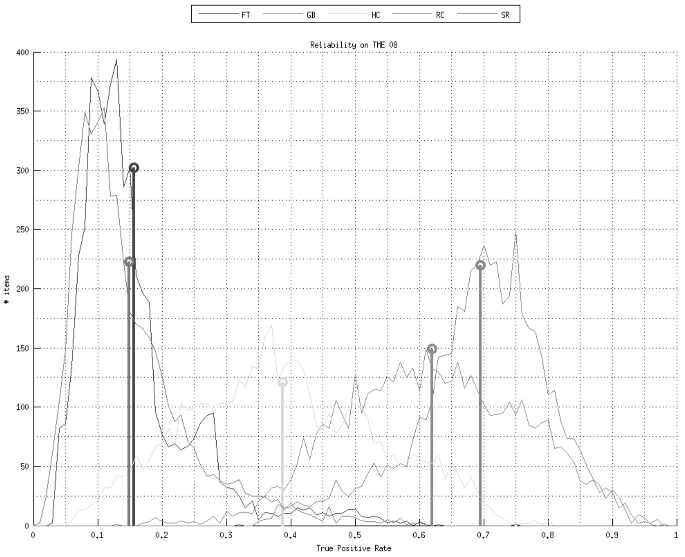
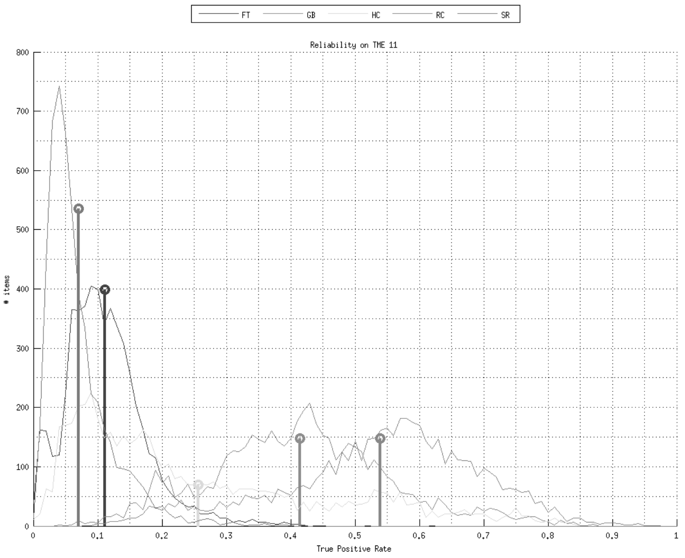


Figure B.7: ROC Curve on the TME dataset

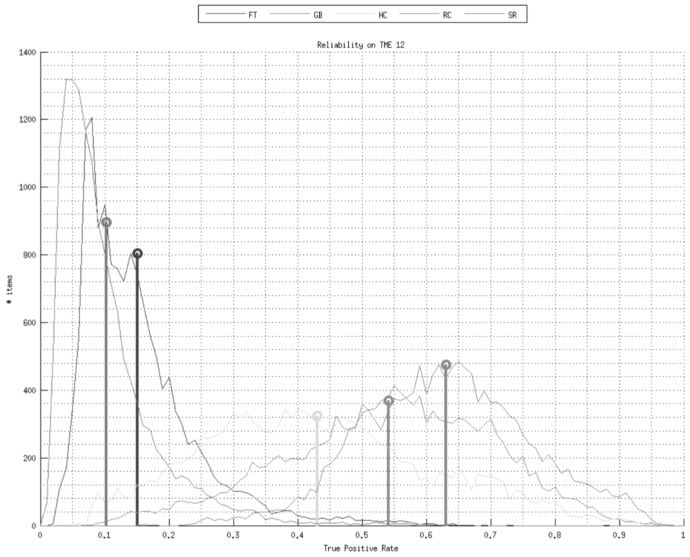
B.7 Test 1: Reliability on the sequences of the TME Dataset



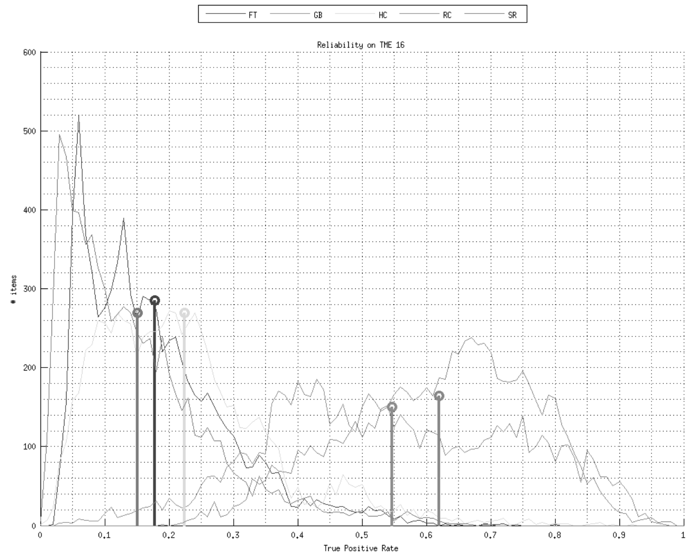
(a) TME 08



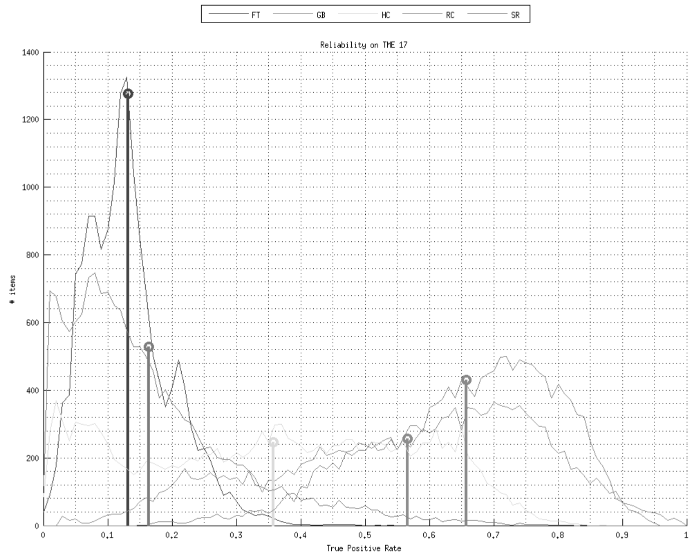
(b) TME 11



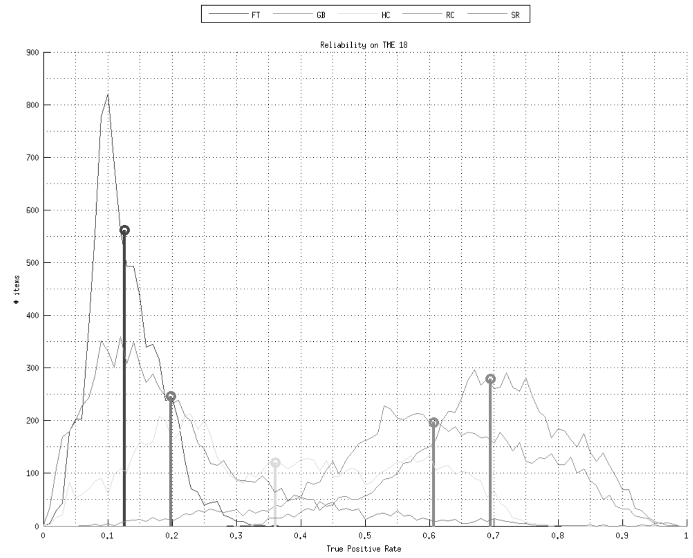
(c) TME 12



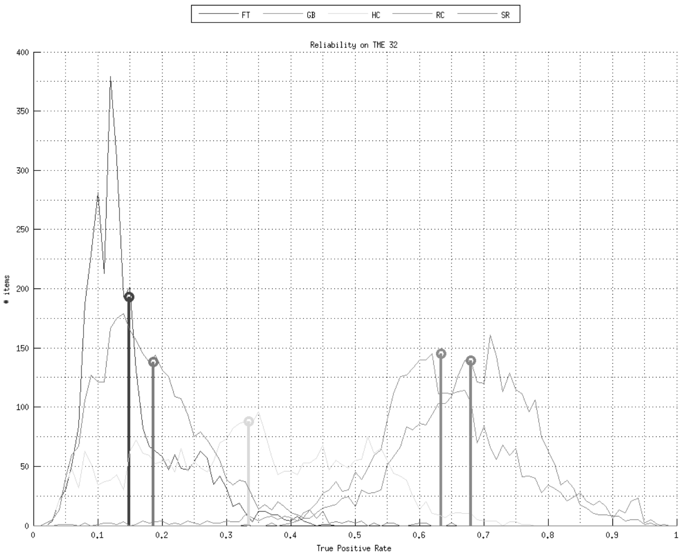
(d) TME 16



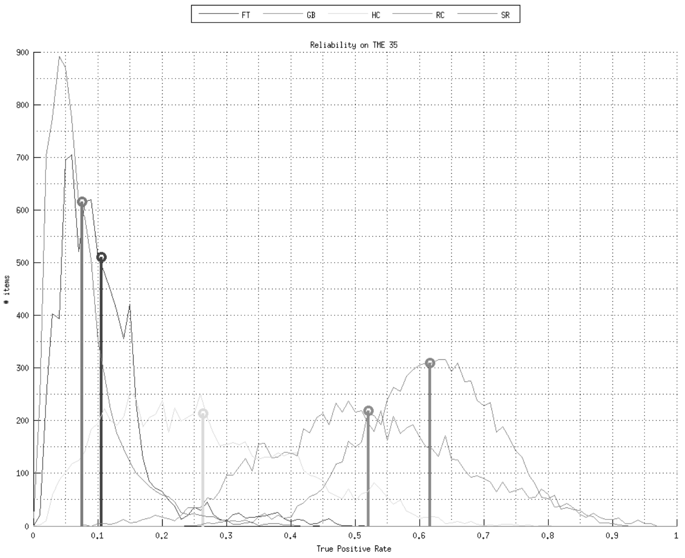
(e) TME 17



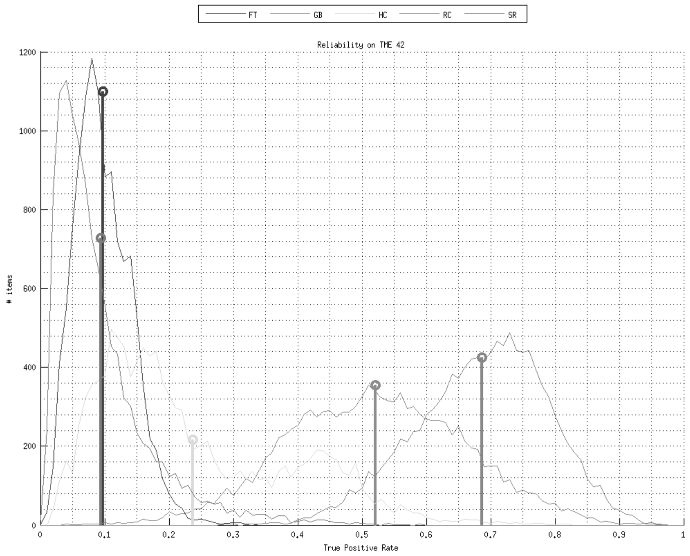
(f) TME 18



(g) TME 32

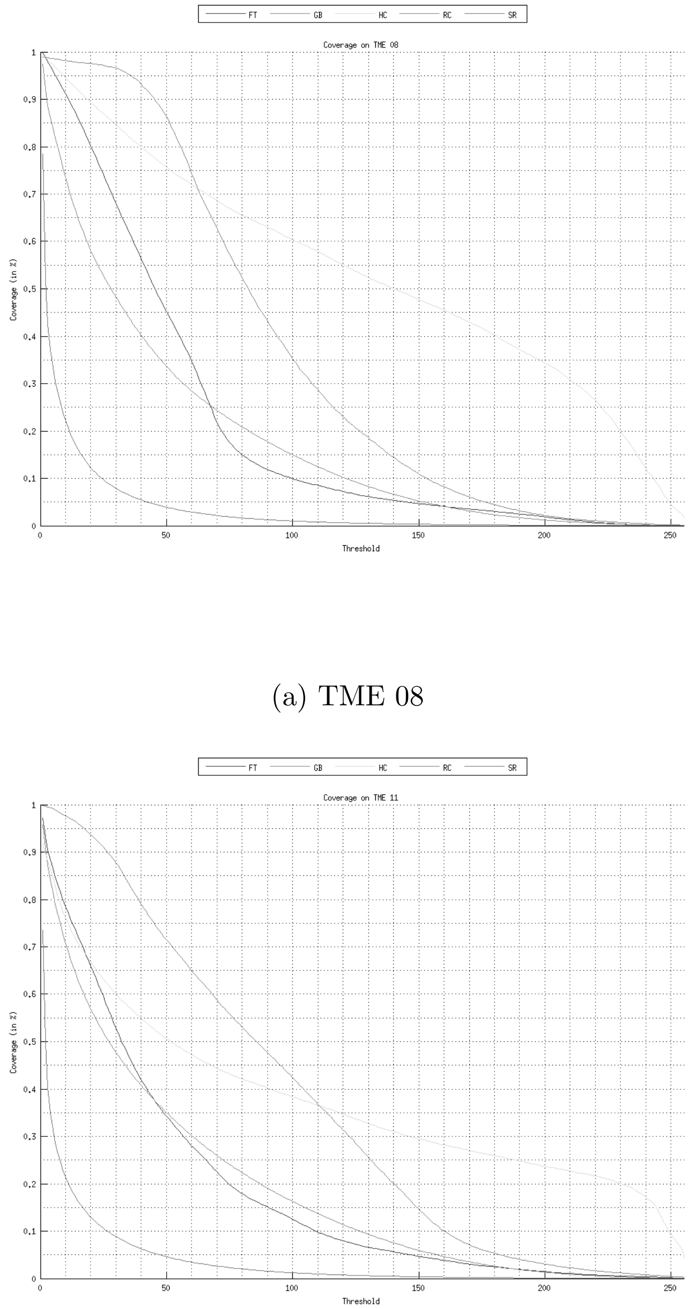


(h) TME 35


FT
GB
HC
RC
SR
Reliability on TME 42
1200
1000
800
600
400
200
0
# items
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True Positive Rate

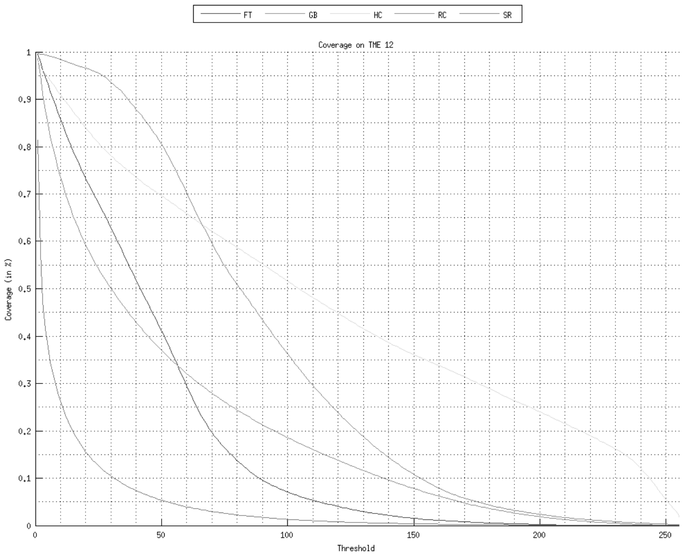
(i) TME 42

B.8 Test 2: Image Coverage on the sequences of the TME dataset

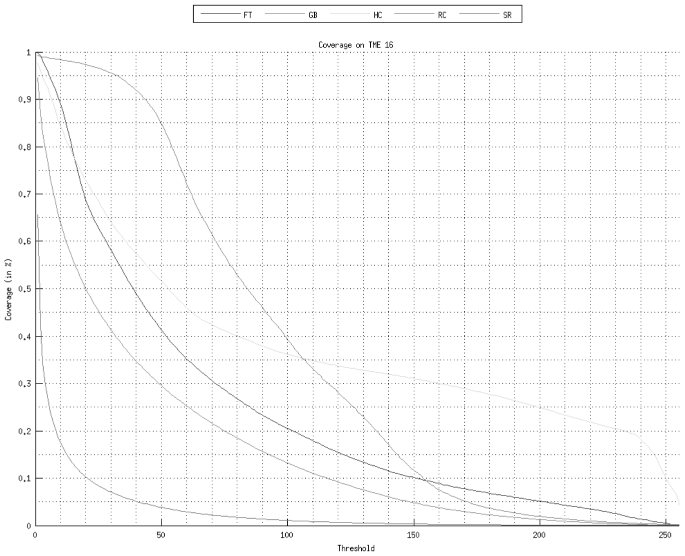


(a) TME 08

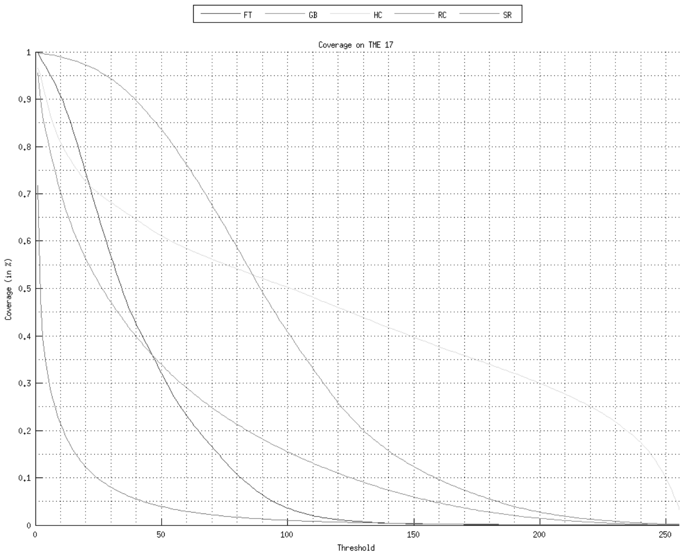
(b) TME 11



(c) TME 12

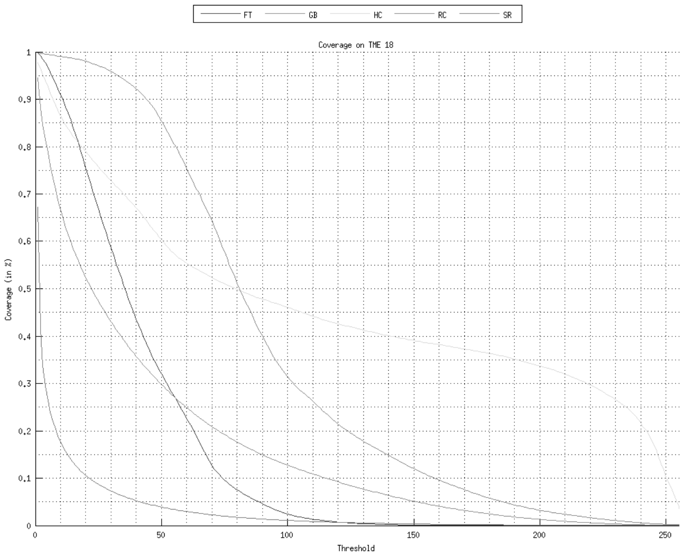


(d) TME 16

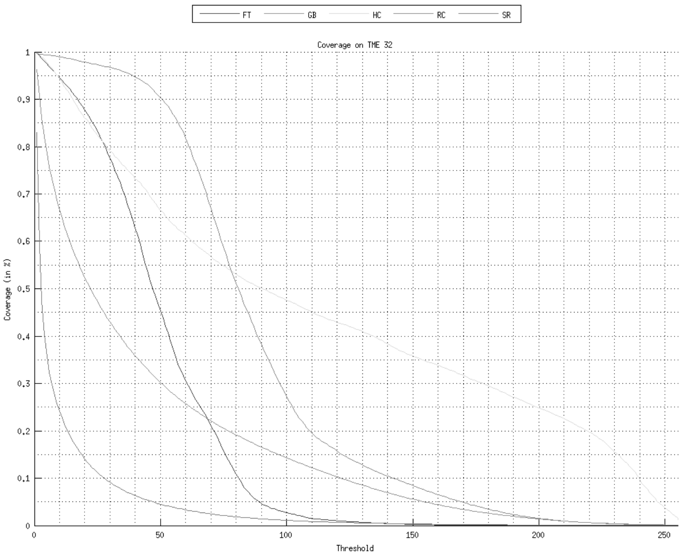


(e) TME 17

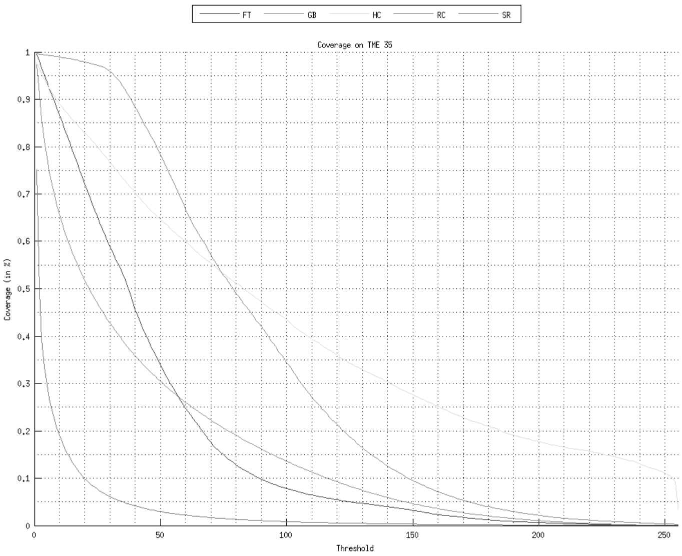
/



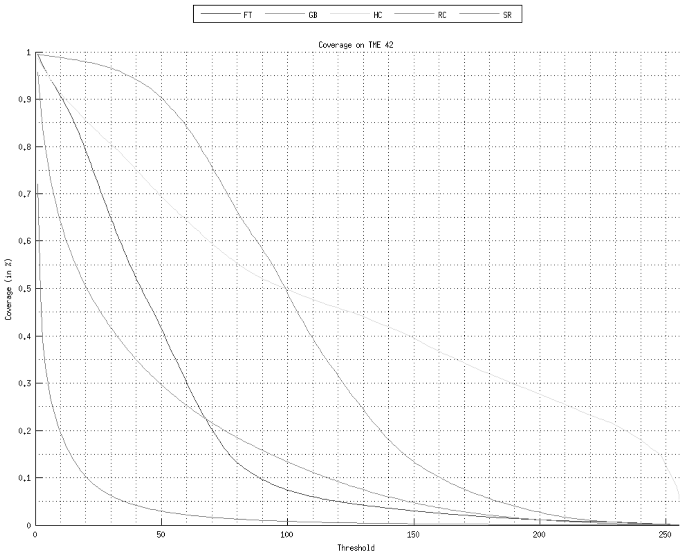
(f) TME 18



(g) TME 32

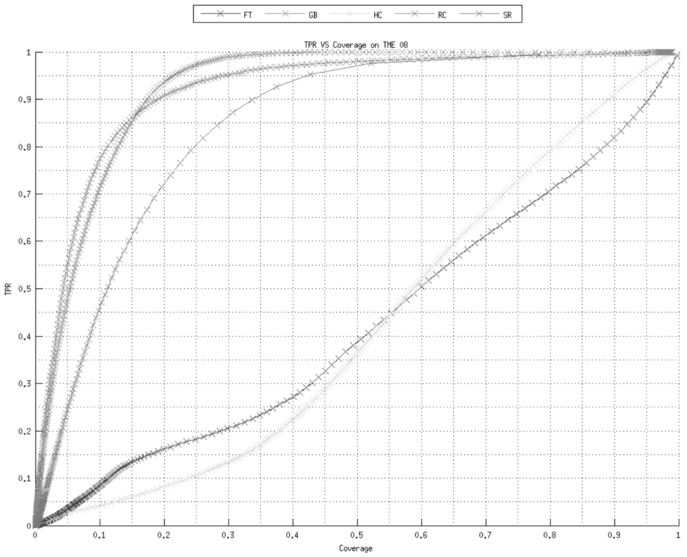


(h) TME 35

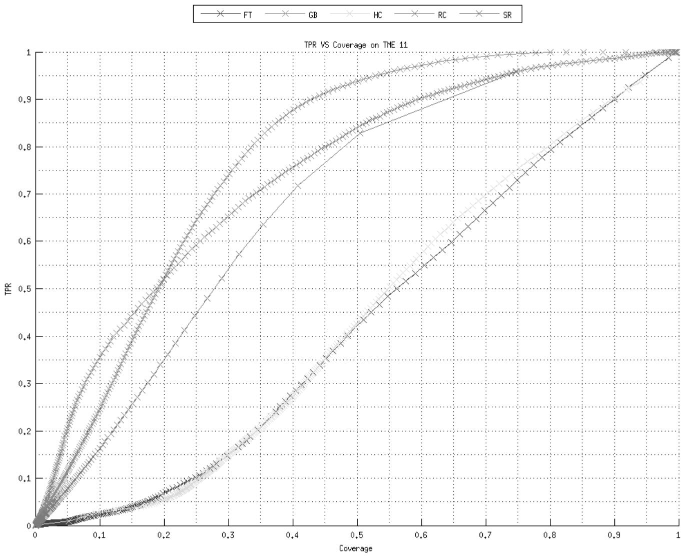

FT
GB
HC
RC
SR
Coverage on TME 42
Coverage (in %)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
Threshold

(i) TME 42

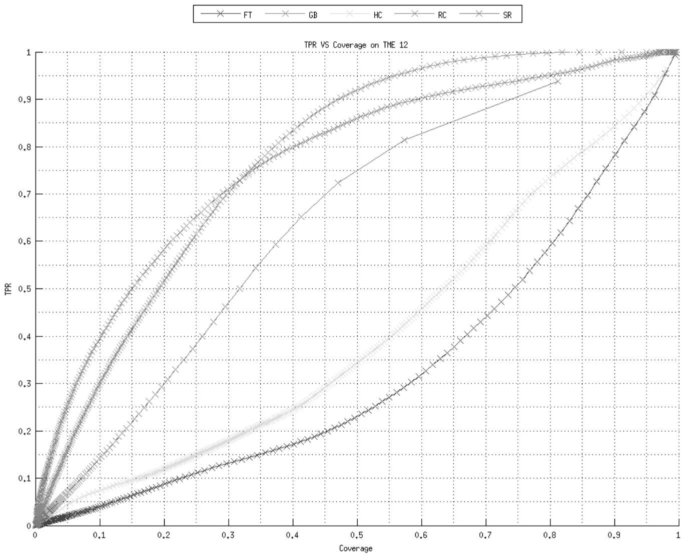
B.9 Test 3: TPR vs Image Coverage on TME sequences



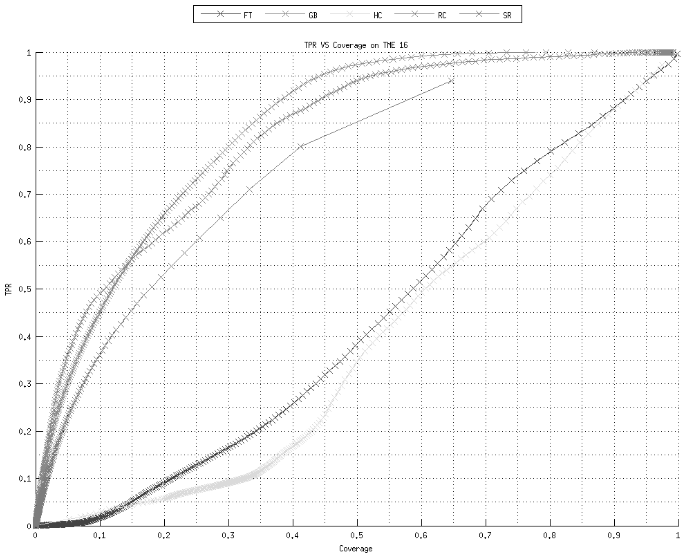
(a) TME 08



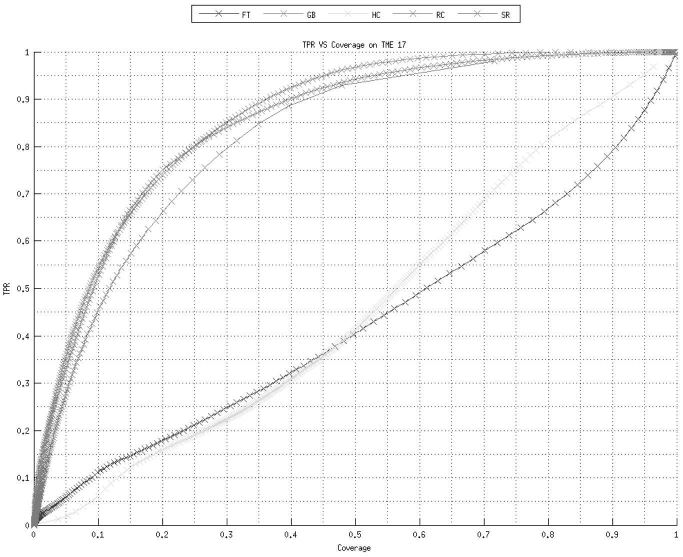
(b) TME 11



(c) TME 12

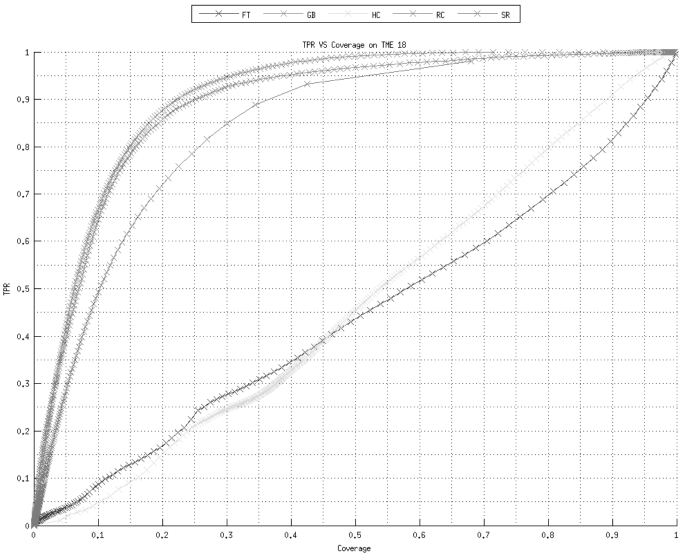


(d) TME 16

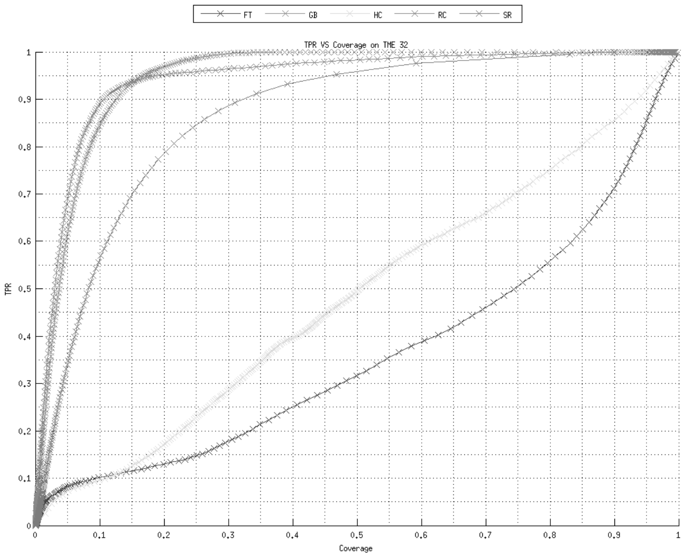


(e) TME 17

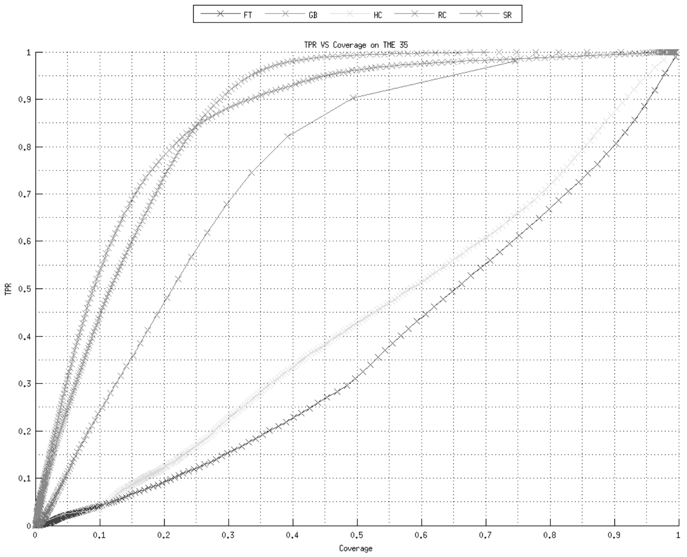
/



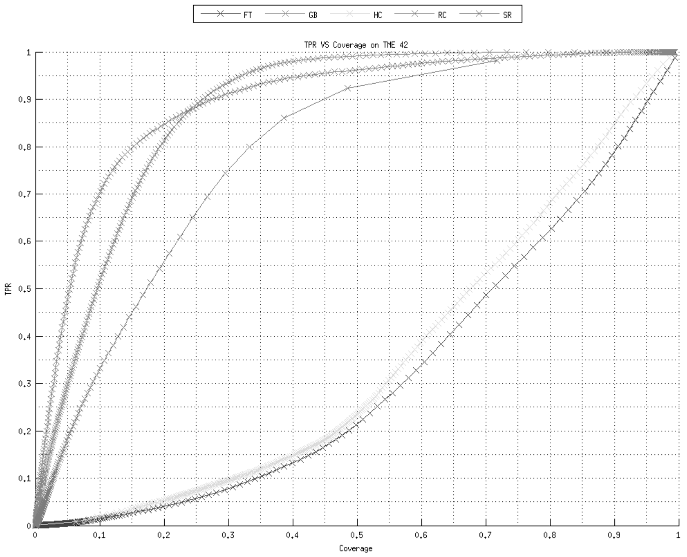
(f) TME 18



(g) TME 32

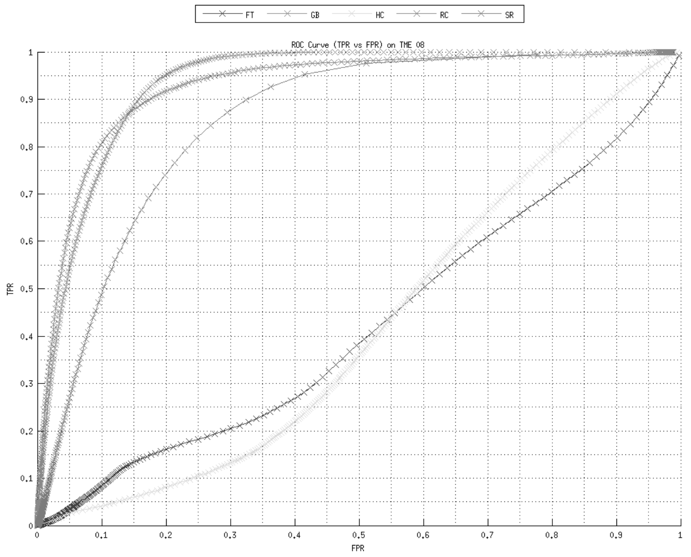


(h) TME 35

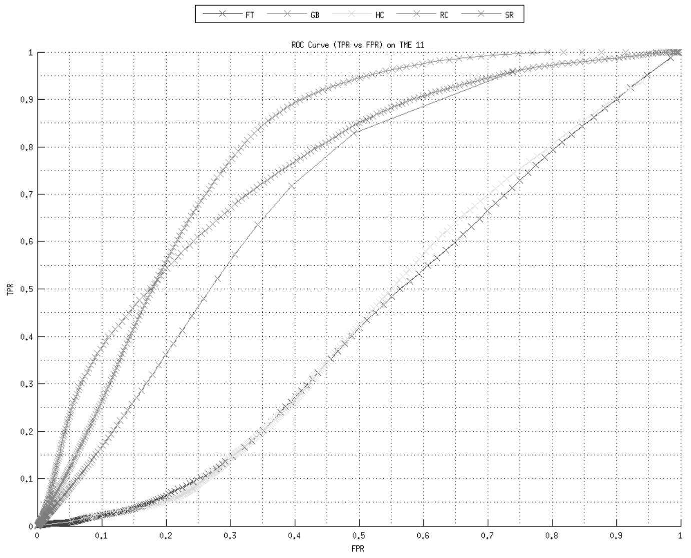

FT
GB
HC
RC
SR
TPR VS Coverage on TME 42
TPR
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Coverage

(i) TME 42

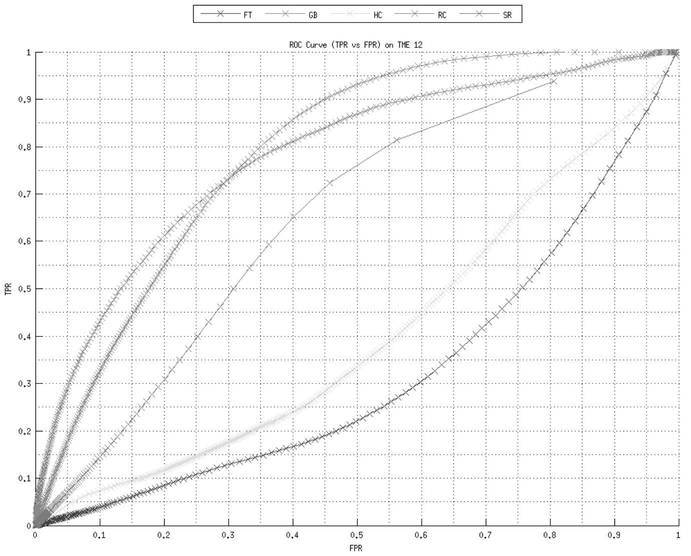
B.10 Test 4: ROC Curve on TME sequences



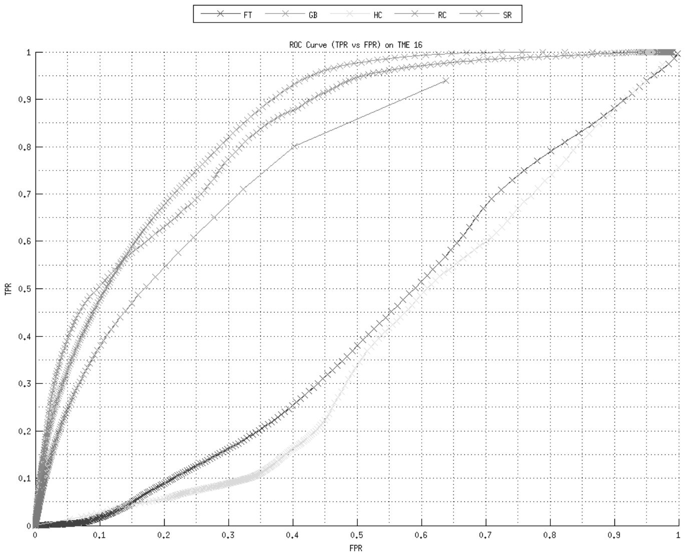
(a) TME 08



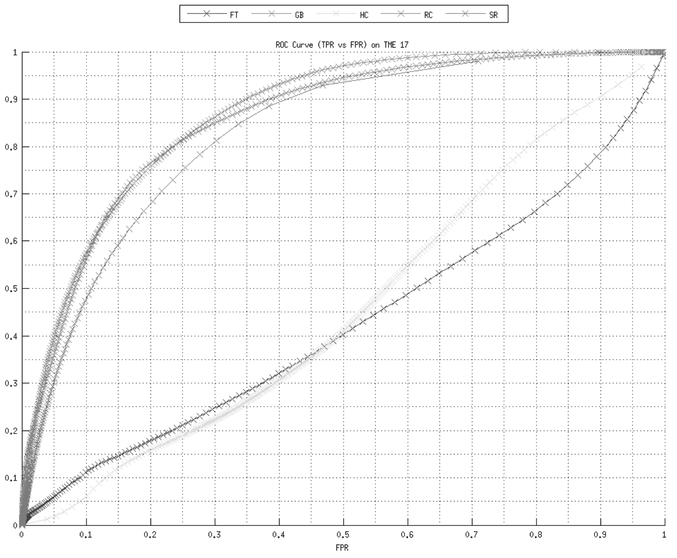
(b) TME 11



(c) TME 12

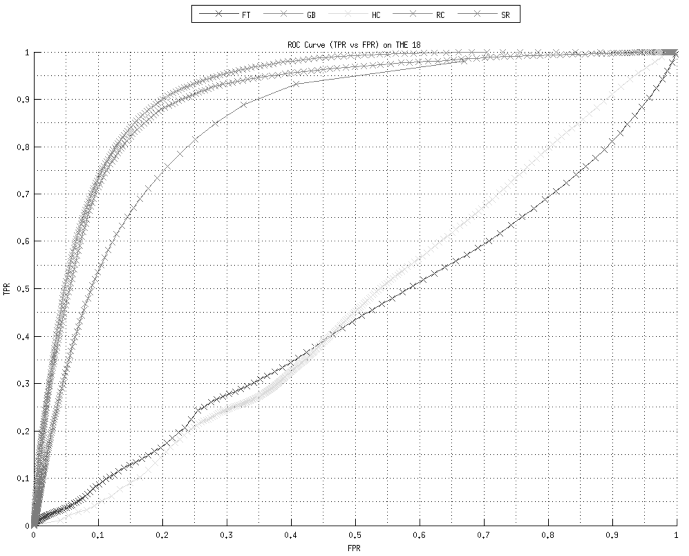


(d) TME 16

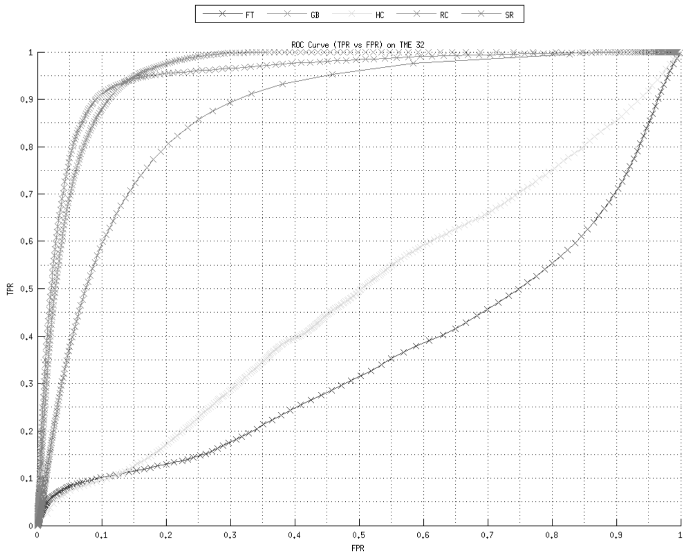


(e) TME 17

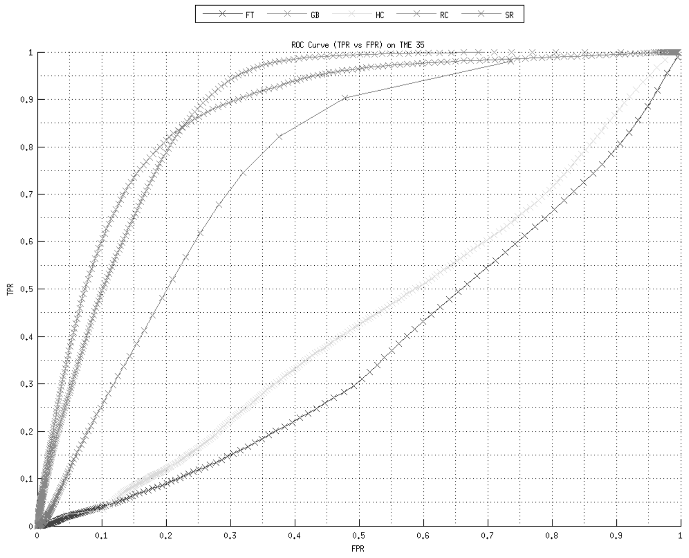
/



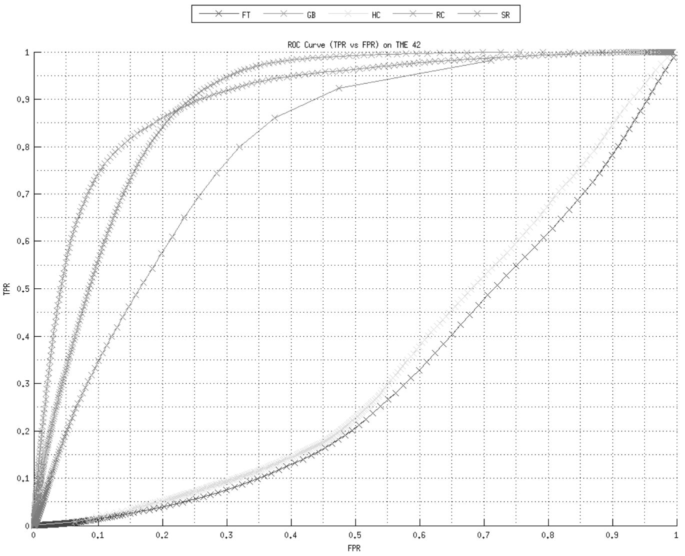
(f) TME 18



(g) TME 32



(h) TME 35


FT
GB
HC
RC
SR
ROC Curve (TPR vs FPR) on TME 42
TPR
FPR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(i) TME 42

Bibliography

[1] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì, "Aligning codebooks for near duplicate image detection," *Multimedia Tools and Applications*, 2013.

[2] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[3] A. Vedaldi, "Invariant representations and learning for computer vision," Ph.D. dissertation, University of California at Los Angeles, 2008.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] B. Julesz, "Textons, the elements of texture perception, and their interactions." *Nature*, vol. 290, pp. 91–97, 1981.

[6] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America A-Optics Image Science and Vision*, vol. 7, no. 5, pp. 923–932, 1990.

[7] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. of the ECCV Interna-*

tional Workshop on Statistical Learning in Computer Vision, 2004, pp. 59–74.

[8] Y. Hu, X. Cheng, L.-T. Chia, X. Xie, D. Rajan, and A.-H. Tan, "Coherent Phrase Model for Efficient Image Near-Duplicate Retrieval," *IEEE Transactions on Multimedia*, vol. 11, no. 8, pp. 1434–1445, 2009.

[9] J. Zhang, M. Marszaek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[11] ——, "A sparse texture representation using local affine regions," pp. 1265–1278, 2005.

[12] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. of the ACM International Conference on Image and Video Retrieval*, 2007.

[13] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[14] V. Estivill-Castro, "Why so many clustering algorithms: A position paper," *ACM SIGKDD Exploration Newsletter*, vol. 4, no. 1, pp. 65–75, 2002.

[15] J. Kleinberg, "An impossibility theorem for clustering," *Proc. of Advances in Neural Information Processing Systems*, pp. 446–453, 2002.

[16] R. Ghaemi, N. Sulaiman, H. Ibrahim, and N. Mustapha, "A Survey : Clustering Ensembles Techniques," *Engineering and Technology*, vol. 38, no. February, pp. 636–645, 2009.

[17] A. Topchy, A. Jain, and W. Punch, "Combining multiple weak clusterings," in *Third IEEE International Conference on Data Mining*, 2003, pp. 0–7.

[18] ——, "A mixture model for clustering ensembles," *Proc. of SIAM International Conference on Data Mining*, pp. 379–390, 2004.

[19] A. Strehl and J. Ghosh, "Cluster Ensembles A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.

[20] A. Fred, "Finding consistent clusters in data partitions," *Multiple classifier systems*, pp. 309–318, 2001.

[21] A. L. N. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Proc. of International Conference on Pattern Recognition*, vol. 4, 2002, pp. 276–280.

[22] H. Luo, F. Jing, and X. Xie, "Combining multiple clusterings using information theory based genetic algorithm," in *Proc. of IEEE International Conference on Computational Intelligence and Security*, vol. 1. IEEE, 2006, pp. 84–89.

[23] P. Kellam, X. Liu, N. Martin, C. Orengo, S. Swift, and A. Tucker, "Comparing, contrasting and combining clusters in viral gene expression data," in *Proceedings of 6th Workshop on Intelligent Data Analysis in Medicine and Pharmocology*, 2001, pp. 56–62.

[24] S. Dudoit and J. Fridlyand, “Bagging to improve the accuracy of a clustering procedure,” *Bioinformatics*, vol. 19, no. 9, pp. 1090–1099, 2003.

[25] B. Fischer and J. Buhmann, “Path-based clustering for grouping of smooth curves and texture segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 1–6, 2003.

[26] J. Azimi, M. Mohammadi, A. Movaghar, and M. Analoui, “Clustering ensembles using genetic algorithm,” in *Proc. of the International Conference on Application-Specific Systems, Architectures and Processors*, 2007, pp. 119–123.

[27] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2006, pp. 2161–2168.

[28] M. J. Pazzani, “Constructive Induction of Cartesian Product Attributes,” in *Science*, 1995.

[29] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, “Fast Keypoint Recognition Using Random Ferns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[30] F. Kong and J. Tan, “Dietcam: Automatic dietary assessment with mobile camera phones.” *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 147–163, 2012.

[31] S. Kim, T. Schap, M. Bosch, R. Maciejewski, E. J. Delp, D. S. Ebert, and C. J. Boushey, “Development of a mobile user interface for image-based dietary assessment,” in *International Conference on Mobile and Ubiquitous Multimedia*, 2010, pp. 13:1–13:7.

[32] L. Arab, D. Estrin, D. H. Kim, J. Burke, and J. Goldman, "Feasibility testing of an automated image-capture method to aid dietary recall," *European Journal of Clinical Nutrition*, vol. 65, no. 10, pp. 1156–1162, 2011.

[33] C. Xu, Y. He, N. Khannan, A. Parra, C. Boushey, and E. Delp, "Image-based food volume estimation," in *International Workshop on Multimedia for Cooking and Eating Activities*, 2013, pp. 75–80.

[34] F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, and E. J. Delp, "The use of mobile devices in aiding dietary assessment and evaluation." *Journal of Selected Topics Signal Processing*, vol. 4, no. 4, pp. 756–766, 2010.

[35] G. O'Loughlin, S. J. Cullen, A. McGoldrick, S. O'Connor, R. Blain, S. O'Malley, and G. D. Warrington, "Using a wearable camera to increase the accuracy of dietary analysis." *American Journal of Preventive Medicine*, vol. 44, no. 3, pp. 297–301, 2013.

[36] J. M. Fontana and E. Sazonov, "Chapter 7.4 - detection and characterization of food intake by wearable sensors," in *Wearable Sensors*, E. Sazonov and M. R. Neuman, Eds. Oxford: Academic Press, 2014, pp. 591 – 616.

[37] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, and G.-Z. Yang, "An Intelligent Food-Intake Monitoring System Using Wearable Sensors," *International Conference on Wearable and Implantable Body Sensor Networks*, pp. 154–160, 2012.

[38] M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, "A Food Recognition System for Diabetic Patients

Based on an Optimized Bag-of-Features Model," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1261–1271, 2014.

[39] D. Ravì, B. Lo, and G.-z. Yang, "Real-time Food Intake Classification and Energy Expenditure Estimation on a Mobile Device," in *International Conference on Body Sensor Networks*, 2015.

[40] K. Kitamura, T. Yamasaki, and K. Aizawa, "Foodlog: Capture, analysis and retrieval of personal food images via web," in *Proc. of the Workshop on Multimedia for Cooking and Eating Activities*, 2009, pp. 23–30.

[41] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, "Menu-match: Restaurant-specific food logging from images," in *Proc. of IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 844–851.

[42] K. Kitamura, T. Yamasaki, and K. Aizawa, "Food log by analyzing food images," in *Proc. of ACM International Conference on Multimedia*, 2008, p. 999. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1459359.1459548

[43] K. Aizawa, G. C. De Silva, M. Ogawa, and Y. Sato, "Food log by snapping and processing images," *Proc. of International Conference on Virtual Systems and Multimedia*, pp. 71–74, 2010.

[44] K. Kitamura, C. D. Silva, T. Yamasaki, and K. Aizawa, "Image processing based approach to food balance analysis for personal food logging," in *Proc. of IEEE International Conference on Multimedia and Expo*, 2010, pp. 625–630.

[45] Y. Maruyama, G. C. De Silva, T. Yamasaki, and K. Aizawa, "Personalization of food image analysis," *Proc. of International Conference on*

Virtual Systems and Multimedia, pp. 75–78, 2010.

[46] G. M. Farinella, D. Allegra, and F. Stanco, "A benchmark dataset to study the representation of food images," in *Workshop on Assistive Computer Vision and Robotics.* Springer, 2014, pp. 584–599.

[47] G. M. Farinella, M. Moltisanti, and S. Battiato, "Food recognition using consensus vocabularies," in *International Workshop on Multimedia Assisted Dietary Management*, 2015.

[48] G. M. Farinella, D. Allegra, F. Stanco, and S. Battiato, "On the exploitation of one class classification to distinguish food vs non-food images," in *International Workshop on Multimedia Assisted Dietary Management*, ser. Lecture Notes in Computer Science, vol. 9281. Springer International Publishing, 2015, pp. 375–383.

[49] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh Fast-food Image Dataset," *Proc. of IEEE International Conference on Image Processing*, pp. 289–292, 2009.

[50] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2249–2256.

[51] G. M. Farinella, M. Moltisanti, and S. Battiato, "Classifying food images represented as bag of textons," in *Proc. of IEEE International Conference on Image Processing*, 2014, pp. 5212 – 5216.

[52] E. Parrish and A. K. Goksel, "Pictorial pattern recognition applied to fruit harvesting," in *Transactions of the ASAE*, no. 20, 1977, pp. 822–827.

[53] P. Levi, A. Falla, and R. Pappalardo, "Image controlled robotics applied to citrus fruit harvesting," in *International Conference on Robot Vision and Sensory Controls*. IFS Publications, 1988.

[54] D. C. Slaughter and R. C. Harrell, "Color vision in robotic fruit harvesting," *Transactions of the American Society of Agricultural and Biological Engineers*, vol. 30, no. 4, pp. 1144–1148, 1987.

[55] ——, "Discriminating fruit for robotic harvest using color in natural outdoor scenes," *Transactions of the American Society of Agricultural and Biological Engineers*, vol. 32, no. 2, pp. 757–763, 1989.

[56] M. Cardenas-Weber, G. Miles, A. Hetzroni, and A. S. of Agricultural Engineers. Summer Meeting, *Machine Vision to Locate Melons and Guide Robotic Harvesting*. American Society of Agricultural and Biological Engineers, 1991.

[57] F. Buemi, M. Massa, and G. Sandini, "Agrobot: a robotic system for greenhouse operations," in *Workshop on Robotics in Agriculture*, 1995, pp. 172–184.

[58] A. R. Jiménez, A. K. Jain, R. Ceres, and J. Pons, "Automatic fruit recognition: a survey and new results using Range/Attenuation images," *Pattern Recognition*, vol. 32, no. 10, pp. 1719–1736, 1999.

[59] T. Brosnan and D. W. Sun, "Improving quality inspection of food products by computer vision - A review," *Journal of Food Engineering*, vol. 61, no. 1, pp. 3–16, 2004.

[60] C.-J. Du and D.-W. Sun, "Learning techniques used in computer vision for food quality evaluation: a review," *Journal of Food Engineering*, vol. 72, no. 1, pp. 39–55, 2006.

[61] S. Gunasekaran, "Computer vision technology for food quality assurance," *Trends in Food Science & Technology*, vol. 7, no. 8, pp. 245–256, 1996.

[62] P. Munkevik, T. Duckett, and G. Hall, "Vision system learning for ready meal characterisation," in *Proc. of International Conference on Engineering and Food*, no. 1, 2004.

[63] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 1–6, 1998.

[64] P. Munkevik, G. Hall, and T. Duckett, "A computer vision system for appearance-based descriptive sensory evaluation of meals," *Journal of Food Engineering*, vol. 78, no. 1, pp. 246–256, 2007.

[65] K. Kiliç, I. H. Boyaci, H. Köksel, and I. Küsmenoglu, "A classification system for beans using computer vision system and artificial neural networks," *Journal of Food Engineering*, vol. 78, no. 3, pp. 897–904, 2007.

[66] D. W. Sun, "Inspecting pizza topping percentage and distribution by a computer vision method," *Journal of Food Engineering*, vol. 44, no. 4, pp. 245–249, 2000.

[67] C. J. Du and D. W. Sun, "Multi-classification of pizza using computer vision and support vector machine," *Journal of Food Engineering*, vol. 86, no. 2, pp. 234–242, 2008.

[68] P. D. Wright, G. Shearing, A. J. Rich, and I. Johnston, "The role of a computer in the management of clinical parenteral nutrition," *Journal of Parenteral and Enteral Nutrition*, vol. 2, pp. 652–657, 1978.

[69] A. J. Rich, “A programmable calculator system for the estimation of nutritional intake of hospital patients,” *The American Journal of Clinical Nutrition*, vol. 34, 1981.

[70] L. Zepeda and D. Deal, “Think before you eat: photographic food diaries as intervention tools to change dietary decision making and attitudes,” *International Journal of Consumer Studies*, vol. 32, no. 6, pp. 692–698, 2008.

[71] G. Shroff, A. Smailagic, and D. P. Siewiorek, “Wearable context-aware food recognition for calorie monitoring,” in *Proc. of International Symposium on Wearable Computers*, 2008, pp. 119–120.

[72] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, “Recognition and volume estimation of food intake using a mobile device,” in *Proc. of IEEE Workshop on Applications of Computer Vision.* IEEE, 2009, pp. 1–8.

[73] M. a. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[74] N. Chen, Y. Y. Lee, M. Rabb, and B. Schatz, “Toward dietary assessment via mobile phone video cameras.” *Annual Symposium*, vol. 2010, pp. 106–110, 2010.

[75] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10 SPEC. ISS., pp. 761–767, 2004.

[76] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Lecture Notes in Computer Science*, vol. 3951 LNCS, 2006, pp. 404–417.

[77] M. Agrawal, K. Konolige, and M. R. Blas, "CenSurE: Center surround extremas for realtime feature detection and matching," in *Lecture Notes in Computer Science*, vol. 5305 LNCS, 2008, pp. 102–115.

[78] J. Dehais, S. Shevchik, P. Diem, and S. G. Mougiakakou, "Food Volume Computation for Self Dietary Assessment Applications," in *Proc. of IEEE International Conference on Bioinformatics and Bioengineering.* IEEE, 2013, pp. 1–4.

[79] G. J. Burghouts and J.-M. Geusebroek, "Performance evaluation of local colour invariants," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 48–62, 2009.

[80] M. Bosch, T. Schap, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Integrated database system for mobile dietary assessment and analysis," in *Proc. of IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6.

[81] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, "Analysis of food images: features and classification," in *Proc. of IEEE International Conference on Image Processing.* IEEE, 2014, pp. 2744–2748.

[82] K. Yanai and T. Joutou, "A Food Image Recognition System With Multiple Kernel Learning," *Proc. of IEEE International Conference on Image Processing*, pp. 285–288, 2009.

[83] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in *Proc. of IEEE International Symposium on Multimedia*, 2010, pp. 296–301.

[84] Y. Matsuda, H. Hoashi, and K. Yanai, "Multiple-Food Recognition Considering Co-occurrence Employing Manifold Ranking," in *Proc. of*

International Conference on Pattern Recognition, no. Icpr, 2012, pp. 2017 – 2020.

[85] ——, “Recognition of multiple-food images by detecting candidate regions,” in *Proc. of IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 25–30.

[86] Y. Kawano and K. Yanai, “Food image recognition with deep convolutional features,” in *Proc. of ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 589–593.

[87] ——, “Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation,” in *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision*, 2014.

[88] W. Wen and Y. Jie, “Fast food recognition from videos of eating for calorie estimation,” in *Proc. of IEEE International Conference on Multimedia and Expo*, 2009, pp. 1210–1213.

[89] Z. Zong, D. T. Nguyen, P. Ogunbona, and W. Li, “On the combination of local texture and global structure for food classification,” in *Proc. of IEEE International Symposium on Multimedia*, 2010, pp. 204–211.

[90] D. D. T. Nguyen, Z. Zong, P. Ogunbona, and W. Li, “Object detection using non-redundant local binary patterns,” in *Proc. of IEEE International Conference on Image Processing*, 2010, pp. 4609–4612.

[91] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. Essa, “Leveraging Context to Support Automated Food Recognition in Restaurants,” in *Proc. of IEEE Winter Conference on Applications of Computer Vision*. Waikoloa: IEEE, 2015, pp. 580–587.

[92] F. Foroni, G. Pergola, G. Argiris, and R. I. Rumiati, "The FoodCast research image database." *Frontiers in human neuroscience*, vol. 7, no. March, p. 51, 2013.

[93] M.-Y. Chen, Y.-H. Yang, C.-J. Ho, S.-H. Wang, S.-M. Liu, E. Chang, C.-H. Yeh, and M. Ouhyoung, "Automatic Chinese food identification and quantity estimation," *SIGGRAPH Asia 2012 Technical Briefs*, pp. 1–4, 2012.

[94] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 Mining Discriminative Components with Random Forests," in *Proc. of European Conference on Computer Vision.* Springer International Publishing, 2014, pp. 446–461.

[95] N. Martinel, C. Piciarelli, C. Micheloni, and G. L. Foresti, "On filter banks of texture features for mobile food classification," in *Proc. of International Conference on Distributed Smart Cameras.* ACM, 2015, pp. 14–19.

[96] P. Pouladzadeh, A. Yassine, and S. Shirmohammadi, "Foodd: Food detection dataset for calorie measurement using food images," in *International Workshop on Multimedia Assisted Dietary Management*, ser. Lecture Notes in Computer Science, vol. 9281. Springer International Publishing, 2015, pp. 441–448.

[97] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravì, "Exploiting textons distributions on spatial hierarchy for scene classification," *Journal on Image and Video Processing*, vol. 2010, p. 7, 2010.

[98] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: The Indian Journal of Statistics*, pp. 401–406, 1946.

[99] S. Marčelja, "Mathematical description of the responses of simple cortical cells*," *Journal of the Optical Society of America*, vol. 70, no. 11, pp. 1297–1300, 1980.

[100] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proc. of IEEE International Conference on Computer Vision*, 2007.

[101] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2009.

[102] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[103] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: A SIFT descriptor with color invariant characteristics," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1978–1983.

[104] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. of Advances In Neural Information Processing Systems*, 2012, pp. 1–9.

[105] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[106] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE, 2007, pp. 1–8.

[107] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. of European Conference on Computer Vision.* Springer, 2010, pp. 143–156.

[108] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[109] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[110] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, 2002.

[111] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Combining global and local features for food identification in dietary assessment," in *Proc. of IEEE International Conference on Image Processing*, 2011, pp. 1789–1792.

[112] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[113] M. H. Rahmana, M. R. Pickering, D. Kerr, C. J. Boushey, and E. J. Delp, "A new texture feature for improved food recognition accuracy in a mobile phone based dietary assessment system," in *Proc. of IEEE International Conference on Multimedia and Expo Workshops*, 2012, pp. 418–423.

[114] Y. Kawano and K. Yanai, "Real-Time Mobile Food Recognition System," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 1–7.

[115] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 480–492, 2012.

[116] C. Pham, D. Jackson, J. Schöning, T. Bartindale, T. Plotz, and P. Olivier, "FoodBoard: surface contact imaging for food recognition," in *UbiComp*, 2013, pp. 749–752.

[117] Tin Kam Ho, "Random decision forests," in *Proc. of International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282.

[118] Y. Hu, X. Cheng, L.-T. Chia, X. Xie, D. Rajan, and A.-H. Tan, "Coherent phrase model for efficient image near-duplicate retrieval," *IEEE Transactions on Multimedia*, vol. 11, no. 8, pp. 1434–1445, 2009.

[119] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[120] X. Qi, R. Xiao, J. Guo, and L. Zhang, "Pairwise rotation invariant co-occurrence local binary pattern." in *Proc. of European Conference on Computer Vision*, vol. 7577, 2012, pp. 158–171.

[121] L. W. Renninger and J. Malik, "When is scene identification just texture recognition?" *Vision Research*, vol. 44, no. 19, pp. 2301–2311, 2004.

[122] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proc. of IEEE In-*

ternational Conference on Computer Vision and Pattern Recognition, 2010, pp. 2249–2256.

[123] C. chung Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," 2001.

[124] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.

[125] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271–1283, 2010.

[126] A. Oliveira, P. Ferrara, A. De Rosa, A. Piva, M. Barni, S. Goldenstein, Z. Dias, and A. Rocha, "Multiple parenting identification in image phylogeny," in *Proc. of IEEE International Conference on Image Processing.* IEEE, 2014, pp. 5347–5351.

[127] Usage of Image File Formats for Websites. http://w3techs.com/technologies/overview/image_format/all.

[128] S. Battiato and M. Moltisanti, "The future of consumer cameras," in *IS&T/SPIE Electronic Imaging.* International Society for Optics and Photonics, 2015, pp. 93 990C–93 990C.

[129] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from jpeg headers," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, 2011.

[130] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.

[131] M. C. Stamm, M. Wu, and K. Liu, "Information forensics: An overview of the first decade," *Access, IEEE*, vol. 1, pp. 167–200, 2013.

[132] A. Bruna, G. Messina, and S. Battiato, "Crop detection through blocking artefacts analysis," in *Proc. of International Conference on Image Analysis and Processing.* Springer, 2011, pp. 650–659.

[133] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2. IEEE, 2007, pp. II–217.

[134] S. Battiato, G. M. Farinella, E. Messina, and G. Puglisi, "Robust image alignment for tampering detection," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1105–1117, 2012.

[135] E. Kee and H. Farid, "Digital image authentication from thumbnails," in *IS&T/SPIE Electronic Imaging.* International Society for Optics and Photonics, 2010, pp. 75 410E–75 410E.

[136] T. Gloe, "Forensic analysis of ordered data structures on the example of jpeg files," in *Proc. of IEEE International Workshop on Information Forensics and Security.* IEEE, 2012, pp. 139–144.

[137] S. Battiato and G. Messina, "Digital forgery estimation into dct domain: a critical analysis," in *Proceedings of the First ACM workshop on Multimedia in Forensics.* ACM, 2009, pp. 37–42.

[138] J. A. Redi, W. Taktak, and J.-L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 133–162, 2011.

[139] F. Galvan, G. Puglisi, A. R. Bruna, and S. Battiato, "First quantization matrix estimation from double compressed jpeg images," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 8, pp. 1299–1310, 2014.

[140] M. Keyvanpour, M. Moradi, and F. Hasanzadeh, "Digital forensics 2.0," in *Computational Intelligence in Digital Forensics: Forensic Investigation and Applications.* Springer, 2014, pp. 17–46.

[141] Camera & Imaging Products Association: Standardization Committee - Exchangeable image file format for digital still cameras: Exif Version 2.3. http://www.cipa.jp/std/documents/e/DC-008-2012_E_C.pdf.

[142] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[143] B. J. Tordoff and D. W. Murray, "Guided-mlesac: Faster image transform estimation by using matching priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1523–1535, 2005.

[144] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE TPAMI*, 2014.

[145] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[146] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. of Advances in Neural Information Processing Systems*, 2006, pp. 545–552.

[147] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE, 2009, pp. 1597–1604.

[148] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition.* IEEE, 2007, pp. 1–8.

[149] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera," in *Proc. of ITS Conference*, Sep. 2012, pp. 975–982.