UNIVERSITY OF CATANIA

Department of Mathematics and Computer Science

TRUE SCENE UNDERSTANDING: CLASSIFICATION, SEMANTIC SEGMENTATION AND RETRIEVAL

DANIELE RAVÌ

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.

SUPERVISOR:
Prof. Sebastiano Battiato

DIRECTOR OF GRADUATE STUDIES:
Prof. Vincenzo Cutello

PhD Course in Computer Science - XXVI Cycle

True Scene Understanding:

Classification, Semantic Segmentation and Retrieval

by

Daniele Ravì

Abstract

The huge volume of images shared in the web sites and on personal archives has provided us challenges on massive multimedia management. Due to the well-known semantic gap between human-understandable high-level semantics and machine generated low-level features, recent years have witnessed plenty of research effort on multimedia content understanding and indexing. Computer vision algorithms for individual tasks such as object recognition, detection and segmentation have reached impressive results. The next challenge is to integrate all these algorithms and address the problem of the complete scene understanding, which involves explaining the image by recognizing all the objects of interest and their spatial extent or shape. True semantic understanding of an image mainly involves the scene classification and the semantic segmentation. The former has the aim to determinate the categories to which an image belongs. The later instead, provide for each pixel a semantic label, which describes the category of object where it appears. Solutions for the semantic interpretation and understanding of images will enable and enhance large variety of computer vision applications. While a human can do these tasks easily, it is laborious and the sheer quantity of data involved can make it prohibitive for a computer. This thesis proposes novel approaches for semantic scene categorization, segmentation and retrieval that enable a device with a limited amount of resources to understand images automatically. The proposed computer vision solutions use machine-learning algorithms to build robust and reusable systems. Since learning is a key component of biological vision systems, the design of automatic artificial systems that are capable to learn, is one of the most important trends in modern computer vision research.

Thesis Supervisor: Sebastiano Battiato
Title: Associate Professor

Acknowledgments

Completing a PhD is truly a marathon event, and I would not have been able to complete this journey without the aid and support of countless people over the past three years. I must first express my gratitude towards my supervisor, S. Battiato. His leadership, supports, attention to details and hard work have set an example, which I hope to match a day. I would like to express my gratitude to the prof. M. Bober whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his assistance in this last year. It allowed me to complete successfully the work related on the semantic segmentation.

I would like to express my gratitude also to Dr. G. M. Farinella and V. Tomaselli, for being outstanding advisors. Their constant encouragement, support, and precious suggestions made this work successful. I would like to thank StMicroelectronics and especially the engineer M. Guarnera for funding my PhD.

I am deeply and forever indebted to my parents, my brother and my girlfriend, N. Farajidavar, for their love, support and encouragement throughout this period.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vision consists of processing images of scenes so as to make explicit what needs to be known about them [101]. Visual categorization is a fundamental cognitive process that refers to the ability to group visual stimuli into meaningful categories. This aptitude allows humans to efficiently and rapidly analyse their surroundings. Humans Vision System (HVS) is able to understand complex visual scenes at a single glance, despite the number of objects with different poses, colours, shadows and textures that may be contained in the scenes. The reason of the robustness and rapidness of this human ability has been a focus of investigation for the cognitive sciences community over many years [135].

In this thesis, we investigate the image understanding process from three different points of view. The first one is the basic "scene classification" that has the aim to understand the context (selected from a predefined set of classes) of a query image. The second one is the "semantic segmentation" that has the aim to understand the object classes for each individual pixels (or group of pixels) of a query image. The last one is the "image indexing" that has the aim to extract from a predefined database all the images containing the same scene represented in the query (see Fig. 1-1).

The Human Visual System and related studies of Cognitive Sciences community have stimulated researches in Computer Vision to build artificial image understanding systems. Motivations beyond that of pure scientific curiosity are provided by several important applications: content-based image retrieval (CBIR) [150], object detection and recognition [138], semantic organization of image databases [88], place recognition for robot nav-

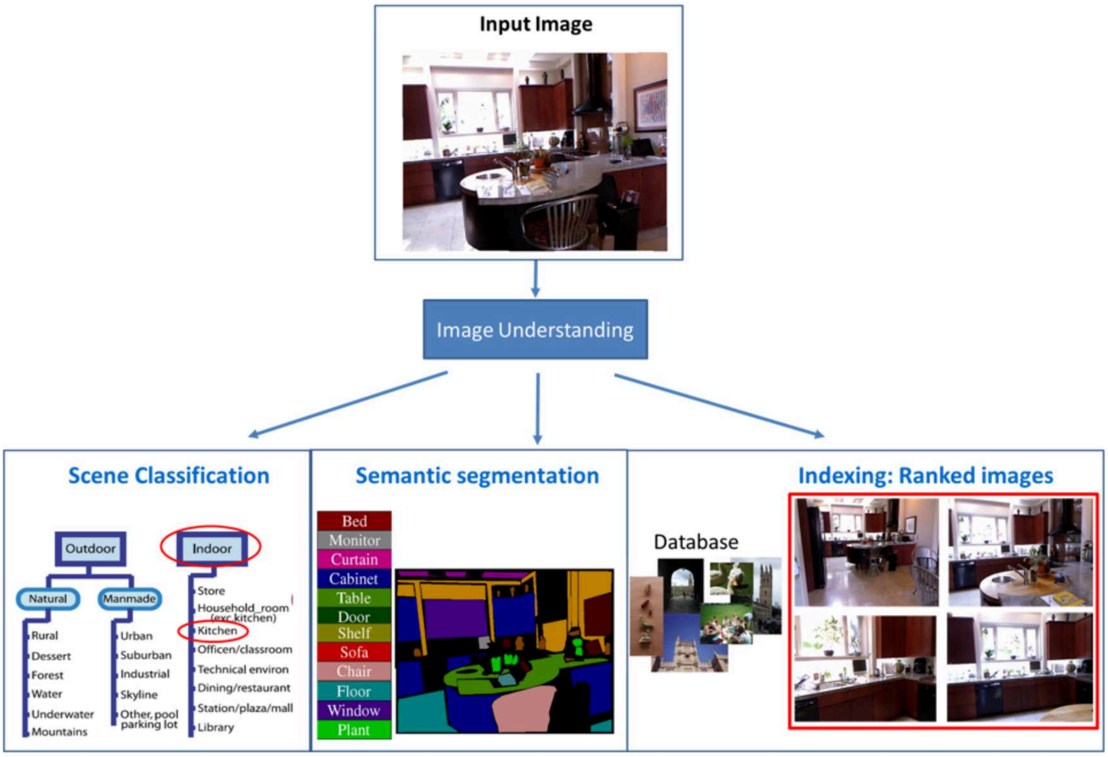


Figure 1-1: Three different points of view for the image understanding process

igation systems [139], direct marketing on multimedia messaging services domain (MMS) [17, 19].

The rest of this thesis is organized as follows: sections 1.1, 1.2 and 1.3 introduce fundamental concepts and state of the art in Computer Vision for the image categorization, semantic segmentation and image-indexing problems, respectively. The publications achieved and the work plans followed during my Phd carrier are listed in section 1.5. In chapter 2, 3 and 4 we focus on describing the proposed solutions for the three aforementioned problems. Finally, conclusions and avenues for further research are given in chapter 5.

1.1 Scene Classification

Scene recognition is a key process of human vision which is exploited to efficiently and rapidly understand the context and objects in front of us. Humans are able to recognize complex visual scenes at a single glance, despite the number of objects with different poses, colors, shadows and textures that may be contained in the scenes. Seminal studies in computational vision [101] have portrayed scene recognition as a progressive reconstruction of the input from local measurements (e.g., edges, surfaces). In contrast, some experimental studies have suggested that recognition of real-world scenes may be initiated from the encoding of the global configuration, bypassing most of the details about local concepts and object information [27]. This ability is achieved mainly by exploiting the holistic cues of scenes that can be processed as single entity over the entire human visual field without requiring attention to local features [113]. Successive studies suggest that the humans rely on local as much as on global information to recognise the scene category [151].

The recognition of the scene is a useful task for many relevant computer vision applications: robot navigation systems [139], semantic organization of databases of digital pictures [88], content-based image retrieval (CBIR) [150], context driven focus attention and object priming [138, 142], scene depths estimation [143]. To build a scene recognition system, consideration about the spatial envelope properties (e.g., degree of naturalness, degree of openness, etc.) and the level of description of the scene (e.g., subordinate, basic, superordinate) should be taken into account [112].

The results reported in [26] demonstrate that a context recognition engine is important for the tuning of color constancy algorithms used in the Imaging Generation Pipeline (IGP) to improve the quality of the final generated image. More in general, in the research area of single sensor imaging devices [9], the scene context information can be used to drive different tasks performed in the IGP during both acquisition time (e.g., autofocus, auto-exposure, white balance, etc.) and post-acquisition time (e.g., image enhancement, image coding). For example, the auto-scene mode within cameras could allow to automatically set the acquisition parameters and hence to improve the perceived quality of the captured image according to the recognised scene (e.g., Landscape, Portrait, etc.). Furthermore, context recognition could be functional for the automatic setting of surveillance cameras which can be usually placed in different scene contexts (e.g., Indoor vs Outdoor scenes, Open vs Closed scenes, etc.), as well as in the application domain of assistive technologies for visually impaired and blind people (e.g., indoor vs outdoor). The need for the development of effective solution for scene recognition systems to be embedded in consumer imaging devices (e.g., consumer digital cameras, smartphones, etc.) is confirmed by the growing interest of consumer devices industry which are including those capabilities in their products (e.g., Nikon, Canon, etc.). Different constraints should be considered in transferring the ability of scene recognition into the IGP of a single sensor imaging devices [56]: memory limitation, low computational power, as well as the input data format to be used in scene recognition task (e.g., JPEG images).

The visual content of the scene can be described with local or global representation models. A local based representation of the image describes the context of the scene as a collection of previously recognized objects/concepts within the scene, whereas a global (or holistic) representation of the scene context considers the scene as a single entity, bypassing the recognition of the constituting concepts (e.g., objects) in the final representation. The representation models can significantly differ for their capability of extracting and representing important information for the context description.

Many Computer Vision researchers have proved that holistic approaches can be effectively used to solve the problem of rapid and automatic context recognition. Most of the holistic approaches share the same basic structure that can be schematically summarized as follows:

1. A suitable features space is considered (e.g., textons vocabularies [10]). This space must emphasize specific image cues such as, for example, corners, oriented edges, textures, etc.

2. Each image under consideration is projected into the considered feature space. A descriptor is built considering the image as a whole entity (e.g., textons distributions [10]).

3. Context recognition is obtained by using Pattern Recognition and Machine Learning algorithms on the computed representation of the images (e.g., by using K-nearest neighbours, SVM, etc.).

A wide class of techniques based on the above scheme, works extracting features on perceptually uniform color spaces (e.g., CIELab). Typically, filter banks [18, 119] or local invariant descriptors [29, 91] are employed to capture image cues and to build the visual vocabulary to be used in a bag of visual words model [45]. An image is considered as a distribution of visual words and this holistic representation is used for classification purposes. Spatial information have been also exploited in order to capture the layout of the visual words within images [18, 91]. A review of some other state-of-the-art methods working with features extracted on spatial domain can be found in [30].

On the other hand, different approaches have considered the frequency domain as an useful and effective source of information to holistically encode an image for scene classification. The statistics of natural images on frequency domain [141] reveal that there are different spectral signatures for different image categories. In particular by considering the shape of the FFT spectrum of an image it is possible to address scene category [112, 140, 141], scene depth [143], and object priming such as identity, scale and location [142].

As suggested by different studies in computational vision, scene recognition may be initiated from the encoding of the global configuration of the scene, disregarding details and object information. Inspired by this knowledge, Torralba and Oliva [140] have introduced computational procedures to extract the global structural information of complex natural scenes looking at the frequency domain [112, 140, 141]. The computational model presented in [140] works in the Fourier domain where Discriminant Structural Templates (DSTs) are built using the power spectrum. A DST is a weighting scheme over the power spectrum that assigns positive values to the frequencies that are representative for one class and negative for the others. In particular the sign of the DST values indicates the correlation between the spectral components and the "spatial envelope" properties of the two groups to be distinguished. When the task is to discriminate between two kinds of scenes (e.g., *Natural* vs. *Artificial*) a suitable DST is built and used for the classification. A DST is learned in a supervised way using Linear Discriminant Analysis. The classification of a new image is hence performed by the sign of the correlation between the power spectrum of the considered image and the DST. A relevant issue in building a DST is the sampling of the power spectrum both at the learning and classification stages (a bank of Gabor filters with different frequencies and orientation is used in [140]). The final classification is performed on the Principal Component of the sampled frequencies. The improved version of the DST descriptor is called GIST [112,141]. Oliva and Torralba [112] performed test using GIST on a dataset containing pictures of 8 different environmental scenes covering a large variety of outdoor places. The GIST descriptor is nowadays one of the most used representation to encode the scene as whole. It has been used in many computer vision application domains, such as robot navigation [139], visual interestingness [66], image retrieval [50], video summarization [97], etc.

Luo and Boutell [98] built on previous works of Torralba and Oliva [140] and proposed to use Independent Component Analysis rather than PCA for features extraction. In addition they have combined the camera metadata related to the image capture conditions with the information provided by the power spectra to perform the final classification.

Farinella et al. [59] proposed to exploit features extracted by ordering the Discrete Fourier Power Spectra (DFPS) to capture the *naturalness* of scenes. By ordering the DFPS the overall “shape” of the scene in frequency domain is captured. In particular the frequencies that better capture the differences in the energy “shapes” related to *Natural* and *Artificial* categories are selected and ordered by their response values in the Discrete Fourier power spectrum. In this way a “ranking number” (corresponding to the relative position in the ordering) is assigned to each discriminative frequency. The vector of the response values and the vector of the relative positions in the ordering of the discriminative frequencies are then used singularly or in combination to provide a holistic representation of the scene. The representation was used with a probabilistic model for *Natural* vs *Artificial* scene classification.

The Discrete Cosine Transform (DCT) domain was explored by Farinella et al. [56] to build histograms of local dominant orientations to be used as scene representation at the abstract level of description (e.g., *Natural* vs *Artificial*, *Indoor* vs *Outdoor*, etc.). The representation is built collecting the information about orientation and strength of the edges related to the JPEG image blocks [88]. This representation was coupled with a logistic classifier to discriminate between the different scene contexts.

The aforementioned techniques disregard the spatial layout of the discriminative frequencies. Seminal studies proposed by Torralba et al. [138, 142, 143] have proposed to further look at the spatial frequency layout to address more specific vision tasks by exploiting contextual information (e.g., object detection and recognition, scene depth estimation, etc.).

1.2 Semantic Segmentation

Semantic segmentation and scene classification can be considered vision tasks with a direct link. The semantic segmentation, is an extension of the scene classification problem where the entity to classify is not anymore the whole image but single pixel or group of pixels. Moreover, usually the scene classification is required as a subsystem in the semantic segmentation solution.

Semantic segmentation aims at pixel-wise classification of images according to semantically meaningful regions (e.g., objects). Semantic interpretation and understanding of images is an important goal in visual recognition and a solution for this task will enable or enhance a large variety of applications such as visual search, scene classification, object detection and recognition. A precise automated image segmentation is still a challenging and an open problem. Local structures, shape, colour and texture are the common features deployed in the semantic segmentation task. Colour or gray level information are essential core features used to segment image into regions [3, 32]. An efficient and computationally light descriptor to build on colour features is the colour histogram. The histogram ignores the spatial organization of the pixels, which is generally an advantage as it support rotation and scale invariance. When spatial organization is required a second order statistic can be used. The most common second order statistical measures are based on the correlation function between the image pixels. Image correlogram [74] describes the correlation of the image colours as a function of their spatial distance. Local structures (i.e., edges, corners, and T-Junctions) are also useful features that are detected by differential operators commonly applied to the luminance information. The shape is one of the most important characteristic of an object; an outline or boundary contour. It allows to discriminate different objects. Texture is finally a visual cue that describe the luminosity fluctuations in the image, which let us interpret a surface as a whole part. Textures can

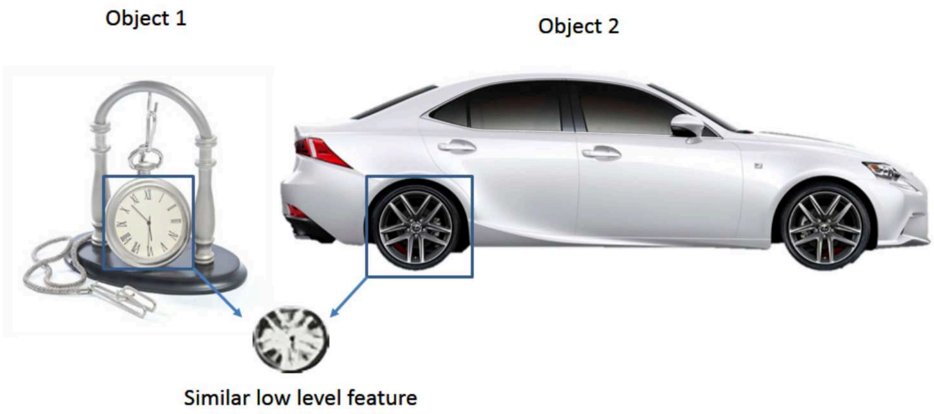


Figure 1-2: Example of different objects with similar low level features.

be characterized using properties such as regularity, coarseness, contrast and directionality. Textures contains important information about the structural arrangement of the surface. It also describes the relationship of the surface to the surrounding environment. One immediate application of image texture is the recognition of image regions using texture properties. Texture features can be extracted by using various methods. Gray-level occurrence matrices (GLCMs) [106], Gabor Filter [94], and Local binary pattern (LBP) [110] are examples of popular methods to extract texture features. Other method to obtain texture features are the fractals representation [146].

The key step to obtain a reliable semantic segmentation system is the selection and design of robust and efficient features that are capable of distinguishing the predefined pixels' classes, such as grass, car, human, etc. The following criteria should be taken into account while considering the design of the overall system and the features extraction method for the considered problem:

- Similar low-lavel features response, can represent different objects (see Fig. 1-2). Each feature alone is hence not adequate for segmenting the object that they belong to. A spatial arrangement of low-level features can be used to increase the object discrimination.

- A semantic segmentation approach is strongly dependent of the nature of the application and each application may have different requirements including different input data type. For example, there are some applications aim to segment images obtained from fluorescence microscope, some other applications use images from video surveillance camera and others use normal photos. Another important parameter that is application dependent is the detail coarseness of required segmentation as shown in Fig. 1-3.

- The information needed for the labelling of a given pixel may come from very distant pixels. The category of a pixel may depend on relatively short-range information (e.g. the presence of a human face generally indicates the presence of a human body nearby), as well as on very long-range dependencies [55].
- The hardware miniaturization has reach impressive levels stimulating the deployment of new devices such as smartphones and tablets. These devices, though powerful, do not have yet the performance of a typical desktop computer. These devices require algorithms that perform on board, complex vision tasks including the semantic segmentation. For these reasons, the segmentation algorithms and associated features should be designed to ensure good performance for computationally limited devices.

Different methods were proposed to address these challenges. Some approaches are region-based as in [25,28,36,52,69,85,86,99,118] other approaches use a multiscale scanning window detector such as Viola-Jones [148] or Dalal-Triggs [46], possible augmented with part detectors as in Felszenszwalb et al. [60] or Bourdev et al. [31]. Other approaches as in [7, 136] unify these paradigms into a single recognition architecture, and leverage on their strengths by designing region-based specific object detectors and combining their outputs. Most of the methods proposed in literature are based on probabilistic models such as the Markov Random Field (MRF) and the Conditional Random Fields (CRF) models. For example, a nonparametric model is proposed in [137]. This approach requires no training and it can easily scaled to datasets with tens of thousands of images and hundreds of labels. It works by scene-level matching with global image descriptors, followed by superpixel-level matching with local features and efficient MRF based optimization for incorporating neighbourhood context. In [162], instead, a framework is presented for semantic scene parsing and object recognition based on dense depth maps. Five view independent 3D features that vary with object class are extracted from dense depth maps at a superpixel level

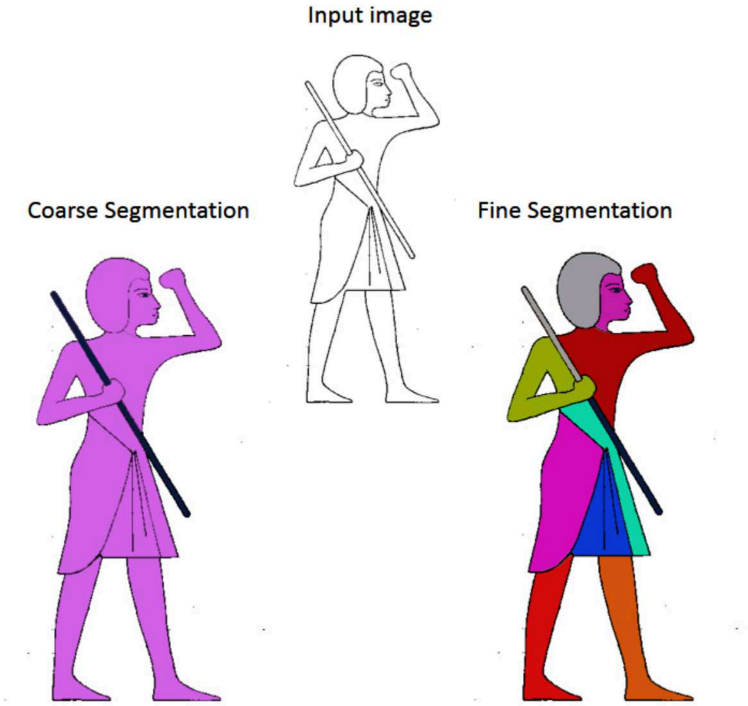


Figure 1-3: Example of image segmentation output with different coarse level. In the left a coarse segmentation reveals a unified object. In the right a fine segmentation reveals multiple sub-region for the same object.

for training a classifier using randomized decision forest technique. The formulation can integrates multiple features in the MRF framework to segment and recognize different object classes in the query street scene images. The result shows that only using dense depth information, results in more accurate segmentation and recognition than that obtained from sparse 3D features or appearance separately, or even the combination of sparse 3D features and appearance. In the Texton boost technique [129] the segmentation is obtained by implementing a CRF and features that automatically learn layout and context information. Similar features were also proposed in [49], although textons were not used, and responses were not aggregated over a spatial region. In contrast with these techniques, the shape context technique in [25] uses a hand-picked descriptor. In [132] a framework is presented for pixel-wise object segmentation of road scenes that combines motion and appearance features. It is designed to handle street-level imagery such as that on Google Street View and Microsoft Bing Maps. The authors formulate the problem in the CRF framework in order to probabilistically model the label likelihoods and the a priori knowledge. An extended set of appearance-based features is used, which consists of textons, colour, location and HOG descriptors. A novel boosting approach is then applied to combine the motion and appearance-based features. The authors also incorporate higher order potentials in the CRF model, which produce segmentations with precise object boundaries. In [65] a novel formulation is proposed for the scene-labelling problem capable to combine object detections with pixel-level information in the CRF framework. Since object detection and multi-class image labelling are mutually informative dependent problems, pixel-wise segmentation can benefit from the powerful object detectors and vice versa. The main contribution of [65] lies in the incorporation of top-down object segmentations as generalized robust potentials into the CRF formulation. These potentials present a principled manner to convey soft object segmentations into a unified energy minimization framework, enabling joint optimization and thus mutual benefit for both problems. A probabilistic framework is presented in [87] for reasoning about regions, objects, and their attributes such as object class, location, and spatial extent. The proposed CRF is defined on pixels, segments and objects. The authors define a global energy function for the model, which combines results from sliding window detectors and low-level pixel-based unary and pairwise relations. It addresses the problems of what, where, and how many by recognizing objects, finding their locations and spatial extent and segmenting them. Although the MRF and the CRF are adequate models to deal with the semantic segmentation problem in terms of performance, they represent a bottleneck in the computation, because the inference is a highly resources consuming process. A better approach with good performance while preserving high efficiency is based on the random forest. For example in the semantic texton forests [128], the authors show that one can build powerful texton codebooks without computing expensive filter-banks or descriptors, and without performing costly k-means clustering and nearest-neighbour assignment. Specifically, the authors propose the bag of Semantic Textons that is an extension of the bag of word model obtained by combining a histogram of the hierarchical visual word with a region prior category. Fine details of this approach are explained in section 3.1. The STF exploits features that are capable to describe very simple textures obtained in the colour channels. In chapter 3, we extend the STF approach and improve the semantic segmentation performance by adding selected DCT features. These features are aimed to describe more complex textures represented in the frequency domain.

1.3 Image Indexing

In this section and in chapter 4 we focus on the image indexing problem under a forensics context. Image Forensics is a science which, among the other questions, aims to answer the following one during investigation: is the image under consideration contained in a specific digital archive? The increasing use of low cost imaging devices and the availability of large databases of digital photos makes the near duplicate image retrieval (NDIR) task a common activity for a number of applications. In particular, NDIR in large databases (such as popular social networks, collections of surveillance images and videos, or digital investigation archives) is a key ingredient for different forensics activities. During digital investigation (e.g., for copyright violation, child abuse, etc.), classic hashing techniques (e.g., MD5 [120], SHA1 [48], etc.) are commonly used to index large quantities of images in order to detect copies in different archives. However, these methods are unsuitable to find altered copies, even in case of slight modifications (e.g., near duplicates). Indeed, classic hashing techniques usually fail because just a small change in the image (even a single bit) will, with overwhelming probability, results in a completely different hash code. For example, two images depicting a scene of crime are perceptually identical under small viewpoint changes, partial occlusion, and/or low photometric distortions, but their hash code is completely different when a classic hashing approach is used to check their similarity. In order to cope with all related problems, robust hashing techniques based on image content must be developed: perceptually identical images in terms of content should have the same (or at least very similar) hash value with high probability, while perceptually different images should have independent hash values. Most of the near duplicate detection techniques based on image content exploit the bag of visual word approach to build the image signature [21, 73, 156, 164, 165]. A problem of the bag-of-visual-word based methods is related to the ambiguity of some generated visual words [130, 145]. On the other hand, since different descriptors represent different aspects of a local region, there is no single descriptor which is superior to the others [105]. Recently, some commercial approaches for robust content based hashing methods have been proposed for photos (PhotoDNA [1]) and videos (Videntifier [93]). These techniques make use of the recent developments in the field of Near Duplicate Image (NDI) retrieval. Note that there is no agreement on the technical definition of near-duplicates (see [47] for an in-depth discussion). The definition of near duplicate depends on the degree of variability (photometric and geometric) that is considered acceptable for each particular application. Some approaches [82] consider as NDI, images obtained by slightly modifying the original ones through common transformations such as changing contrast or saturation, scaling, cropping, etc. Other techniques [73] consider as NDI, images of the same scene but with different viewpoint and illumination. A drawback in testing near duplicate retrieval approaches is that usually near duplicate images used in the experiments are synthetically generated from a set of images or correspond to different frames of a video, hence there is an high correlation in terms of visual content, and there is no variability in terms of resolution and compression. To better evaluate the different algorithms it is needed a database composed by images depicting the same scene and/or subject whose have been acquired by different cameras, with different viewpoint, luminance condition, and variability in terms of background. In the last few years, different image hashing techniques have been proposed in literature to cope with image retrieval and near-duplicate image detection problems. Most of these techniques are based on the Bags of Visual Words paradigm (BoVW) [95, 134] to build a holistic representation of the images. Ke et al. [82] detected near-duplicate images by employing local descriptors [105] extracted on interest points [104] to represent and match images under several transformations. They used a hash-based indexing technique to efficiently search into the image databases, and also applied an optimized storage layout to further improve efficiency. Chum et al. [42] proposed two novel image similarity measures for image indexing through local feature descriptors and enhanced min-Hash techniques. The authors of [43] introduced a method to combine visual words with geometric information to improve hashing-based image retrieval and object detection, obtaining a novel algorithm (called Geometric min-Hash) which shows significant advantages against geometrical deformations and occlusions. Cheng et al. [40] considered local dependencies among descriptors both in scale and space, and encoded not only visual appearance but also their scale and space co-occurrence. Moreover they built SuperNodes that embody the neighbor information to speed up the retrieval execution time. Wu et al. [156] proposed a novel scheme to exploit geometrical constraints by spatially grouping features to improve the retrieval precision. Spatial verification stage to re-rank the results with the bag of-words model have been also exploited by Philbin et al. in [115]. Wang et al. [155] combined appearance-based and keypoint-based methods. The algorithm is able to extract and match keypoints from images by discarding outliers through a voting procedure based on the affine invariant ratio of normalized lengths. Later, to further validate the correspondences, the algorithm compares the color histograms of the corresponding areas which have been previously identified by the matched points. In [158], Xu et al. proposed a two stage method based on the Spatial Pyramid Matching (SPM) technique [91] and image blocks alignment through linear programming [159]. The aim of the cascade is to deal with spatial shifts and scale variation; both transformations frequently occur between frames of a video. Since there is an increasing interest in the scalability of the Bag-of-Words based near duplicate visual search paradigm, a method to parallelize the near duplicate visual search architecture to index images over multiple servers have been proposed by Rongrong et al. in [78]. Near duplicate image retrieval based on BoVW paradigm has been also exploited for the annotation of web videos as addressed by Zhao et al. in [152, 165]. Taking into account their previous work [164], the authors extract keypoints from keyframes and generate a visual dictionary by using a clustering algorithm. Each keyframe is then described by a BoVW representation. Moreover, to speed up the keyframe retrieval, inverted file indexing plus Hamming embedding is employed. A re-rank strategy based on a weak geometric consistency checking is also proposed to improve the overall performance of the system. The final similarity of a video is obtained considering both the scores of keyframes and their temporal consistency with respect to the query video. The memory usage and query-response time are two of the main issues in the retrieval task. The problem of compressing the visual codebook to better handle with storage and retrieval complexity has been studied by Rongrong et a. in [79]. In a recent study of Hu et al. [73], the BoVW paradigm has been augmented by using multiple descriptors (Bags of Visual Phrases) to exploit the coherence between different feature spaces in which local image regions are described. Specifically, to reduce the amount of false matchings in the BoVW model the authors of [73] introduced the coherent phrase model. In this model, a local image region (i.e., the patch surrounding the local interest point [104]) is described by a visual phrase of multiple descriptors instead of a visual word of a single descriptor. In the Bags of Visual Phrases approach, both feature (local regions are described by descriptors of different types) and spatial coherence (multiple descriptors are obtained from local areas at different sizes) are taken into account. To further improve the Bags of Visual Phrases model, taking into account our preliminary work [20], we 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, which leads to a more distinctive representation. In particular, to reduce the amount of false matchings, instead of separately obtain the codebooks corresponding to the different feature spaces as proposed in [73], we generate the final codebooks taking into account the correspondence of the clusters of the involved spaces of descriptors to further enforce feature correspondence. To properly perform tests, a new image database of near duplicate images has been built by collecting images from Flickr [61] and private collections. The dataset contains 3148 images of 525 different scenes which have from 3 to 34 real near duplicates. Finally, a method to compress the image representation to be stored for near duplicate purposes is suggested. The experiments performed on the aforementioned dataset show the effectiveness of the proposed approach, which obtains a good margin of performances with respect to the approach described in [73].

1.4 Content-aware image resizing

Content-aware image resizing techniques allow to take into account the visual content of images during the resizing process. The basic idea beyond these algorithms is the removal of vertical and/or horizontal paths of pixels (i.e., seams) containing low salient information. In appendix A we present a method which exploits the Gradient Vector Flow (GVF) of the image to establish the paths to be considered during the resizing. The relevance of each

GVF path is straightforward derived from an energy map related to the magnitude of the GVF associated to the image to be resized. To make more relevant the visual content of the images during the content-aware resizing, we also propose to select the generated GVF paths based on their visual saliency properties. In this way visually important image regions are better preserved in the final resized image. The proposed technique has been tested, both qualitatively and quantitatively, by considering a representative dataset of 1000 images labeled with corresponding salient objects (i.e., ground-truth maps). Experimental results demonstrate that our method preserves crucial salient regions better than other state-of-the-art algorithms.

1.5 Work plans and achieved publications

I developed my PhD experience through an academic/industrial research. Just before starting my PhD, I worked in the AST lab of STMicroelectronics as a consultant. Implementing a Red-eye removal algorithm using machine learning and computer vision techniques was the project that I have been assigned to. The results achieved are concluded in different papers and journals. Noteworthy are the chapter book [12] and the patent [103].

By the end of my project, my supervisors, Eng. M. Guarnera, commended me for my communication and creative problem solving skills, and for my ability to work well with different people within a development team. In November 2010, I started my PhD with the topic of "developing machine learning and computer vision algorithms for image understanding". The PhD became a great opportunity for learning new techniques and fundamental mathematics in this field. All this happen thanks to numerous summer schools (VISMAC 2010, ICVSS 2011, ISSPR 2011 and ICVSS 2012), course and seminars, which I attended, following helpful advices of my well-known supervisors G.M. Farinella, S. Battiato (University supervisor) and V. Tomaselli (Industrial supervisor). During my PhD, several works have been completed and published. The first work is related to the implementation of a scene classification algorithm for mobile phone. There, we defined a novel set of fast features for the problem of image classification using DCT coefficients. We obtained good performance in terms of accuracy and low time complexity allowing the algorithm being runnable on a low power system like a smartphones. The software has been released for different platform (based on Windows, Maemo/Meego and Android Os) and the results of this work are published in [13, 14, 22, 57, 58].

The second work that I was assigned is related to the implementation of a system for detection near duplicate images in large databases. Specifically we propose to further improve the Bags of Visual Phrases approach considering the coherence between feature spaces not only at the level of image representation, but also during the codebook generation phase. Finally, we suggest also a method to compress the proposed image representation for storage purposes. Experiments show the effectiveness of the proposed near duplicate retrieval technique, which outperforms the original Bags of Visual Phrases approach. The results of this work are published in [23].

The third project is related to the implementation of a content-aware image resizing framework. The basic idea beyond this algorithm is the resizing of an image by considering vertical and/or horizontal paths of pixels (i.e., seams) which contain low salient information. In this work we exploit the Gradient Vector Flow (GVF) of the image to establish the paths to be considered during the resizing. The relevance of each path is derived from a saliency map obtained by considering the magnitude of the GVF associated to the image under consideration. The proposed technique has been tested, both qualitatively and quantitatively, by considering a representative set of images labelled with corresponding salient objects (i.e., ground-truth maps). Experimental results demonstrate that our method preserves crucial salient regions better than other state-of-the-art algorithms. The results of this work are published in [15, 16].

In the last year of my PhD, I had the possibility to carry on the research as a visitor researcher at the Centre for Vision, Speech and Signal Processing (CVSSP) of the University of Surrey, supervised by the prof. M. Bober. Here another task has been successfully completed. Specifically we implemented an extension of the Semantic Texton Forest approach that includes novel texture features. The results of this work will be submitted soon in a journal. Other academic works was also accomplished through my PhD. For example, I have taken part on several external projects for image/video forensic analysis (noteworthy is the project aimed to verify the image tampering of scientific results within medical imaging papers), I also carried out reviews for different conferences (Oxford Journal, JEI, TCSVT, IPMU, JET, TCSVT, ICIAP and VISAPP) and presented my works in seminars and in tutoring activities. The following list briefly reports all the achieved publications organized by topic:

- Red-Eye Removal
  - G. Messina, D. Ravì, M. Guarnera, G. M. Farinella, Method and apparatus for filtering red and/or golden eye artifact, US Application number 12969252, 30 June 2011
  - S. Battiato, G. M. Farinella, M. Guarnera, G. Messina, D. Ravì, "A Cluster-Based Boosting Strategy for Red-Eyes Removal" - Chapter in Modern Image Processing Algorithms Employing Computational Intelligence Techniques - Patrick Siarry, Amitava Chatterjee Eds. - Springer (2012) (INPRESS)
- Scene Classification
  - G. M. Farinella, D. Ravì, V. Tomaselli, M. Guarnera, S. Battiato "Representing Scenes For Real-Time Context Classification on Mobile Devices" Journal Pattern Recognition 2013 (SUBMITTED)
  - S. Battiato, G. M. Farinella, M. Guarnera, D. Ravì, V. Tomaselli, "Instant Scene Recognition on Mobile Platform" European Conference on Computer Vision (ECCV) 2012
  - S.Battiato,G. M. Farinella, E. Messina, G. Puglisi, D. Ravì, V. Tomaselli, A. Capra - "On the performances of computer vision algorithms on mobile platforms" IS&T/SPIEElectronic Imaging 22-26 January 2012 Burlingame, California United States
  - S. Battiato, G. M. Farinella, E. Messina, G. Puglisi, D. Ravì - "Computer Vision on Mobile Devices: A few case studies" STDAY 2011 September 30, 2011 - Torino - Italy

  - G. M. Farinella, D. Ravì, "Image Categorization", Chapter in Image Processing for Embedded Devices, Applied Digital Imaging ebook series - Bentham Science Publisher, ISSN: 1879-7458, 2010.

- Indexing

  - S. Battiato, G. M. Farinella, G. Puglisi, Member, D. Ravì: "Aligning Codebooks for Near Duplicate Image Detection" Multimedia Tools and Applications 2013

- Content Aware Image Resizing

  - S. Battiato, G. M. Farinella, G. Puglisi, D. Ravì: "Saliency Based Selection of Gradient Vector Flow Paths for Content Aware Image Resizing" IEEE Transactions on Image Processing 2013 (SUBMITTED)
  - S. Battiato, G. M. Farinella, G. Puglisi and D. Ravì, "Content-Aware Image Resizing With Seam Selection Based on Gradient Vector Flow", International Conference on Image Processing. (ICIP) 2012

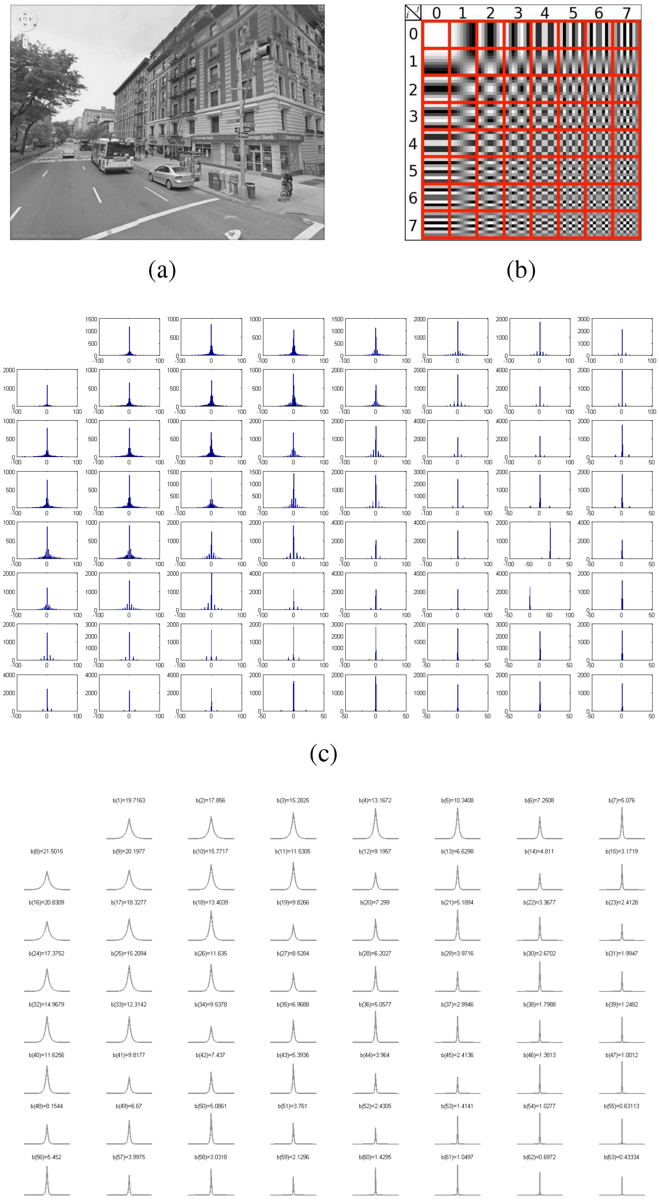
Chapter 2

Scene Classification

This chapter presents a new computational model to represent the context of the scene based on the image statistics collected in the Discrete Cosine Transform (DCT) domain. Since the DCT of the image acquired by a device is always computed for JPEG conversion/storage[1], the feature extraction process, useful to compute the signature of the scene context, is "free of charge" for the IGP and can be performed in real-time independently from the computational power of the device. The rationale beyond the proposed image representation is that the distributions of the AC DCT coefficients (with respect to the different AC DCT basis) differ from one class of context to another and can be used to discriminate the content. The statistics of the AC DCT coefficients can be approximated by a Laplacian distribution [89] almost centered at zero; we extract an image signature which encodes the statistics of the scene by considering the scales of Laplacian models fitted over the distribution of AC DCT coefficients of the image under consideration (See Fig. 2-1). This signature computed on a spatial pyramid [10, 91], together with the information on colors obtained considering the DC components, is then used for the automatic scene context categorization.

To reduce the computational complexity involved in the image representation extraction, only a subset of the DCT frequencies (summarizing edges and textures) are considered. To this purpose a supervised greedy based selection of the most discriminative frequencies is performed. To improve the discrimination power, the spatial envelope of the scene is encoded by a spatial hierarchy approach useful to collect the AC DCT statistics

[1] JPEG is the most common used format for images and videos.



(a)

(b)

(c)

(d)

Figure 2-1: Given the luminance channel of an image (a), the feature vector associated to the context of the scene is obtained considering the statistics of the AC coefficients corresponding to the different AC DCT basis (b). For each AC frequency, the coefficients distribution is computed (c) and fitted with a Laplacian model (d). Each fitted Laplacian is characterized by a scale parameter (i.e., related to the slope). The final image signature is obtained collecting the scale parameters of the fitted Laplacians among the different AC DCT coefficient distributions. As specified in Section 2.2, information on colors (i.e., DC components) as well as on the spatial arrangement of the DCT feature can be included to obtain a more discriminative representation.

on image sub-regions [10, 91]. We have coupled the proposed image representation with a Support Vector Machine classifier for final context recognition purpose. The experiments performed on the 8 Scene Context Dataset demonstrate that the proposed image representation achieves better results with respect to the popular GIST scene descriptor [112]. Moreover, the novel image signature outperforms GIST in terms of computational costs. Finally, with the proposed image descriptor we obtain results comparable with other more complex state-of-the-art methods exploiting spatial pyramids [10].

The primary contribution of this work is related to the new descriptor for scene context classification purpose. We emphasise once again the fact that the proposed descriptor is built on information already available in the IGP of single sensor devices as well as in any image coded in JPEG format. Compared to many other scene descriptors extracted starting from RGB images [10, 26, 29, 91, 112, 119, 151], the proposed model has the following peculiarities/advantages:

- the decoding/decompression of JPEG is no needed to extract the scene signature;
- visual vocabularies are not necessary to be computed and maintained in memory to represent training and test images;
- the extraction of the scene descriptor does not need complex operation such as convolutions with bank of filters or domain transformations (e.g., FFT);
- there is no need of a supervised/unsupervised learning process to build the scene descriptor (e.g., there is no need of pre-labeled data and/or clustering procedure);
- it can be extracted directly into an IGP with low computational resources;
- the recognition results closely match state-of-the-art methods cutting down the computational resources (e.g., computational time needed to compute the image representation).

The remainder of this chapter is organized as follows: Section 2.1 gives the background about the AC DCT coefficients distributions for different image categories. Section 2.2 presents the proposed image representation, whereas the new Image Generation Pipeline architecture is described in Section 2.3. Section 2.4 reports the details about the experimental settings and discusses the obtained results.

2.1 The Statistics of Natural Image Categories in DCT domain

One of the most popular standard for lossy compression of images is the JPEG [153]. The JPEG compression is available in every IGP of single sensor consumer devices such as digital cameras and smartphones. Moreover, most of the images on Internet (e.g., in social networks, websites) are stored in JPEG format. Nowadays, around 70% of the total images on the top 10 million websites are in JPEG format[2]. Taking into account these facts, a scene context descriptor that can be efficiently extracted in the IGP and/or directly in the JPEG compressed domain is desirable.

The JPEG algorithm divides the image into non-overlapping blocks of size 8×8 pixels and each block is then processed with the Discrete Cosine Transform (DCT) before quantization and entropy coding. The DCT has been studied by many researchers which have proposed different models for the distributions of the DCT coefficients. One of the first conjecture was that the AC coefficients have Gaussian distributions [116]. Different other possible distributions of the coefficients have also been proposed, including Cauchy, generalized Gaussian, as well as a sum of Gaussians [51,53,107,131,160]. The knowledge about the mathematical form of the statistical distribution of the DCT coefficient is useful in quantizer design and noise mitigation for image enhancement. Although methods to extract features directly from JPEG compressed domain have been presented in literature for the application context of image retrieval [37, 126], for the best of our knowledge, there aren't works in literature where the DCT coefficients distributions are exploited for scene classification. The proposed image representation is inspired by the works of Lam [89, 90], where the semantic content of the images has been characterised in terms of DCT distributions modelled with Laplacian and generalized Gaussian models.

[2] Source: `http://w3techs.com/technologies/overview/image_format/all`. The statistics is computed on the top 10 million websites according to the Amazon.com company (Nov 2013).

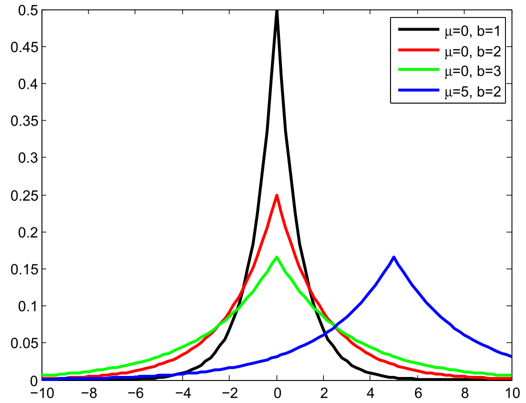


Figure 2-2: Laplacian distribution at varying of μ and b.

After performing the DCT on each of the blocks of an image and collecting the corresponding coefficients to the different AC basis of the DCT, a simple observation of the distribution indicates that they resemble a Laplacian (see Fig. 2-1(c)). This guess has been demonstrated through a rigorous mathematical analysis in [89]. The probability density function of a Laplacian distribution can be written as:

$$f(x|\mu, b) = \frac{1}{2b} exp\left(-\frac{|x-\mu|}{b}\right) \tag{2.1}$$

where μ is a location parameter and $b \geq 0$ is a scale parameter. Fig. 2-2 reports the examples of Laplacian distributions. At varying of the scale parameter, the Laplacian distribution changes its shape. Given N samples $\{x_1, \ldots, x_N\}$, the parameters μ and b can be simply estimated with the maximum likelihood estimator [109]. Specifically, μ corresponds to the median of the samples[3], whereas b is computed as follows:

$$b = \frac{1}{N} \sum_{i=1}^{N} |x_i - \mu|. \tag{2.2}$$

The rationale beyond the proposed representation for scene context classification is that the context of different classes of scenes differs in the scales of the AC DCT coefficient distributions. Hence, to represent the context of the scene we can use the feature vector of the scales of the AC DCT coefficients distributions of an image after a simple Laplacian fitting. Fig. 2-3 reports the average "shapes" of the AC DCT coefficient Laplacian dis-

[3] Note that for the different AC DCT distributions the μ value is not equal to zero.

tributions related to the 8 Scene Context Dataset [112]. The dataset contains 2600 color images (256x256 pixels) belonging to the following 8 outdoor scene categories: *coast*, *mountain*, *forest*, *open country*, *street*, *inside city*, *tall buildings* and *highways*. The Laplacian shapes in Fig. 2-3 are computed by fitting the Laplacian distributions for the different AC DCT coefficient of the luminance channel of each image and then averaging the Laplacians parameters with respect to the 8 different classes (color coded in Fig. 2-3). A simple observation of the slopes of the different Laplacian distributions (corresponding to the b parameter) is useful to better understand the rationale beyond the proposed scene descriptor. The slopes related to the different classes can be captured by the b parameters computed (with low computational cost) from the images directly encoded in the DCT domain (i.e., JPEG format). The guess is that the multidimensional space of the b parameters is discriminative enough for scene context recognition. Although it is difficult to visualize the N-dimensional distributions of the b parameters, a simple intuition of the discriminativeness of the space can be obtained considering two AC DCT frequencies and plotting the 2-dimensional distributions of the related Laplacian parameters. Fig. 2-4 shows the 2-dimensional distributions obtained by considering two DCT frequencies corresponding to the DCT basis $(0,1)$ and $(1,0)$ which are useful to reconstruct the vertical/horizontal edges of each image block (see Fig. 2-1(b)). As the figure points out, already considering only two AC DCT frequencies there is a good separation among the different classes. The experiments reported in Section 2.4 quantitatively confirm our rationale.

2.2 Proposed Image Representation

In this section we formalize the proposed image representation which builds on the main rationale that different scene classes have different AC DCT coefficient distributions (see Section 2.1). Fig. 2-3 shows the average of the AC DCT coefficient distributions after a Laplacian fitting on images belonging to different scene contexts. Differences in the slopes of the Laplacian distributions are evident and are related to the different classes. As a consequence of this observation, we propose to encode the scene context by concatenating all the Laplacian parameters related to the median and slope (μ and b) which are computed

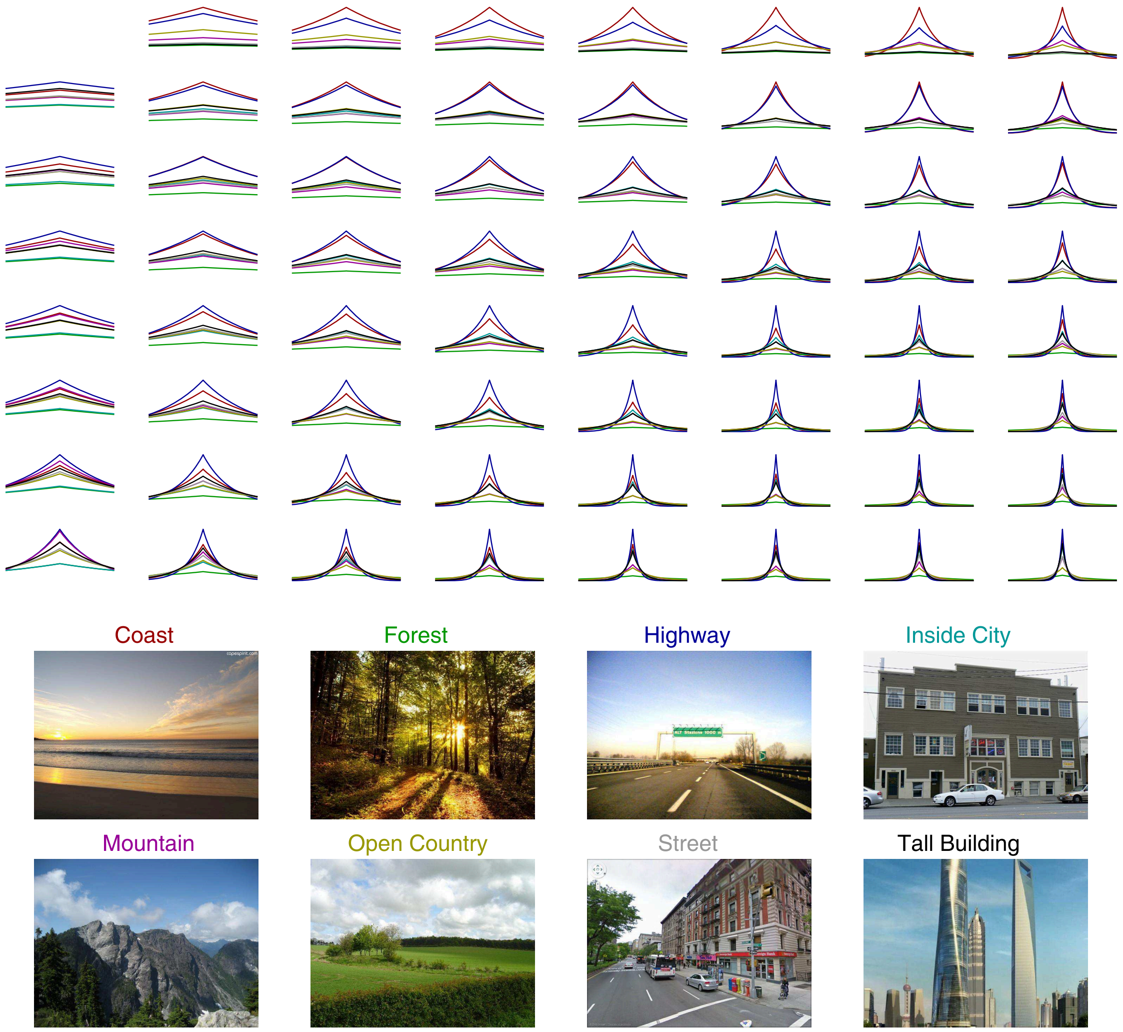


Figure 2-3: Average Laplacian distributions of the AC DCT coefficients considering the 8 Scene Context Dataset [112]. The different scene classes are color coded.

by considering the different AC DCT coefficients distributions of the luminance channel of the image [4]. In addition to these information, the mean and variance of the DC coefficients can be also included into the feature vector to capture the color information, as well as the AC DCT Laplacian distributions parameters obtained considering the C_b and C_r channels[5]. In Section 2.4 we show the contribution of each component involved in the proposed image descriptor.

The aforementioned image features are extracted in the IGP just after the image acqui-

[4] Note that in the JPEG format the image is converted in the YC_bC_r color model as first step.

[5] The DCT chrominance exhibits the same distribution as for the luminance channel [89].

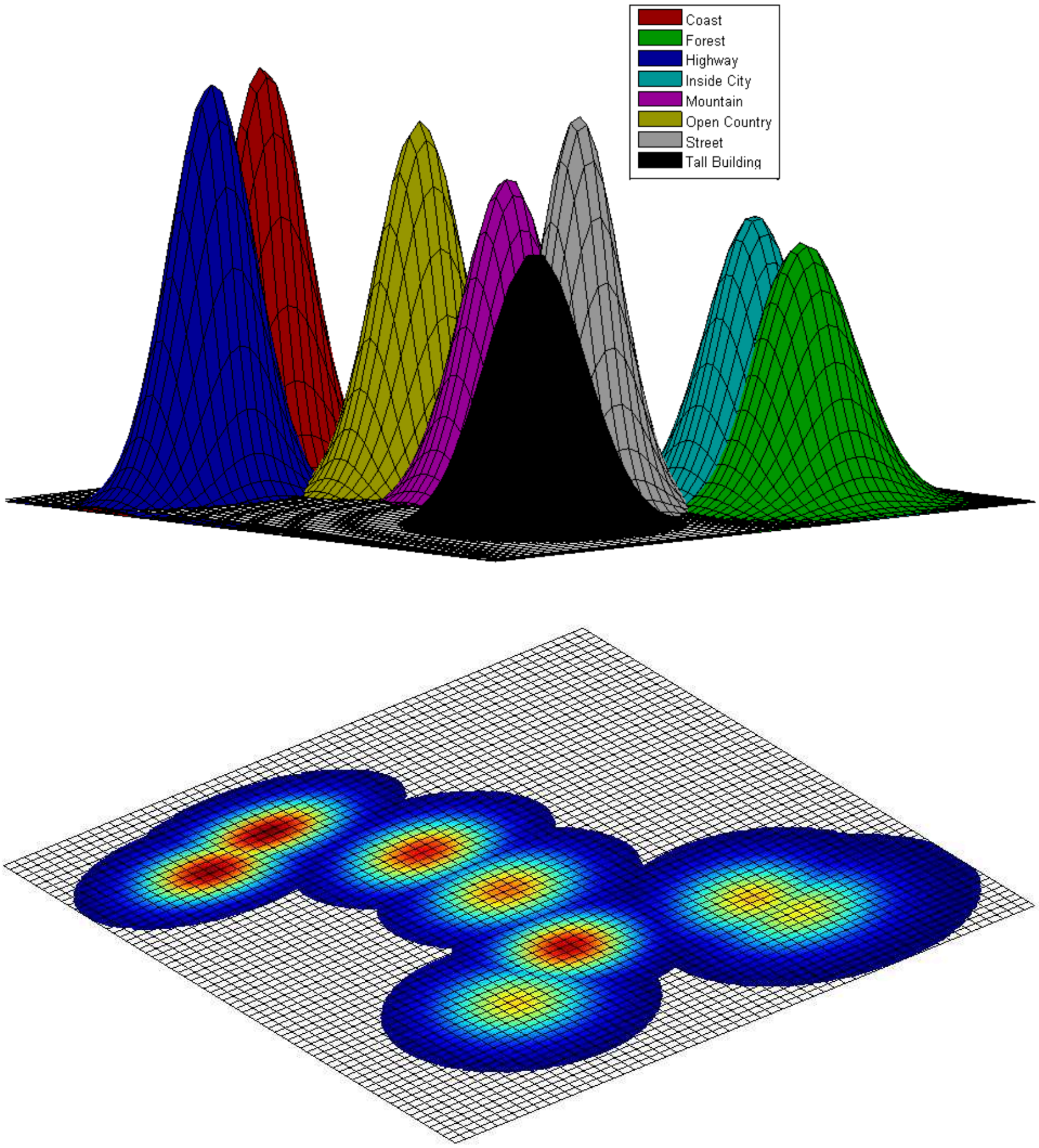


Figure 2-4: 2-dimensional distributions (fitted with a Gaussian model) related to the Laplacian distribution parameters of the DCT frequency $(0,1)$ and $(1,0)$ in Fig. 2-1 (b).

sition step, without any extra complex processing. Specifically, the Laplacian parameters related to the AC DCT coefficients are obtained collecting the AC DCT coefficients inside the JPEG encoding module performed before the image storage. In case the image is already stored in JPEG (e.g., a picture from the web), the information useful for scene context representation can be directly collected in the compressed domain without any further processing. Indeed, to build the scene descriptor in the DCT compressed domain, only simple operations (i.e., the median and the mean absolute deviations from the median) are needed to compute μ and b for the different image channels, as well as to compute the mean and variance on the DC components. This cuts down the computational complexity with respect to other descriptors which usually involve convolution operations (e.g., with bank of filters [10] or Gaussian Kernels [112]) or other more complex pipelines (e.g., Bag of Words representation [91]) to build the final scene context representation.

It is well-known that some of the DCT basis are related to the reconstruction of edges of an 8×8 image block (i.e., first row and first column of Fig. 2-1 (b)), whereas the others are more related to the reconstruction of the textured blocks. As shown in [161] the most prominent patterns composing natural images are the edges. High frequencies are usually affected by noise and could be not really useful for discriminating the context. For this reason we have performed an analysis to understand which of the AC DCT basis can give a real contribution to discriminate between different classes of scenes. One more motivation to select only the most discriminative AC DCT frequencies is the reduction of the complexity of the overall system.

To properly select the AC DCT frequencies to be employed in the final image representation, we have collected (from Flickr) and labelled a set of 847 uncompressed images to be used as validation set. These images belong to the 8 different classes of scene context [112] (see Fig. 2-3) and have variable size (max size 6000×4000, min size 800×600). We used uncompressed images to avoid that the selection processes of the most discriminative frequencies could be biased by the JPEG quantization step. On this dataset we have performed scene context classification by representing images through the Laplacian fitting of a single AC DCT basis. This step has been repeated for each AC DCT basis. A greedy fashion approach has been hence employed to select the most discriminative frequencies. This means that as first round the classification has been performed for all the AC DCT basis separately. The images have been hence classified after performing the learning of a support vector machine. A leave one out modality has been used to evaluate the discriminativeness of each AC DCT basis. Then we have selected the most discriminative

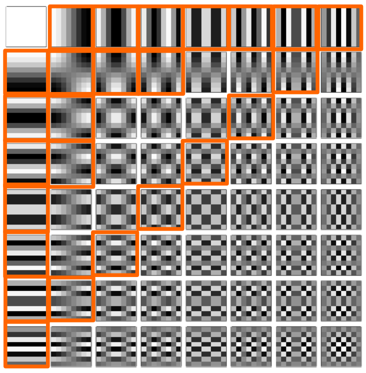


Figure 2-5: Final AC DCT frequencies considered for representing the context of the scene (marked in red).

frequency and we have performed another round of learning and classification considering the selected frequency coupled with one of the remaining in order to jointly consider two AC DCT basis. This procedure has been recursively repeated to greedily select frequencies. The experiments on the validation set suggested that a good trade-off between context classification accuracy and computational complexity (i.e., the number of AC DCT frequencies to be included in a real IGP to fit with required computational time and memory resources) is the one which considers the AC DCT frequencies marked in red in Fig. 2-5. Let $D(i, j)$, $i = 1, \ldots, 7$, $j = 1, \ldots, 7$, be the DCT components corresponding to the 2D DCT basis (i, j) in Fig. 2-1(b). The final set of the selected AC DCT basis in Fig. 2-5 is defined as

$$F = \{(i,0)|i = 1,\ldots,7\} \bigcup \{(i,1)|i = 1,\ldots,3\} \bigcup \{(0,j)|j = 1,\ldots,7\} \bigcup \{(1,j)|j = 1,\ldots,3\} \bigcup \{(i,j)|i = 0\ldots,7; j = 7-i\}. \tag{2.3}$$

Table 2.1 reports the accuracy obtained on the aforementioned validation dataset considering the Laplacian fitting of all the 63 AC DCT basis, as well as the results obtained considering the 25 selected basis in Eq. 2.3 (see Fig. 2-5). Notice that the overall accuracy obtained with the only 25 selected AC DCT basis is higher than the one obtained by considering all the 63 AC DCT basis. This is due to the fact that high frequencies (i.e., the ones below the diagonal in Fig. 2-5) could contain more noise information than the other frequencies, making confusion into the feature space.

The scene context descriptor proposed so far, uses a global feature vector for describing an image leaving out the information about the spatial layout of the local features. The

Table 2.1: Accuracy of scene context classification on the validation dataset.

Approach	*Accuracy*
All Frequencies (63 AC DCT basis)	0.7410
Selected AC DCT basis (Eq. 2.3)	0.7549
Selected AC DCT basis (Eq. 2.3) and spatial hierarchy	0.8233

relative position of a local descriptor can help to disambiguate concepts that are similar in terms of local descriptor. For instance, the visual concepts "sky" and "sea" could be similar in terms of local descriptor, but they are typically different in terms of position within the scene. The relative position can be thought as the context in which a feature takes part with respect to the other features within an image. To encode information of the spatial layout of the scene, different pooling strategies have been proposed in literature [10,91]. Building on our previous work [10] we have augmented the image representation discussed above by collecting the AC DCT distributions over a hierarchy of sub-regions. Specifically, the image is partitioned using three different modalities: horizontal, vertical and regular grid. These schemes are recursively applied to obtain a hierarchy of sub-regions as shown in Fig. 2-6. For each sub-region at each resolution level, the Laplacian parameters (μ and b) over the selected AC DCT coefficients are computed and concatenated to compose the feature vector, thus introducing spatial information. As in [10] we have used three levels in the hierarchy. The integral imaging approach [10, 149] is exploited to efficiently compute the Laplacian parameters of the different AC DCT coefficients. The accuracy obtained on the validation set, by considering the spatial hierarchy based representation was 0.8233%, improving the previous result of more than 6% (see Table 2.1).

We can formalize the proposed scene descriptor as following. Let $r^{l,s}$ be a sub-region of the image under consideration at level $l \in \{0, 1, 2\}$ of the subdivision scheme $s \in S = \{Horizontal, Vertical, Grid\}$ (see Fig. 2-6)[6]. Let $H^{l,s}$ and $W^{l,s}$ be the number of 8×8 blocks of pixel with respect to the height and width of the region $r^{l,s}$. We indicate with the notation $B^{l,s}_{h,w,c}$, $h = 1, \ldots, H^{l,s}$, $w = 1, \ldots, W^{l,s}$, an 8×8 block of pixels of the region $r^{l,s}$ considering the color channel $c \in \{Y, C_b, C_r\}$. Let $D^{l,s}_{h,w,c}$ be the DCT components obtained from $B^{l,s}_{h,w,c}$ through a 2-dimensional DCT processing. We indicate

[6] Note that we define $r^{0,s}$ as the entire image under consideration for every $s \in S$.

with $D^{l,s}_{h,w,c}(i,j)$, $i = 1, \ldots, 7$, $j = 1, \ldots, 7$, the DCT components corresponding to the 2D DCT base (i, j) of Fig. 2-1(b). Let F be the set of the selected AC DCT basis defined above (Eq. 2.3). Then, the scene context descriptor of the region $r^{l,s}$ is computed as follows:

$$\mu^{l,s}_c(0,0) = \frac{1}{H^{l,s}W^{l,s}} \sum_{h=1}^{H^{l,s}} \sum_{w=1}^{W^{l,s}} D^{l,s}_{h,w,c}(0,0) \tag{2.4}$$

$$b^{l,s}_c(0,0) = \frac{1}{H^{l,s}W^{l,s}} \sum_{h=1}^{H^{l,s}} \sum_{w=1}^{W^{l,s}} (D^{l,s}_{h,w,c}(0,0) - \mu^{l,s}_c(0,0))^2 \tag{2.5}$$

where Eq. 2.4 and 2.5 are evaluated for each $c \in \{Y, C_b, C_r\}$. The features in Eqs. 2.4 and 2.5 are related to the DC components of the DCT.

$$\mu^{l,s}_c(i,j) = Median\left(\{D^{l,s}_{h,w,c}(i,j) | h = 1, \ldots, H^{l,s}; w = 1, \ldots, W^{l,s}\}\right) \tag{2.6}$$

$$b^{l,s}_c(i,j) = \frac{1}{H^{l,s}W^{l,s}} \sum_{h=1}^{H^{l,s}} \sum_{w=1}^{W^{l,s}} \left| D^{l,s}_{h,w,c}(i,j) - \mu^{l,s}_c(i,j) \right| \tag{2.7}$$

where Eq. 2.6 and 2.7 are evaluated for each $(c, i, j) \in \{c \in \{Y, C_b, C_r\}; (i, j) \in F\}$. The features in Eqs. 2.6 and 2.7 are related to the 25 selected AC components of the DCT.

Let $[\boldsymbol{\mu}^{l,s}, \mathbf{b}^{l,s}]$ be the feature vector related to the region $r^{l,s}$ computed considering the Eqs. 2.4, 2.5, 2.6 and 2.7. The final image representation is obtained concatenating the representations $[\boldsymbol{\mu}^{l,s}$, $\mathbf{b}^{l,s}]$ of all the sub-regions in the spatial hierarchy. The computational complexity to compute the proposed image representation is linear with respect to the number of 8 × 8 blocks composing the image region under consideration.

2.3 The image generation pipeline architecture

In this section we describe the system architecture to embed the scene context classification engine into an Image Generation Pipeline. The overall scheme is shown in Fig. 2-7. The "Scene Context Classification" module is connected to the "DCT" module. The "High resolution Pipe" block represents a group of algorithms devoted to the generation of high

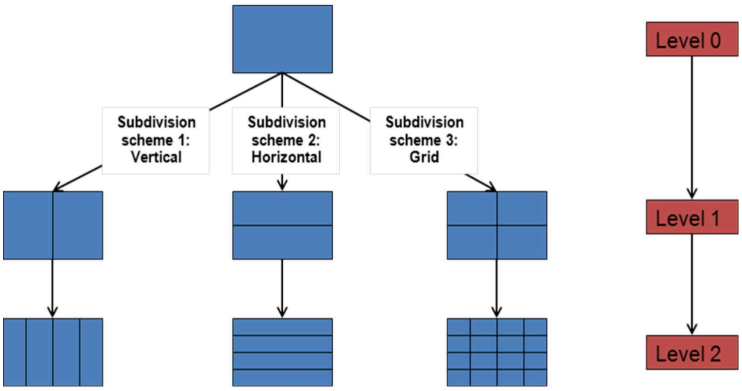

Subdivision scheme 1: Vertical
Subdivision scheme 2: Horizontal
Subdivision scheme 3: Grid
Level 0
Level 1
Level 2

Figure 2-6: Hierarchical subdivision of the image.

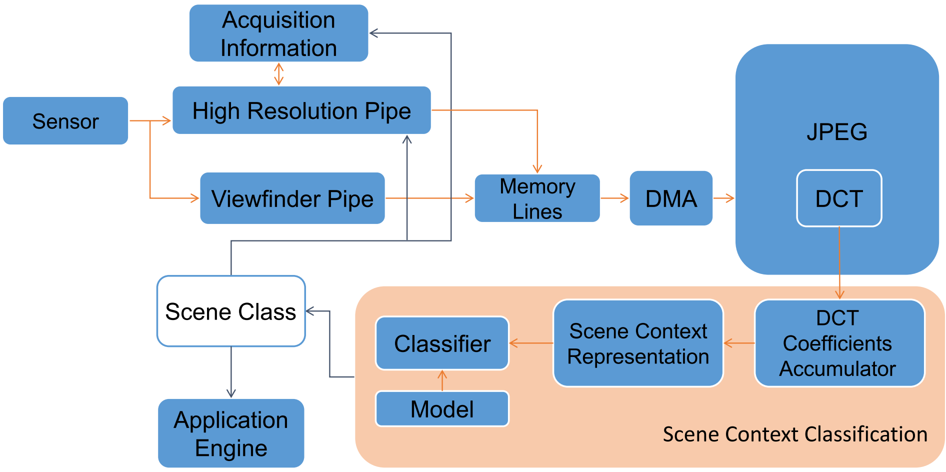


Figure 2-7: Architecture of the IGP including the proposed scene context classification engine.

resolution images. This block is linked to the “Acquisition Information” block devoted to collect different information related to the image (e.g., exposure, gain, focus, white balance, etc.). These information are used to capture and process the image itself. The “Viewfinder Pipe” block represents a group of algorithms which usually work on downscaled images to be shown in the viewfinder of a camera. The “Scene Context Classification” block works taking the input from the viewfinder pipe to determine the scene class of the image. The recognized class of the scene influence both the “Acquisition Information” and the “High resolution pipe” blocks in setting the parameters for the image acquisition. Moreover, the information obtained by the “Scene Context Classification” block can be exploited by the “Application Engine” block which can perform different operations according to the detected scene category. The “Memory lines” and “DMA” blocks provide the data arranged in 8×8 blocks to the “DCT” module for each image channel (Y, C_b, C_b). The sub-blocks, composing the “Scene Context Classification” module, are described in the next subsections.

2.3.1 DCT Coefficients Accumulator

This block is directly linked to the “DCT” block, and thus it receives the DCT coefficients for the luminance and both chrominance channels. With reference to the hierarchical scheme shown in Fig. 2-6, this block accumulates DCT coefficients in histograms starting from the configuration having the smallest region size (e.g., level 2 of grid subdivision). For all the larger regions in the hierarchy, the computations can be performed by merging corresponding histogram bins previously computed at fine resolution level (e.g., the information already computed at level 2 can be exploited to compute the table at level 1 of grid subdivision).

2.3.2 Scene Context Representation

Starting from the histograms obtained by the “DCT Coefficients Accumulator” block, all the pair of Laplacian parameters (μ and b) are computed by using the Laplacian fitting equations presented in Section 2.2. The scene context representation is then obtained by concatenating all the computed Laplacian parameters related to the selected DCT frequencies of all the sub-regions in the hierarchy for the three channels composing the image. In addition to this information, the mean and variance of the DC coefficients upon the hierarchy are computed exploiting the equations presented in Section 2.2.

2.3.3 Classifier

The “Classifier” block takes the feature vector (i.e., the scene context representation) as input to perform the final scene context classification. It takes into account a classifier learned offline (i.e., the block “Model” in Fig. 2-7 which is learned out of the device). A Support Vector Machine is employed in our system architecture.

2.4 Experimental Settings and Results

In this section we report the experiments performed to quantitatively assess the effectiveness of the proposed scene context descriptor with respect to other related approaches. In particular, we compare the performances obtained by the proposed representation model with respect to the ones achieved by the popular GIST descriptor [112]. Moreover, since the proposed representation is obtained collecting information on a spatial hierarchy, we have compared it with respect to the one which use bags of textons on the same spatial hierarchy [10]. Finally, we describe how the architecture presented in Section 2.3 has been implemented on an IGP of a mobile device to demonstrate the effectiveness and the real-time performances of the proposed method. Experiments have been done by using a SVM and a 10-fold cross-validation protocol on each considered dataset. The images are first partitioned into 10 folds by making a random reshuffling of the dataset. Subsequently, 10 iterations of training and testing are performed such that within each iteration a different fold of the data is held-out for testing while the remaining folds are used for learning. The final results are obtained by averaging over the 10 runs.

2.4.1 Proposed Representation vs GIST

To perform this comparison we have taken into account the scene dataset used in the paper introducing the GIST descriptor [112]. The dataset is composed by 2688 color images with resolution of 256×256 pixels (JPEG format) belonging to 8 scene categories: *Tall Building*, *Inside City*, *Street*, *Highway*, *Coast*, *Open Country*, *Mountain*, *Forest*. This dataset, together with the original code for computing the GIST descriptor are available on the web [111]. To better highlight the contribution of the different components involved in the proposed representation (see Section 2.2) we have considered the following configurations in representing the images (Table 2.2):

(A) Laplacian parameters of the 63 AC DCT components computed on Y channel;

(B) Laplacian parameters of the 25 selected AC DCT components computed on Y channel;

Table 2.2: Configurations of the proposed image representation.

Representation Configuration	**DCT Frequencies**	**Image Channels**	**Spatial Hierarchy**
(A)	All 63 AC components	Y	No
(B)	Selected 25 AC components	Y	No
(C)	Selected 25 AC components	Y	Yes
(D)	Selected 25 AC components + DC component	Y	Yes
(E)	Selected 25 AC components + DC component	YC_bC_r	No
(F)	Selected 25 AC components + DC component	YC_bC_r	Yes

(C) Laplacian parameters of the 25 selected AC DCT components computed on Y channel and spatial hierarchy with 3 levels ($l = 0, 1, 2$);

(D) Laplacian parameters of the 25 selected AC DCT components computed on Y channel, mean and variance of the DC DCT components computed on Y channel, and spatial hierarchy with 3 levels ($l = 0, 1, 2$);

(E) Laplacian parameters of the 25 selected AC DCT components computed on YC_bC_r channels, mean and variance of the DC DCT components computed on YC_bC_r channels;

(F) Laplacian parameters of the 25 selected AC DCT components computed on YC_bC_r channels, mean and variance of the DC DCT components computed on YC_bC_r channels, and spatial hierarchy with 3 levels ($l = 0, 1, 2$)).

Fig. 2-8 reports the average per class accuracy obtained considering all the above representation configurations together with the results obtained employing the GIST descriptor. The results show that the scene representation which considers only the Laplacian parameters of the 25 selected AC DCT frequencies fitted on the Y, i.e., the configuration (B), already obtains an accuracy of 75.20%. Encoding the information on the spatial hierarchy, i.e., configuration (C), is useful to improve the results of more than 6%. A small, but still useful, contribution is given by the color information obtained considering the DC DCT components, i.e., configuration (D). Table 2.3 reports the confusion matrix related to the proposed representation with configuration (F), whereas Table 2.4 shows the confusion matrix obtained by employing the GIST descriptor. The proposed representation obtains better results with respect to the GIST descriptor in both cases with and without spatial hierarchy (our with spatial hierarchy: 85.25%, our without spatial hierarchy: 84.60%, GIST:

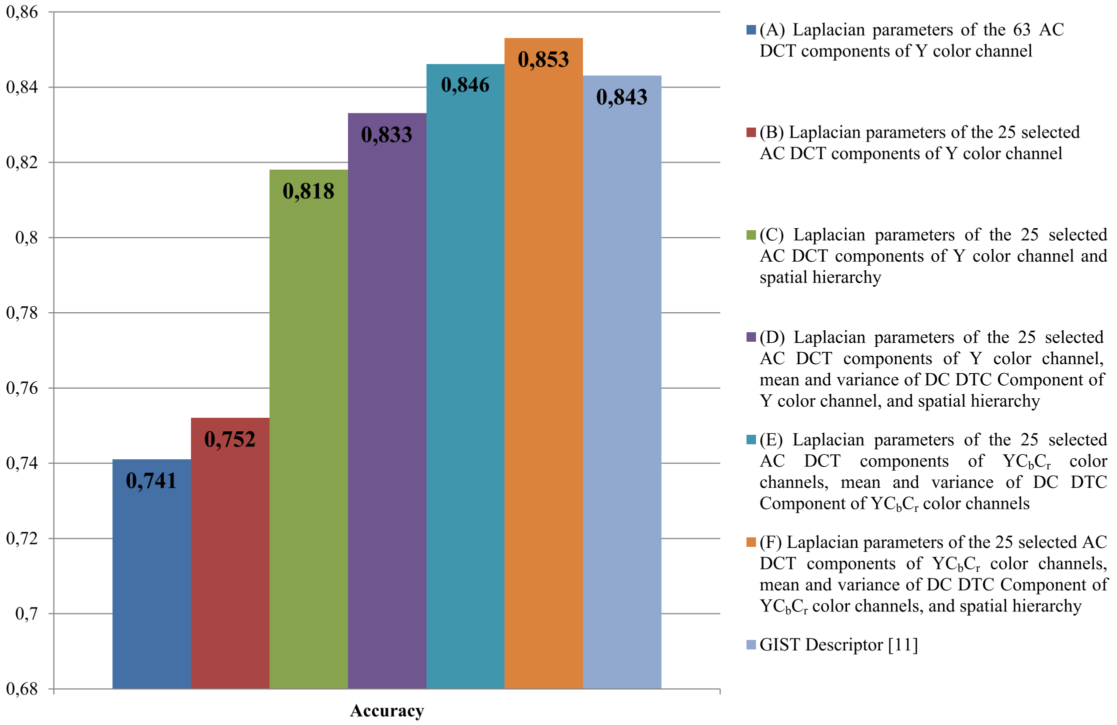


Figure 2-8: Contribution of each component involved in the proposed image representation and comparison with respect to the GIST descriptor [112].

84.28%). One should not overlook that the proposed representation has a very limited computational overhead for the image signature generation because it is directly computed on DCT coefficients already available from the JPEG encoder/format. Specifically, the computation of the image representation (F) requires about 1 operation per pixel (i.e., it is linear with respect to the image size). This highly reduces the complexity of a scene recognition system. Moreover, differently than GIST descriptor, the proposed representation is suitable for mobile platforms since the DCT is already embedded in the Image Generation Pipeline, whereas the GIST descriptor needs extra overhead in computing the signature of the image and it employs operations which are not present in the current IGP of single sensors imaging devices (e.g., FFT on the overall image).

Further tests have been done to demonstrate the effectiveness of the proposed representation in discriminating the *Naturalness* and *Openness* of the scene [112]. Specifically, taking into account of the definition given in [112], the *Naturalness* of the scene is related to the structure of a scene which strongly differs between man-made and natural environ-

Table 2.3: Results obtained by exploiting the proposed image representation (F) on the 8 Scene Context Dataset [112]. Columns correspond to the inferred classes.

Confusion Matrix	*Tall Building*	*Inside City*	*Street*	*Highway*	*Coast*	*Open Country*	*Mountain*	*Forest*
Tall Building	**0.88**	0.07	0.00	0.01	0.01	0.00	0.01	0.02
Inside City	0.07	**0.87**	0.04	0.02	0.00	0.00	0.00	0.00
Street	0.03	0.04	**0.89**	0.02	0.00	0.01	0.01	0.01
Highway	0.00	0.03	0.02	**0.82**	0.07	0.03	0.03	0.00
Coast	0.00	0.00	0.00	0.02	**0.85**	0.11	0.01	0.01
Open Country	0.00	0.00	0.01	0.02	0.15	**0.74**	0.05	0.03
Mountain	0.01	0.00	0.00	0.01	0.02	0.05	**0.85**	0.06
Forest	0.00	0.00	0.00	0.00	0.00	0.02	0.05	**0.93**

Table 2.4: Results obtained exploiting the GIST representation on the 8 Scene Context Dataset [112]. Columns correspond to the inferred classes.

Confusion Matrix	*Tall Building*	*Inside City*	*Street*	*Highway*	*Coast*	*Open Country*	*Mountain*	*Forest*
Tall Building	**0.83**	0.01	0.03	0.00	0.00	0.13	0.00	0.00
Inside City	0.00	**0.94**	0.00	0.00	0.05	0.01	0.00	0.01
Street	0.07	0.00	**0.82**	0.03	0.03	0.03	0.02	0.00
Highway	0.02	0.01	0.01	**0.84**	0.00	0.01	0.04	0.08
Coast	0.01	0.05	0.01	0.00	**0.86**	0.05	0.00	0.02
Open Country	0.14	0.04	0.02	0.00	0.05	**0.73**	0.01	0.00
Mountain	0.00	0.01	0.03	0.05	0.01	0.02	**0.87**	0.02
Forest	0.00	0.01	0.00	0.08	0.02	0.00	0.00	**0.88**

ments. The notion of *Openness* is related to the open vs closed-enclosed environment, scenes with horizon vs no horizon, a vast or empty space vs a full, filled-in space [112]. A closed scene is a scene with small perceived depth, whereas an open scene is a scene with a big perceived depth. Information about *Naturalness* and/or *Openness* of the scene can be very useful in setting parameters of the algorithms involved in the image generation pipeline [26].

For the *Naturalness* experiment we have split the 8 scene dataset as in [56, 112] by considering the classes *Coast*, *Open Country*, *Mountain* and *Forest* as *Natural* environments, whereas the classes *Tall Building*, *Inside City*, *Street* and *Highway* as belonging to the *Man-Made* environments. For the Openness experiment, the images belonging to the classes *Coast*, *Open Country*, *Street* and *Highway* have been considered as *Open* scenes, whereas the images of the classes *Forest*, *Mountain*, *Tall Building* and *Inside City* have been considered as *Closed* scenes. The results obtained employing the proposed representation (F) are reported in Table 2.5 and 2.6. The obtained results closely match the performances of other state-of-the-art methods [10, 56, 140] by employing less computational resources.

Table 2.5: *Natural* vs *Man-Made* classification performances of the proposed image representation (F). Columns correspond to the inferred classes.

	Natural	**Man-Made**
Natural	**97.88**	2.12
Man-Made	4.75	**95.25**

Table 2.6: *Open* vs *Closed* classification performances considering the proposed image representation (F). Columns correspond to the inferred classes.

	Open	**Closed**
Open	**94.17**	5.83
Closed	4.63	**95.37**

Table 2.7: Results obtained by the proposed representation (F) on four classes usually used in the auto-scene mode of consumer digital cameras.

	Landscape	**Man-Made Outdoor**	**Portrait**	**Snow**
Landscape	**87.76**	1.22	0.61	10.41
Man-Made Outdoor	3.78	**91.33**	2.22	2.67
Portrait	1.02	1.84	**94.29**	2.86
Snow	9.62	1.13	3.02	**86.23**

Table 2.8: Results obtained by GIST [112] on four classes usually used in the auto-scene mode of consumer digital cameras.

	Landscape	**Man-Made Outdoor**	**Portrait**	**Snow**
Landscape	**84.69**	3.27	0.20	11.84
Man-Made Outdoor	4.44	**87.78**	2.44	5.33
Portrait	0.41	3.47	**91.84**	4.29
Snow	11.70	3.40	4.34	**80.57**

Finally, we have considered the problem of recognizing four scene context usually available in the auto-scene mode of digital consumer cameras: *Landscape*, *Man-Made Outdoor*, *Portrait*, *Snow*. To this purpose we have collected 2000 colour images (i.e., 500 per class) with resolution 640×480 pixels from Flickr . This dataset has been used to perform a comparative test of the proposed image representation with configuration (F) with respect to the GIST descriptor. The results are reported on Table 2.7 and 2.8. The proposed image representation obtained an average accuracy of 89.80%, whereas GIST achieved 86.07%.

Table 2.9: Results obtained on the 15 Scene Dataset [91]

Bags of Textons with spatial hierarchy [10]	79.43%
Proposed representation (D)	78.45%
GIST [112]	73.25%

2.4.2 Proposed Representation vs Bags of Textons on Spatial Hierarchy

Since the proposed scene context representation works exploiting information collected on spatial hierarchy, we have compared it with respect to the method presented in [10], where Bags of Textons are collected for each region in the spatial hierarchy to represent the images for scene classification purposes. For this comparison we have considered the 15 Scene Classes Dataset introduced in [91]. The dataset is composed by 4485 images of the following fifteen categories: *highway*, *inside of cities*, *tall buildings*, *streets*, *forest*, *coast*, *mountain*, *open country*, *suburb residence*, *bedroom*, *kitchen*, *living room*, *office*, *industrial* and *store*. Since a subset of the images of the dataset does not have color information, this test has been performed taking into account only the Y channel and using the scene descriptor with configuration (D) reported in Table 2.2. The results obtained on this dataset are reported in Table 2.9. The average per class accuracy achieved by the proposed approach is 78.45%, whereas the method which exploit textons distributions on spatial hierarchy [10] obtained an accuracy of 79.43%. Both representations outperform the GIST one, which obtains 73.25% of accuracy on this dataset. Although the results are slightly in favour for the method proposed in [10] (of less than 1%), one should not forget that the proposed method is suitable for an implementation on the image generation pipeline of single sensor devices, whereas the method in [10] requires extra memory to store textons vocabularies (i.e., hardware costs for industry) as well as a bigger computational overhead to represent the image to be classified (e.g., convolution with bank of filters, Textons distributions for every sub-regions, ecc.).

We have performed one more test to assess the ability of the proposed representation in discriminating among *Indoor* vs *Outdoor* scenes. This prior can be very useful for autofocus, auto-exposure and white balance algorithms. To this aim we have divided the images

Table 2.10: *Indoor* vs *Outdoor* classification performances considering the proposed image representation (D). Columns correspond to the inferred classes.

	Indoor	**Outdoor**
Indoor	**89.75**	10.25
Outdoor	3.86	**96.14**

of the 15 Scene Classes Dataset as indoor or outdoor images. The classification results are reported in Table 2.10. Again the results confirm that the proposed representation can be employed to distinguish classes of scenes at superordinate level of description [112].

2.4.3 Instant Scene Context Classification on Mobile Device

The experiments presented in Sections 2.4.1 and 2.4.2 have been performed on representative datasets used as benchmark in the literature. For those tests the scene context representation has been obtained directly by extracting the DCT information from the compressed domain (JPEG format). The main contribution of this work is related to the possibility to obtain a signature for the scene context directly into the image generation pipeline of a mobile platform, taking into account the architecture presented in Section 2.3. To this aim we have implemented the proposed architecture on a Nokia N900 smartphone [22]. This mobile platform has been chosen because it has less computational power of the other smartphones (i.e., the scene context classification engine should able to classify in real-time independently of the computational power of the device). Moreover, with the chosen mobile platform, the FCam API can be employed to work within the Image Generation Pipeline of the device [5,64]. This allows to effectively build the proposed architecture and test it with real settings. Although the limited resources of the hand-held device, the implemented system works in real-time as demonstrated by the video available at the following URL:
`http://iplab.dmi.unict.it/SceneClassificationMobile.wmv`.
For the implemented system, we have used a SVM model learned offline on the 8 Scene Context Dataset (see Section 2.4.1) and the configuration (F) for the image representation (see Table 2.2). The scene context representation is computed on the fly during the generation of the image to be displayed in the viewfinder. The implemented architecture can also

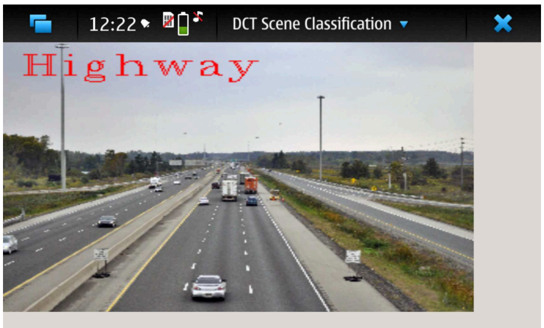


Figure 2-9: Example scene context classification of the system implemented on the Nokia N900.

perform classification of images already stored in the mobile (Fig. 2-9).

The proposed scene context classifier has been also tested on a NovaThor U9500 with Android OS. The board mounts a 1 GHz Dual-core ARM Cortex-A9 CPU. The computational time performances have been evaluated by considering the average latencies of the different scene classification blocks on a set of QVGA images. We have measured the computational time of all the steps involved in the scene classification: DCT computation, image representation with configuration (F) (see Table 2.2) and the SVM classification. The DCT computation required 15.6 ms on the average (this value could be disregarded when DCT coefficients are directly provided by the integrated JPEG encoder or by working directly on compressed domain). The overally computational time to build the image signature with configuration (F) (i.e., the one with spatial hierarchy and all the three image channels of the image) was only 0.3 ms. Finally, the SVM classification required 117.4 ms. This test confirmed that the proposed image signature can be computed in realtime within a mobile platform.

Chapter 3

Semantic Segmentation

In this chapter we present a new approach for generating class-specific image segmentation. Two novel features that use quantized DCT data are introduced in the Semantic Texton Forest system [128] with the aim to combined colour and texture information. The combination of multiple features in a segmentation system is not a straightforward process. The proposed system is designed to exploit complementary features in a computationally efficient manner. Our DCT features describe complex textures represented in the frequency domain and not just textures obtained using simple differences between intensity of pixels. Current approaches usually computes texture features using filter-bank responses that drastically increases the execution time of the segmentation system. Our approach, instead, uses a limited amount of resources. The proposed method has been tested on the popular CAMVID database [34,35]. Comparison with respect to the recent approaches available in this field shows improvement in term of semantic classifications. The rest of this chapter is organized as follows: section 3.1 describe the random forest algorithm and how to integrate the novel features in the STF system. Section 3.2 presents the pipeline to segment the image whereas section 3.3 introduces the extraction pipelines for each of the novel features. Section 3.4 describe the experimental settings and the results.

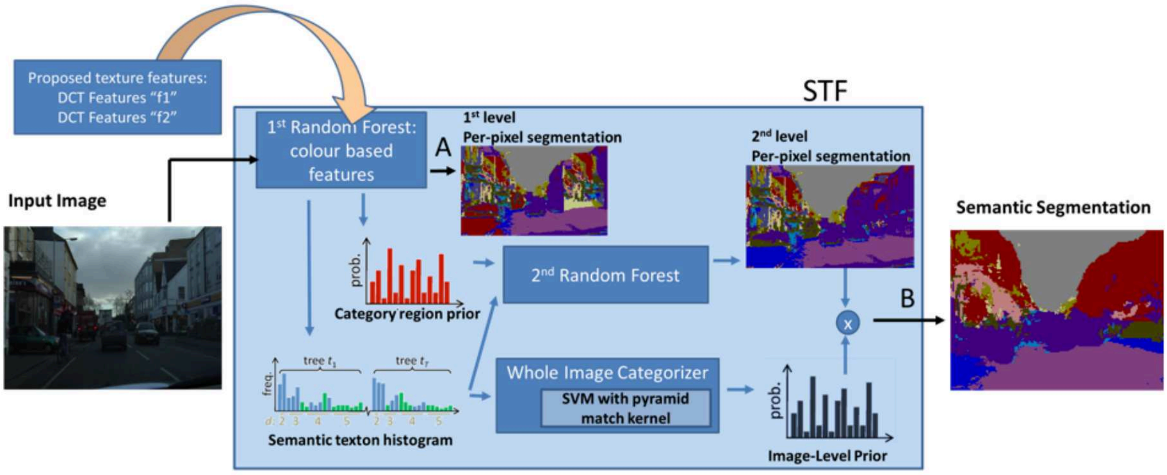


Figure 3-1: Integration of texture features in the STF system

3.1 Random Forests and Semantic Texton Forest

Before presenting our approach, we will briefly review the randomized decision forest. Random forests are an ensemble of separately trained binary decision trees. These decision trees are trained to solve a problem together and the results are predicted combining all the partial results obtained by each tree. This process leads to a significantly better generalization and avoids over fitting to the data. Maximizing the information gain and minimizing the information entropy are the goals of the training to optimally separate the data points for classification problems or to predict a continuous variable. The decision tree concept was described for the first time in [33] and sub sequentially more and more applications used an ensemble of randomly trained decision trees for machine learning. [127] used an ensemble to solve machine-learning tasks with the Boosting algorithm. As result of this work, the authors found out that an ensemble of (weak) learners achieved a significantly better generalization. More complex applications were implemented in [6] for a shape classification system, in [72] for an automatic handwriting recognition and in [44] for a medical imaging applications. A Random Forests can solve divers problems like the prediction of class for specific data, the predicting of a continuous variable, the learning of a probability density function or the learning of manifolds. The Random Forest uses weak classifiers to solve its tasks. A weak classifier is specialized on a sub problem and significantly faster compared to a strong classifier, which is usually designed to tackle complex problems. Every Random Forest can be described by the number of the trees used T, the maximum depth D and the type of weak learner model that contributes to the corresponding energy function. The STF model is a complex system that ensembles 2 randomized decision forests and an image categorization block. The randomized decision forests obtains semantic segmentation acting directly on image pixels, and therefore do not need the expensive computation of filter-bank responses or local descriptors. They are extremely fast to both train and test. Specifically, the first randomized decision forest in the STF uses only simple pixel comparisons on local image patches of size dxd pixels. The split functions f_1 in this forest can directly take the value $p(x, y, b)$ at pixel location (x, y) in the colour channel b or some other function defined on two different location $p_1(x_1, y_1, b_1)$ and $p_2(x_2, y_2, b_2)$ relieved within the square patches dxd. Given for each pixel i the leaf nodes $L_i = (l_1, ..., l_T)_i$ and inferred class distribution $P(c|L_i)$, one can compute over an image region r a non-normalized histogram $H_r(n)$ that concatenates the occurrences of tree nodes n across the different T trees, and a prior over the region given by the average class distribution $P(c|r) = \sum_{i \in r} P(c|L_i)$ (see Fig. 3-1). The second randomized decision forest in the STF uses the category region prior $P(c|r)$ and the semantic texton histogram $H_r(n)$ to achieves efficient and accurate segmentation. Specifically, the split node functions f_2 of the second forest evaluate either the numbers $H_{r+1}(n = n')$ of a generic semantic textons n' or the probability $P(c = c'|r + i)$ within a rectangle r translated relative to the pixel i that we want to classify. The categorization module determine finally the image categories to which an image belongs. This categorization is obtained by exploiting again the semantic texton histogram $H_r(n)$ computed on the whole image using a non-linear support vector machine (SVM) with a pyramid match kernel. The STF runs separately the categorization and the segmentation steps, producing an image-level prior (ILP) distribution P(c) and a per-pixel segmentation distribution P(c|i) respectively. The ILP is used to emphasize the likely categories and discourage unlikely categories:

$$P'(c|i) = P(c|i)P(c)^a \text{ using parameter } a \text{ to soften the prior} \tag{3.1}$$

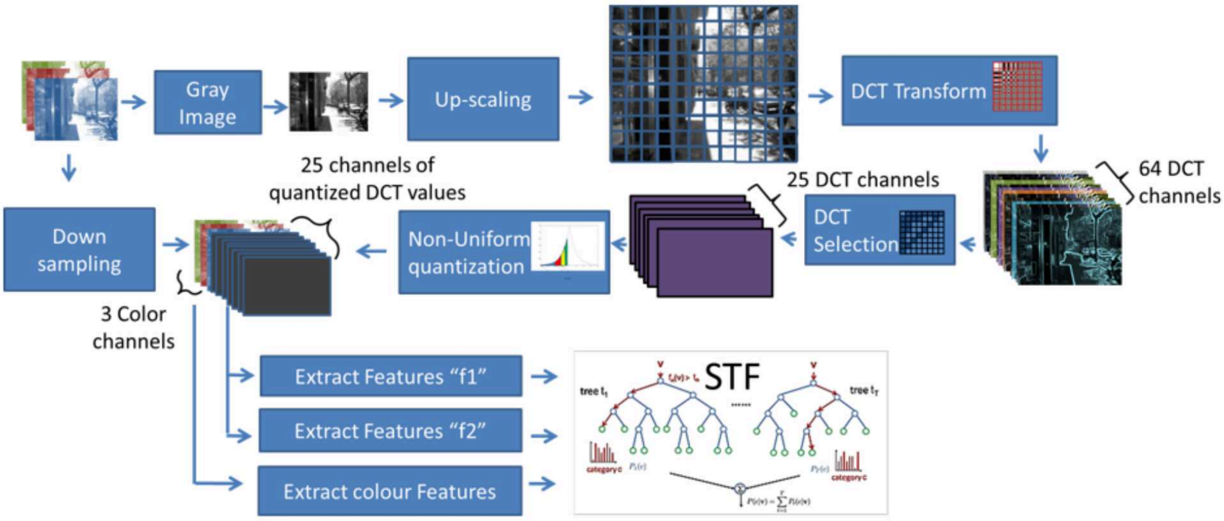
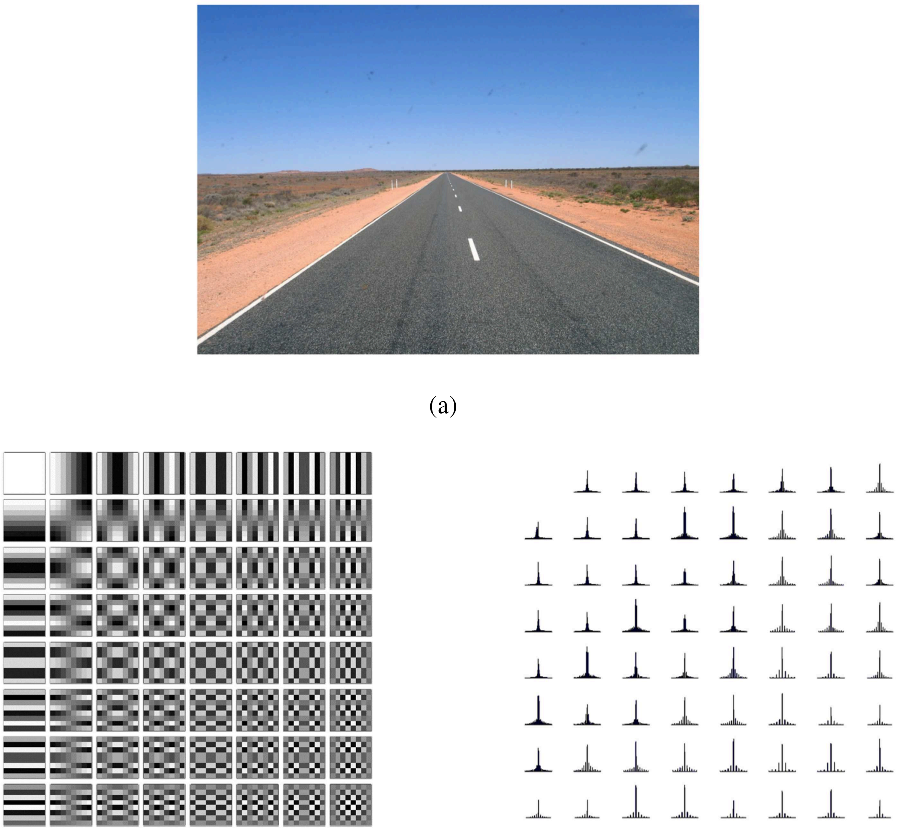


Figure 3-2: Pipeline to integrate the novel DCT features in the STF system.

As previous mentioned, our approach combines texture and colour clues within a STF (see Fig. 3-1). Adding the texture features in the first random forest allow us either to catch the semantic segmentation output after performing entirely the STF system (point B in the Fig. 3-1) or after perform just the first random forest (point A in the Fig. 3-1). The last solution is preferred for real time applications, when the execution time is more important respect to the accuracy. In the experimental section 3.4, we show that including the proposed DCT features increase the accuracy in both the semantic segmentation output 3-13.

3.2 Proposed approach

The workflow of our method is shown in fig 3-2. Each image is first converted into a grayscale channel and then up-scaled by a variable factor. Sub sequentially, the DCT transformation is applied and the most discriminative DCT coefficients are selected. The DCT coefficients are suitably quantized and combined with a subsampled version of the colour data to generate input features to the STF system. Next sections explain the functionality of each of the introduced block.



(a)

(b)

(c)

Figure 3-3: Laplacian distributions of DCT coefficients for natural images. Top: image under consideration; bottom-left: the 64 basis related to the 8×8 DCT transformation; bottom-right: the different DCT distributions related to the 64 DCT basis reported at bottom-left, obtained considering the image at top.

3.2.1 DCT Transform e DCT Selection

One of the most popular standard for lossy compression of images is JPEG. JPEG is an hardware/software codec engine virtually present in all the consumer devices such as digital cameras, smartphones etc. Moreover, the great majority of the images on Internet are stored in JPEG format. Image segmentation features that can be extracted directly in the JPEG compressed domain are hence desirable. The JPEG algorithm, divide the image into non-overlapping blocks typically 8x8 pixels in size and each block is transformed using the discrete cosine transform (DCT) followed by quantization and entropy coding. The DCT has been extensively studied and hence there is a very good understanding of the statistical distributions of the DCT coefficients and their quantization. Different statistical models for AC coefficients were proposed including Gaussian [116], Cauchy, generalize Gaussian and sum of Gaussian distributions [51, 53, 107, 131, 160]. The knowledge of the statistical distribution of the DCT coefficient is useful in quantizer design and noise mitigation for image enhancement. In our model we assume that the distribution of the AC coefficients resemble the Laplacian distribution. (See Fig. 3-3(c)). This guess has been demonstrated through a rigorous mathematical analysis in [89, 90]. The probability density function of a Laplacian distribution can be written as:

$$F(x \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \tag{3.2}$$

where μ and b are the parameters of the Laplacian model. Given N independent and identically distributed samples $x_1, x_2, ..., x_N,$ (i.e., the DCT coefficient related a specific frequency in our case) an estimator $\hat{\mu}$ of μ is the sample median and the maximum likelihood estimator of the slope b is:

$$b = \frac{1}{N}\sum_{i=1}^{N} |x_i - \mu| \tag{3.3}$$

In a recent work [22] describes how to use these parameters to classify the scene in real time. In section 3.2.2, instead, we show how to use the Laplacian model to quantize properly the DCT coefficient and use them to extract texture features for the image segmentation problem. As shown in [33] the most prominent patterns composing images are edges. Some of the DCT basis are related to the reconstruction of edges of an 8x8 image block (i.e., first row and first column of Fig. 3-3(c)), whereas the other are more related to the reconstruction of the textured blocks. Moreover, high frequencies are usually affected by noise and could be not useful for segment the image. For this reason, we have performed an analysis to understand which of the AC DCT basis really can contribute in our pipeline. One more motivation to look only for the most important frequencies is that we can reducing the complexity of the overall system. To select the most important frequencies we used a greedy fashion approach. Our analysis suggested that a good compromise between segmentation accuracy and computational complexity (i.e., the number of AC DCT frequencies to be included in the pipeline to fit with required computational time and memory

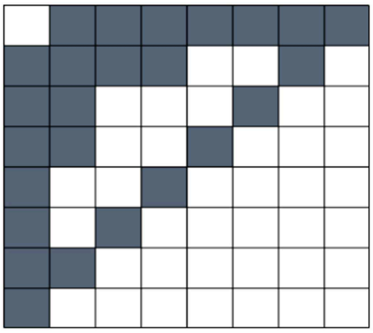


Figure 3-4: Schema used to select the DCT frequencies

resources) is the one which considers the AC DCT components related the DCT basis of Fig. 3-4. According this schema only 25 frequencies out of 64 are selected. We will refer to this set of frequencies as *selDCTfreqs* and the related number as $N_{selFreq}$.

3.2.2 Quantization

Two important observations regarding the DCT data should be taken into account when these data are used inside a pipeline.

- 1st observation: It is a common knowledge that in the real world, the human vision is more sensitive to some frequencies rather than others.
- 2nd observation: The distribution of the DCT data is not a normal distribution (as described in the previous section the DCT coefficient distributions for natural images indicates that they resemble to the Laplacian distributions).

These observations convey the fact that before using the DCT data, they need to be properly processed. In our process, the first condition is obtained replacing the uniform random

Table 3.1: Quantization matrix, specified in the original JPEG process

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	54	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

function used to select the features in each node of the 1st random forest (see Fig. 3-1), with a weighted selection function which steers the learning process towards using more frequently the DCT coefficient that are more important. So, each weight in the quantization table 3.1 is transformed in a probability selection value according to the following function:

$$p_i = \begin{cases} \frac{1}{q_i \sum_{j=1}^{N_{selFreq}} q_j} & \text{if i} \in \text{selDCTfreqs} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Where q_i are the quantization values, j $\in selDCTfreqs$ and $N_{selFreq}$ is the number of selected DCT frequencies. The standard quantization table 3.1 is then transformed in a probability table that we refer with the symbol pt (see table 3.2). This table speed up the learning increasing the probability to discover good features that maximize the information gain of the data, in each node of the 1st random forest of the STF system. In order to consider the second observation, we proposed a quantization step that is capable to generate more centroids in the DCT space where the data distribution is denser (all the value that are near to the center of the Laplacian) and less in the areas where only a few DCT data fall in. This process produces centroids that are conforming to the natural distribution of the considered DCT data. Classical k-means works well for data having uniform distribution. In the case of non-uniform distribution the k-means devotes most of its centres in a marginal area where only few elements occurs, and the final coding suffers (see Fig. 3-5). For this reason, a non-uniformity clustering is essential to quantize the DCT data. To obtain the quantization with the aforementioned non-uniform property, we propose an analytic solution. An uncompressed training database of images is used to obtain the two Laplacian parameters (median and slop) of each DCT coefficient. The cluster centroids are

Table 3.2: Probability table pt obtained from the standard Jpeg quantization table

0	0.078	0.086	0.054	0.036	0.022	0.017	0.014
0.072	0.072	0.061	0.045	0	0	0.014	0
0.061	0.066	0	0	0	0.015	0	0
0.061	0.051	0	0	0.017	0	0	0
0.0]48	0	0	0.015	0	0	0	0
0.036	0	0.016	0	0	0	0	0
0.018	0.013	0	0	0	0	0	0
0.012	0	0	0	0	0	0	0

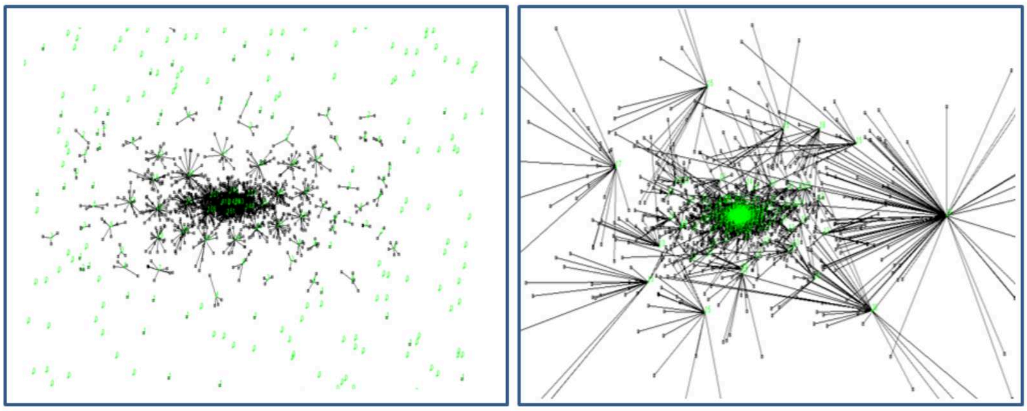


Figure 3-5: Difference between uniform k-mean and non-uniform k-means: the two images represent a 2D space where the samples are represented in black and the obtained centroids in green. In the left, the k-means produce centroids uniformly distributed, in the right a non-uniform quantization produces cluster having approximately the same number of sample.

then computed performing an integration on the area of each Laplacian model. The points that divide this area in k equal spaces (starting from the minimum of this distribution and arriving to the maximum) are the proposed quantization points. Fig. 3-6 shows an example for one of the DCT coefficient. This process is repeated for all the DCT coefficients, separately and it produces a table $tTex$ with k x $N_{selFreq}$ entries. In each column of this table, the values are arranged in an ascending order to be used later as stopping criteria during the clustering process. When the segmentation process is performed, each DCT coefficient extracted from the image, is converted in a quantized value and saved in a specific channel according to the corresponding DCT index. In the experimental section we use clustering with 8, 16, 32 and 64 centroids. Table 3.8 shows that increasing the number of the clusters will not provide substantial improvement to the system. For this reason, the clustering with 8 centroids is the one that we propose in the final configuration. With 8 cluster for each frequency we can represent 8x$N_{selFreq}$ different textons (in our case with 25 frequencies selected there are 200 textons). Furthermore, with this configuration, each quantized DCT data, occurred in the DCT layers, can be saved in memory, employing only 3 bits.

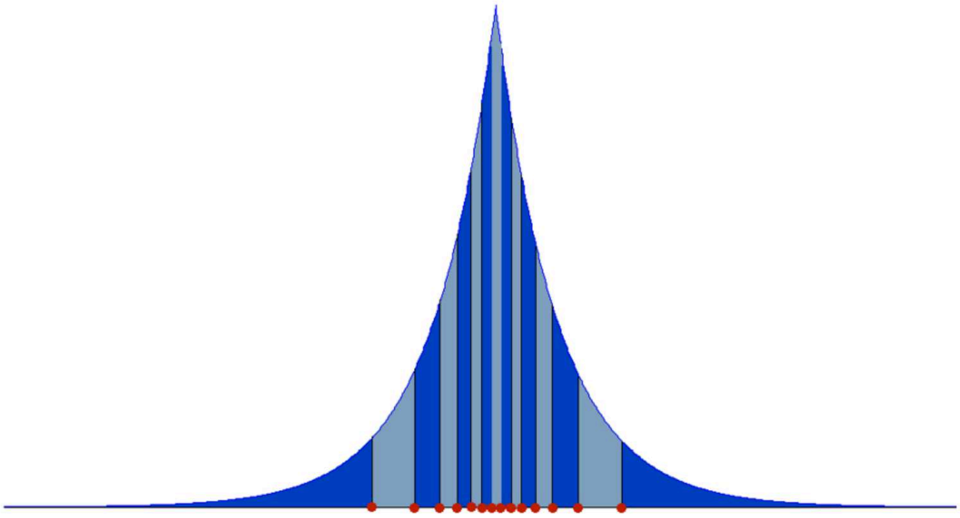


Figure 3-6: Laplacian model representing one of the $selDCTfreqs$ frequencies. The red points represent the clusters obtained using the proposed analytical process for quantize this DCT frequency.

3.2.3 Up-Scaling and down-sampling

In the original STF approach [128] only 3 colour channels are used to generate the features in each node of the 1st random forest. In the proposed approach, other $N_{selFreq}$ channels containing quantized DCT data are added to the system. The DCT data are obtained using a block base process, which means that this information is not pixel specific. In order to reach the same size of the colour layers, we enlarge the input image before the DCT transformation is applied (see Fig. 3-2). Furthermore, one should also consider that generally, the semantic segmentation is not required for each pixel and hence a down sampling step is also considered in another section of the pipeline. In order to have consistent size between the texture layers and the colour layers, the product between the up-scaling and down sampling factor must be equal to the size of the DCT transformation block (that in our solution is 8).

$$up\_scaling_fact * down\_sampling\_fact = black\_size \tag{3.5}$$

Compatible colour and texture channels data are now available to generating features for the STF system.

3.3 Proposed features

After quantizing the DCT data, we are ready to define our semantic segmentation features. In our system, we propose two new features. The first one, that we call feature $f1$, has the purpose to compute statistics of frequencies on a specific region of the image with respect to the pixel i that we want classify. The aim of the proposed statistics is to recognize objects occurred in the images. For example, the system will be capable to distinguish an area with many trees (that contains high distribution of the DCT coefficients related to the high frequencies) from an area representing a road (that instead contains a low distribution of the same DCT coefficients). The features $f1$, is defined as a triplet $[r(x, y, h, w), t, s]$ of an image region, r, on the DCT layer t, using a statistics operator with parameter s that represent a random quantized value. The region r is defined in coordinates relative to the pixel i which we need to classify. For efficiency, we only investigate rectangular regions and for simplicity, a set R of candidate rectangles are chosen at random, such that their top-left and bottom-right corners lie within a fixed bounding box that we define as B_1. Fine

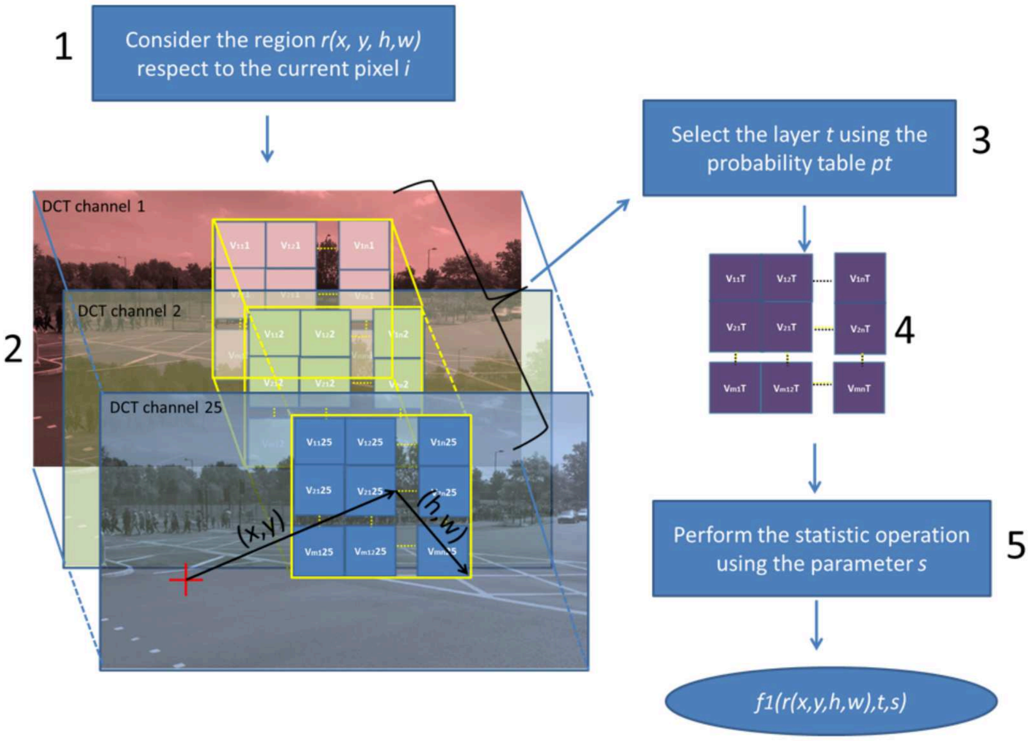

1
Consider the region r(x, y, h,w) respect to the current pixel i
2
DCT channel 1
DCT channel 2
DCT channel 25
(x,y)
(h,w)
3
Select the layer t using the probability table pt
4
5
Perform the statistic operation using the parameter s
f1(r(x,y,h,w),t,s)

Figure 3-7: Extraction pipeline for feature $f1$

details of the extract process used to obtain this feature is explained in Fig. 3-7. One of the $selDCTfreqs$ available is first selected using the probability table pt. Each layer has a different probability of selection according the table 3.2. After that, only the region r is taken into account and a statistical measurement is performed on this region. Fixed the value s as one of the quantized value (selected randomly when the features is generate) and the region r, we propose the following 3 types of statistics evaluation:

$$stat1\,(r, s) = \frac{\sum_{scan \in r} |scan - s|}{||r||} \tag{3.6}$$

$$stat2\,(r, s) = \frac{\sum_{scan \in r} (scan - s)^2}{||r||} \tag{3.7}$$

$$stat3\,(r, s) = \frac{\sum_{scan \in r} scan == s}{||r||} \tag{3.8}$$

where $||r||$ represent the area of the region r. The performance of each statistic measures are included in the table 3.8. Although these statistic measures can be efficiently computed over a whole image exploiting the integral histogram [147], its use is not always granted. Using the integral histogram is a system design choice that depends on the industry manufacturer constraints. If the available memory is enough to contain the integral histogram ($N_{selFreq}$ new layers of integer), the features $f1$ can be computed in constant time exploiting this integral. Otherwise, a specific number of operations are computed each time that the feature $f1$ is required in the system. It is important know the precise number of operations required to compute $f1$ when the integral histogram is not available because this feature can drastically reduce the system performances. According to how this feature is defined, the bounding box of the region r have a maximum size B_1 and all the sizes are distributed uniformly in the range $0 - B_1$. According to this fact, the formula that describe the average number of operations required to compute $f1$ is:

$$\frac{\sum_{i}^{B_1} i^2 * sOp}{B_1} = \frac{(B_1 + 1)(2 * B_1 + 1) * sOp}{6} \tag{3.9}$$

where sOp is a value that depend on the statistic measures and specifically is 2 for stat1, 3 for stat2 and 1 for stat3. Table 3.9 summarizes the number of required operations when

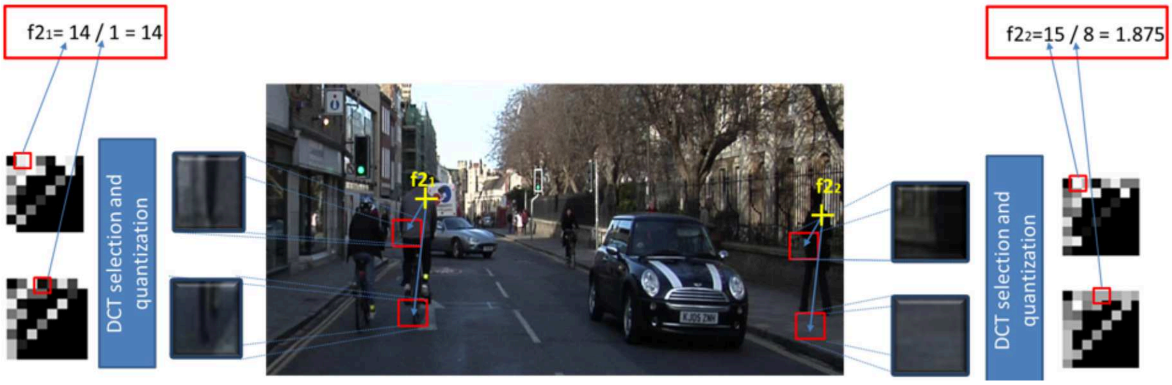

f2$_1$= 14 / 1 = 14
DCT selection and quantization
f2$_1$
f2$_2$
DCT selection and quantization
f2$_2$=15 / 8 = 1.875

Figure 3-8: Example of features $f2$. The feature $f2$ is performed in two different points specified by yellow crosses. The points represent respectively a bicyclist and a pedestrian pixel class. Thanks to the vertical high frequencies correlated to the wheel under the human the features $f2$ is capable to classify properly the under considerate classes.

different statistics and different up-scaling factors Us are used. Table 3.9 is obtained using an input image of 640x480 pixels and running the system with the bounding box B_1 equal to width/3 (best value according to the table 3.9 in the experimental section). The second

Table 3.3: Number of operations required to compute $f1$ for an image of 640x480 pixels

	Us=4	Us=2	Us=1
Stat1	7692	1950	501
Stat2	11538	2924	751
Stat3	3846	974	250

proposed feature called feature $f2$, is designed to compare two generic points $P1$ and $P2$ with respect to the pixel i that we want to classify. Fine details of the extract process used to obtain this feature is explained in Fig. 3-9. The channel W and Z are respectively selected and the two value w and z are combined through some operations listed in the "Available Operations" of Fig. 3-9. These operations are first analysed one by one in the validation step described in the experimental section 3.4 and then the outperforming operations are selected in the final configuration. The first 4 rows of table 3.5 shows the classification results obtained by the system when each of the proposed features $f1$ and $f2$ are included in the STF system. Some tests use also a feature called "unary" that is obtained when the point $P1$ and $P2$ are the same and the selected DCT channels W and Z are equal. The next 6 rows of table 3.5 shows, instead, the results obtained using the different type of operations

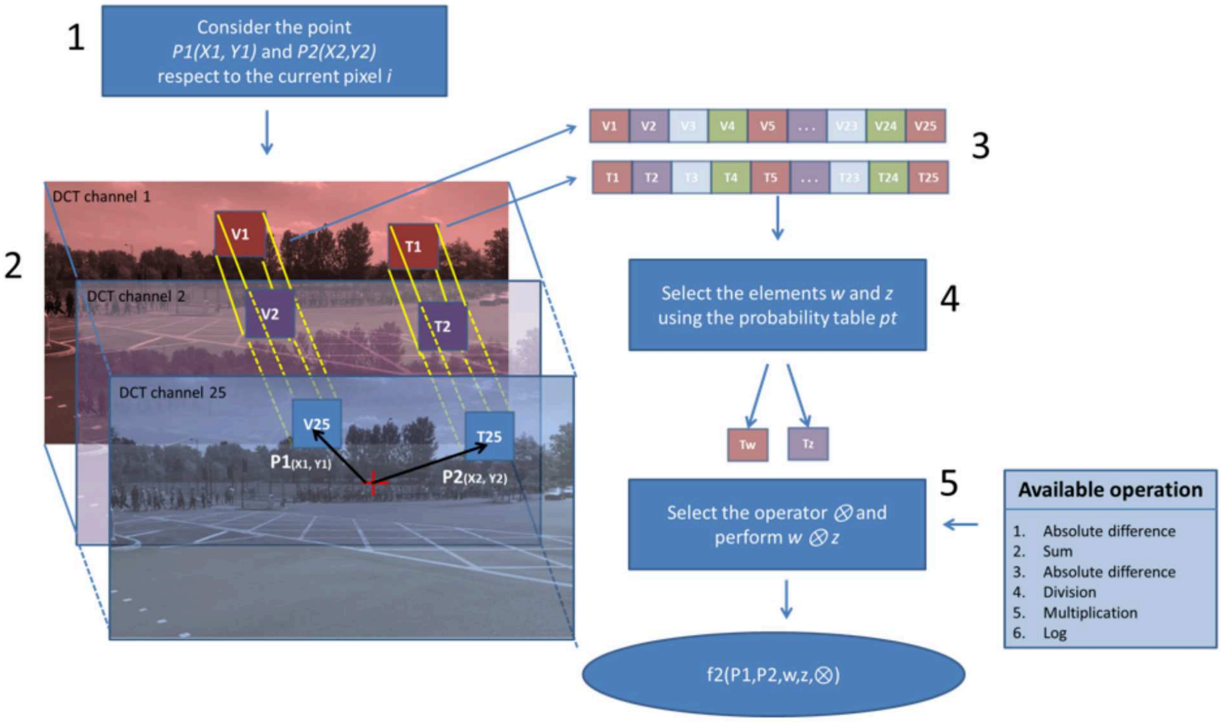

1
Consider the point P1(X1, Y1) and P2(X2,Y2) respect to the current pixel i
2
DCT channel 1
V1
T1
DCT channel 2
V2
T2
DCT channel 25
V25
T25
P1(X1, Y1)
P2(X2, Y2)
V1 V2 V3 V4 V5 ... V23 V24 V25
T1 T2 T3 T4 T5 ... T23 T24 T25
3
Select the elements w and z using the probability table pt
4
Tw
Tz
5
Select the operator ⊗ and perform w ⊗ z
Available operation
1. Absolute difference
2. Sum
3. Absolute difference
4. Division
5. Multiplication
6. Log
f2(P1,P2,w,z,⊗)

Figure 3-9: Extraction pipeline for feature $f2$

to compute feature $f2$.

The feature $f2$, despite being very simple, allows to recognize complex structures inside the image. For example, it is possible to combine the value of the high frequencies in the point $P1$ with the low frequencies in the point $P2$ or even combine the high frequencies in $P1$ and in $P2$ (see Fig. 3-8) that allow to describe sophisticated visual cues.

3.4 Experimental setting and results

3.4.1 Database

To analyse the proposed solution we have used the Cambridge-driving Labeled Video Database (CamVid) [35], [34]. This is a collection of videos captured on road driving scenes. It consists of more than 10 minutes of high quality (970 x 720), 30 Hz footage and is divided into four sequences. Three sequences were taken during daylight and one at dusk. A subset of 711 images is almost entirely annotated into 32 categories, but we used only the 11 object categories, forming a majority of the overall labelled pixels (89.16%) While most videos are filmed with fixed-position CCTV-style cameras, this data was captured from the perspective of a driving automobile. The driving scenario increases the number and heterogeneity of the observed object classes.

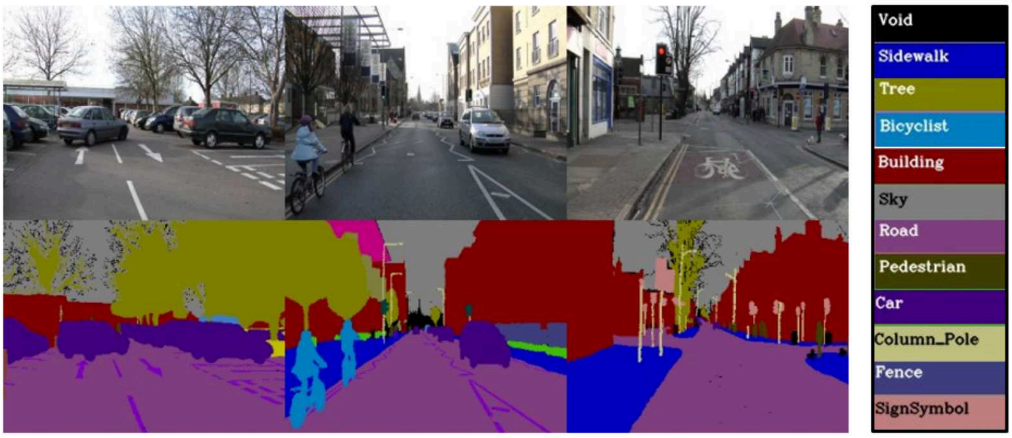


Figure 3-10: Camvid database

3.4.2 Parameters optimization

The parameters of the system are summarized in table 3.4. Our system has been extensively evaluated with the purpose to optimize these parameters. The database is split into 468 training images and 233 test images. A validation step is applied to obtain the best

Table 3.4: System parameters

Related to	Name	Description
Proposed features	M	Modality for different features setting
	Us	Up scaling factor used to enlarge the image
	S	Type of statistic used to generate the feature $f1$
	Qp	Number of quantization points used to quantize the Laplacian area
	B_1	Box size used to generate the feature $f1$
	B_2	Box size used to generate the feature $f2$
STF system	D_1	Depth for the 1st forest
	D_2	Depth for the 2nd forest
	N_1	Number of the features randomly chosen to generate the nodes in the 1st forest
	N_2	Number of the features randomly chosen to generate the nodes in the 2st forest

configuration for each parameter. In the test phase, the configuration set that obtains the best performance is used to training the final system. The semantic segmentation accuracy is computed by comparing the ground truth pixels to the inferred segmentation. We report per-class accuracies (the normalized diagonal of the pixel-wise confusion matrix), the class average accuracy, and the global segmentation accuracy.

Table 3.5 shows the results obtained when the novel features are introduced in the STF system and when different operations are used to compute the features $f2$. From the results, we can see that adding both the features $f1$ and $f2$ to the STF system, improves the classification performance. Specifically the best results are obtained when the "division" and the "sum" operations are considered to generate the feature $f2$.

Table 3.5: Results for different configuration of M

M	**Overall**	**MeanClass**
M1= STF	74.40	68.99
M2= STF & $f2$ unary	74.86	69.90
M3= STF & $f1$	75.55	70.46
M4= STF & $f1$ & $f2$ unary	74.84	70.42
M5= STF & $f1$& $f2$ unary & $f2$ |diff|	75.12	70.52
M6= STF & $f1$ & $f2$ unary & $f2$ sum	**75.51**	**71.01**
M7= STF & $f1$ & $f2$ unary & $f2$ diff	75.24	70.94
M8= STF & $f1$ & $f2$ unary & $f2$ div	**75.50**	**71.21**
M9= STF & $f1$ & $f2$ unary & $f2$ mol	75.19	70.75
M10= STF & $f1$ & $f2$ unary & $f2$ log	75.00	70.75

Tables 3.6(a),3.6(b) and 3.6(c) analyze the performance related to the forest parameters, specifically the depths D_1 and D_2 of the 2 random forests and the number of the features N_1 and N_2 randomly selected to generate each node.

Table 3.7 analyze the behaviours of the system when different up-scaling factor Us are used. The best results are obtained when the image is up-scaled by a factor of 4. To have good efficiency, it is recommended to use an up-scaling factor of 4 only when the integral histogram is used in the system, otherwise, reminding the computational analysis proposed in section 3.3 and according to the table 3.3, a good trade-off between performance and high efficiency is obtained using an up-scaling factor equal to 2.

Table 3.8 shows the system accuracy obtained using each of the proposed statistics when different number of clusters are computed for quantize the DCT data. The best results

Table 3.6: Analisys of the parameters related to the STF system

(a) Results for different values of N_1

N_1	**Overall**	**MeanClass**
400	75.12	70.52
600	75.59	70.83
800	75.16	70.90
1000	75.36	71.19

(b) Results for different values of N_2

N_2	**Overall**	**MeanClass**
400	75.12	70.52
600	74.96	71.39
800	75.29	71.38
1000	75.49	71.76

(c) Results for different values of D_1 & D_2

D_1 **&** D_2	**Overall**	**MeanClass**
D_1=12 & D_2=15	75.15	70.42
D_1=13 & D_2=14	74.19	70.25
D_1=13 & D_2=15	75.12	70.52
D_1=13 & D_2=16	75.68	70.71
D_1=14 & D_2=15	75.44	71.40

Table 3.7: Results for different values of Us

Us	**Overall**	**MeanClass**
4	75.32	71.91
2	75.12	70.52
1	74.46	69.43

are obtained when the statistic type 2 is selected. Moreover, only 8 clusters are enough to quantize the DCT data.

Table 3.8: Results for different values of Qp and for different type of statistic

	Qp=8		**Qp=16**		**Qp=32**		**Qp=64**		**Average**	
	Overall	**MeanClass**	**Overall**	**MeanClass**	**Overall**	**MeanClass**	**Overall**	**MeanClass**	**Overall**	**MeanClass**
Stat1	75.12	70.52	75.57	70.99	74.43	71.00	74.46	71.30	74.90	70.95
Stat2	75.48	71.21	75.41	71.11	74.89	71.63	74.90	71.06	**75.17**	**71.25**
Stat3	75.46	71.15	74.67	69.80	74.66	70.16	74.64	69.98	74.86	70.27
Average	**75.35**	**70.96**	75.22	70.63	74.66	70.93	74.67	70.78		

Table 3.9 and 3.10 show the performance obtained using different sizes for the bounding box B_1 and B_2. The best results are obtained when a bounding box equal to $wth * Us/(3 * DCTblocSize)$ pixels is used for the feature $f1$ and equal to $DCTblocSize * 15/Us$ pixels is used for the feature $f2$ (where wth is the width of the image, Us is the up-scaling factor and $DCTblocSize$ is the size of the DCT transformation block).

Table 3.9: Results for different values of B_1. W is $wth * Us/DCTblocSize$ where wth is the width of the image, Us is the up-scaling factor and DCTblocSize is the size of the DCT transformation block

B_1	**Overall**	**MeanClass**
w/2	75.08	70.92
w/3	**75.12**	**71.13**
w/4	75.00	70.52
w/5	74.78	70.12

Table 3.10: Results for different values of B_2. ResF is $DCTblocSize/Us$ where DCTblocSize is the size of the DCT transformation block and Us the up-scaling factor

B_2	**Overall**	**MeanClass**
resF*17	74.86	70.58
resF*15	**75.06**	**71.21**
resF*13	74.98	71.12

3.4.3 Experimental results

Table 3.11 compares the results obtained by the state of the art approaches with our approach when the best configuration set is used. Instead, Table 3.12, shows the confusion matrix obtained by our solution. Although on this database, our approach does not overcame the performance related to the overall accuracy, it shows a significant improvement in the per-class accuracy. The per-class measure applies equal importance to all 11 classes, despite the widely varying class prevalence, and is thus a much harder performance metric than the global accuracy measure, especially in this database that contain unbalanced classes.

Table 3.11: Comparison to state of the art on the CamVid dataset

	Classification for each class												
Approach	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	**Overall**	**Mean-class**
Proposed	49.16	**77.14**	93.51	80.84	**63.92**	88.05	**75.00**	**76.28**	**28.62**	88.54	**76.16**	76.35	**72.47**
Shot. [128]	44.83	75.31	93.39	80.53	59.96	88.99	71.15	70.40	27.90	**89.27**	73.89	74.90	70.51
Tighe [136]	83.10	73.50	94.60	78.10	48.00	96.00	58.60	32.80	5.30	71.20	45.90	**83.90**	62.50
Tighe [137]	**87.00**	67.10	96.90	62.70	30.10	95.90	14.70	17.90	1.70	70.00	19.40	83.30	51.20
Brostow [35]	46.20	61.90	89.70	68.60	42.90	89.50	53.60	46.60	0.70	60.50	22.50	69.10	53.00
Sturgess [132]	84.50	72.60	**97.50**	72.70	34.10	95.30	34.20	45.70	8.10	77.60	28.50	83.80	59.20
Zhang [162]	85.30	57.30	95.40	69.20	46.50	**98.50**	23.80	44.30	22.00	38.10	28.70	82.10	55.40
Floros [65]	80.40	76.10	96.10	**86.70**	20.40	95.10	47.10	47.30	8.30	79.10	19.50	83.20	59.60
Ladicky [87]	81.50	76.60	96.20	78.70	40.20	93.90	43.00	47.60	14.30	81.50	33.90	83.80	62.50

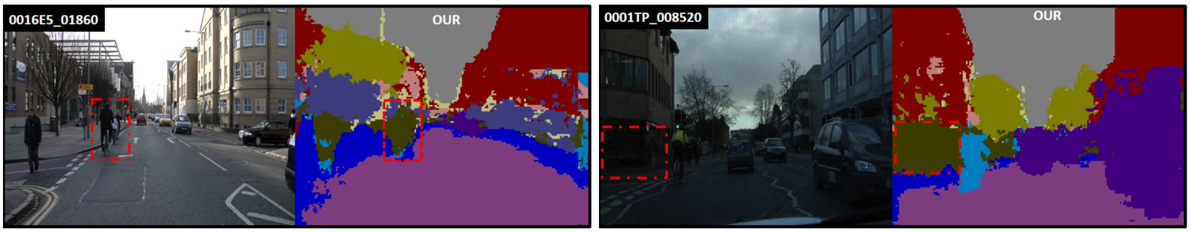
In Fig. 3-11 are shown some classification errors obtained when our approach is used.

In the first case a region containing a bicycle is confused with a pedestrian; instead in the second case an area belonging to a building is confused with the class pedestrian. The reason behind these errors can be explained as follows: in the first case, the long distance between the subject and the camera, does not allow to distinguish whether the high frequency under each person is relative to the wheel's bike or to the legs of the subjects. In the second case, the low brightness makes difficult even for a human to distinguish whether the textures in that area belong to a group of people or to the structure of the building.

Table 3.12: 11x11 Confusion Matric obtained on the CamVid

	Building	Tree	Sky	Car	Sign	Road	Pedestian	Fance	Pole	Sidewalk	Bycyclist
Building	**49.16**	4.76	1.49	5.24	11.54	0.12	10.56	6.79	6.80	2.66	0.87
Tree	3.18	**77.14**	2.91	1.49	3.29	0.07	2.67	6.81	1.64	0.69	0.11
Sky	0.45	4.34	**93.51**	0.06	0.23	0.00	0.00	0.01	1.40	0.00	0.00
Car	2.07	0.94	0.31	**80.84**	1.60	0.71	6.85	1.70	1.14	1.65	2.21
Sign	9.77	6.53	0.24	3.55	**63.92**	0.00	5.11	5.04	5.14	0.27	0.42
Road	0.01	0.01	0.00	2.19	0.01	**88.05**	0.33	0.16	0.22	8.17	0.85
Pedestian	1.57	0.32	0.00	5.19	2.21	0.22	**75.00**	4.51	3.30	2.94	4.73
Fance	0.68	3.10	0.00	3.36	0.76	0.35	8.31	**76.28**	1.82	5.07	0.27
Pole	9.71	11.45	4.06	2.33	9.96	0.49	16.66	9.65	**28.62**	5.98	1.10
Sidewalk	0.03	0.02	0.00	0.95	0.01	3.93	3.38	1.11	1.02	**88.54**	1.02
Bycyclist	0.11	0.37	0.00	3.66	0.50	1.03	13.05	2.68	1.13	1.30	**76.16**

In Fig. 3-12 is showed an example of visual segmentation outputs, obtained using our approach and the STF. In this case, our approach has the ability to segment more properly an area containing a bicyclist that with the STF approach is not recognized at all.



(a) (b)

Figure 3-11: Examples of classification error obtained using our approach. In 3-11(a) a region containing a bicycle is confused with a pedestrian; in 3-11(b) an area belonging to a building is confused with the class pedestrian.

In Fig. 3-13 are compared the computation time obtained by our approach and the STF during the two semantic segmentation levels. These tests are performed on a pc with

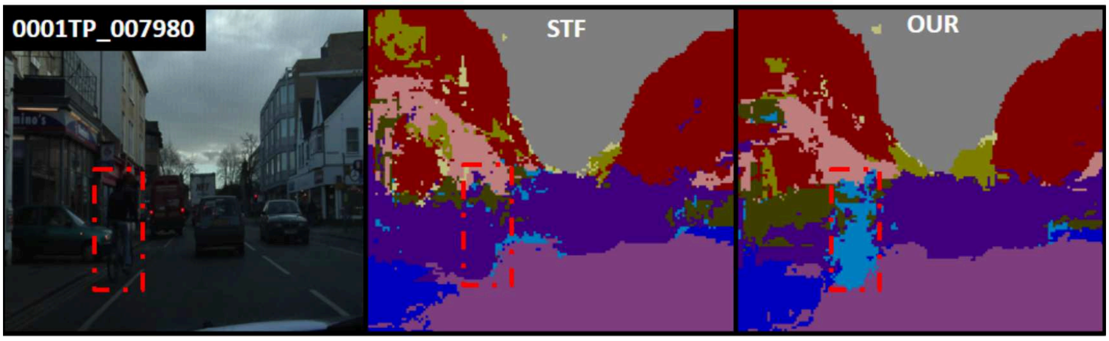


Figure 3-12: Example of visual segmentation improvement, obtained using our approach with respect to the STF.

a processor i73930k 3.20 Ghz (6 cores) and with 32 Gb of memory RAM. Both the approaches use the best parameters configuration (number of trees, depth, number of features analysed, bounding box, etc.). Moreover, features $f1$ are computed without the support of the integral image. As we can see from the image 3-13, the proposed features increases significantly the accuracy obtained on the first level (+8%) while are just slightly better on the second level (+2%). On the other hand, the complexity of our features have a negligible impact on the execution time. Hence, for real-time systems that cannot perform both the semantic segmentation levels, the introduction of our features is crucial to have a good classification improvement with a reduced amount of resources.

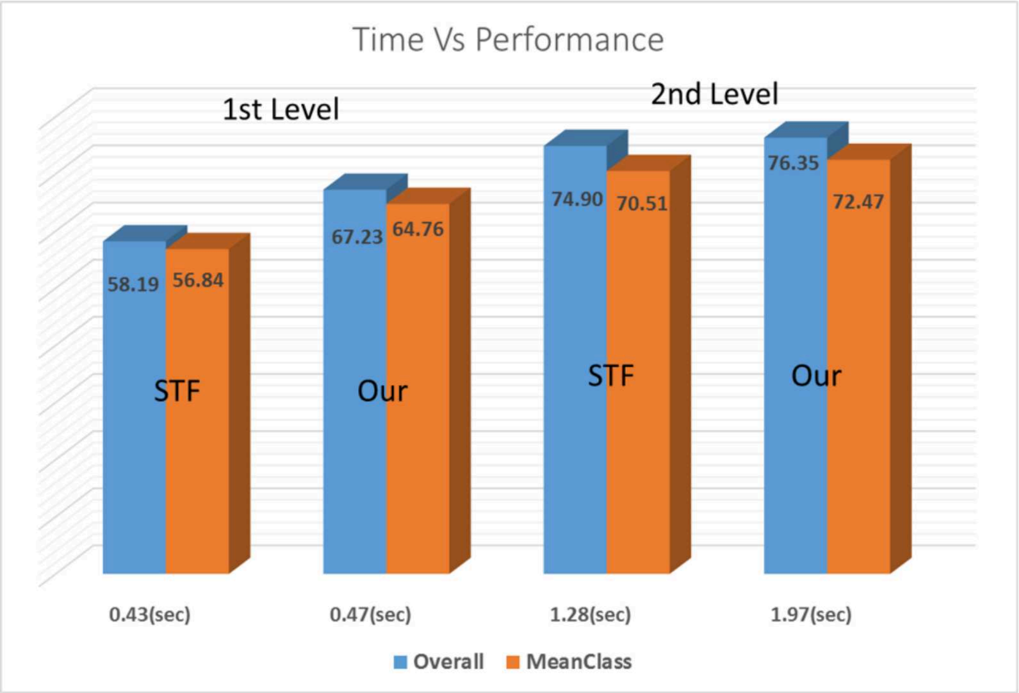


Figure 3-13: Computational time obtained by our approach and the STF during the two segmentation phases

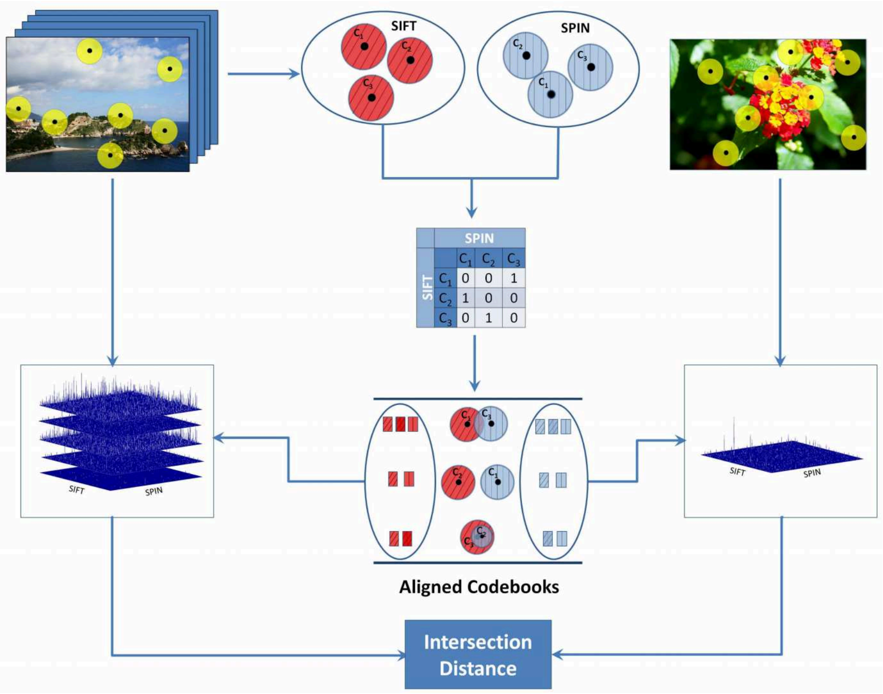
Chapter 4

Image Indexing

In this chapter we present a computer vision application for image indexing applied in the forensic context. We believe that the detection of near duplicate images in large databases, such as the ones of popular social networks, digital investigation archives, and surveillance systems, is an important task for a number of vision applications. In digital investigation, hashing techniques are commonly used to index large quantities of images belonging to different archives. In the last few years, different image hashing techniques based on the Bags of Visual Features paradigm appeared in literature for the detection of copies belonging to different archives. Recently, this paradigm has been augmented by using multiple descriptors (e.g., Bags of Visual Phrases) in order to exploit the coherence between different feature spaces. In our solution, we propose to further improve the Bags of Visual Phrases approach considering the coherence between feature spaces not only at the level of image representation, but also during the codebook generation phase outperforms the other state of the art approaches. The advantage of the proposed codebook alignment method is related to the enforcement of the coherence across multiple descriptors in order to capture different aspects of the considered local region (e.g., shape, texture, etc.) and hence reduce both, the visual word ambiguity and the quantization error in the visual codebook generation [73, 92]. The different aspects of a local regions are captured by the alignment during the codebook generation in the sense that the local regions falling in the intersection of two aligned clusters agree with respect to both descriptors, whereas the others agree just with one descriptor and not with the other. Taking into account such peculiarity, we split the clusters of each feature domain obtaining new codebook prototypes which consider the intersecting part of the aligned clusters, as well as the part which not intersect. Since, by using multiple descriptors there is an overhead in terms of storage of the representations of the images, and considering that image datasets are becoming more and more popular and huge (i.e., Facebook proceeds at a rate of about 22,000 uploads per minute), we also propose a method to compress the image representation by maintaining performances in terms of near duplicate image detection accuracy. The remainder of this chapter is organized as follows: Section 4.1 describes the proposed model. In Section 4.2 the method to compress the image descriptors is suggested. The dataset built for testing purposes is described in Section 4.3, whereas Section 4.4 details the experimental settings and reports the obtained results.

4.1 Proposed Model

Most of the image hashing techniques for near-duplicate image detection problems typically represent images through feature vectors encoding color, texture, and/or other visual cues such as corners, edges or local interest points [24,25,62,80,84,96,102,104,105,121]. These information are automatically extracted using several algorithms and then represented by many different local descriptors. Most of these techniques are based on the Bags of Visual Words paradigm (BoVW) [134] to build a global representation of the visual content within the images. The basic idea is to consider images as visual documents composed of repeatable and distinctive visual elements, which are comparable to the words in texts. Indeed, the BoVW originates in the text categorization community [125] where it was used to describe documents by how many words (belonging to a pre-built vocabulary) occur within them. Each word embracing a semantic meaning, has an inherent set of topics where it is used more often than others. To exploit this model in computer vision and multimedia, a vocabulary of distinctive patterns, usually called "visual words", is built through a clustering approach from a set of local descriptors [25, 62, 80, 84, 96, 105] extracted in correspondence of interest points [24, 102, 104, 121] which have been previously detected on images of a training database. A local descriptor encodes properties of the region sur-



		SPIN		
		C_1	C_2	C_3
SIFT	C_1	0	0	1
	C_2	1	0	0
	C_3	0	1	0

Figure 4-1: Proposed Bags of Visual Phrases with codebooks alignment. First a set of keypoints are extracted from a training dataset of images by using a local detector (Hessian-Laplace in our experiments). Each local keypoints is then described by two different descriptors (SIFT [96], SPIN [80] in our experiments) and clustering is performed separately in these two feature spaces. A similarity matrix between pairs of clusters belonging to the two partitions is obtained counting the number of elements (local image regions) they share. The Hungarian algorithm is then used to find the best assignment for the cluster correspondence problem which is encoded in the similarity matrix. The obtained cluster correspondences are then used to create two novel vocabularies where visual words are generated considering the centroids relative to both common and uncommon elements between aligned clusters. The training set images are then represented by using 2D histograms of co-occurrence of visual words related to the generated codebooks. When a query is performed on the training dataset, the test image is represented by using the codebooks obtained in the training phase. Test image representation is compared with those of the training images by using the intersection distance. Finally, the training image corresponding to the lowest distance is selected as the output of the query.

rounding the interest point in the image from which have been generated. Hence, the "visual words" obtained by clustering the training set of local descriptor are used to identify properties, structures and textures present in the images whose are finally described as an unordered set (a bag) of "visual words". Specifically, each image is represented as a normalized histogram whose bins correspond to "visual words" of the built codebook.

Since the bag of visual words description is compact, it is suitable to represent huge image databases. The proposed approach is built by taking into account the coherent phrase model introduced in [73], where the BoVW paradigm has been augmented by using multiple descriptors (called Bags of Visual Phrases Model) to exploit the coherence between different feature spaces (i.e., local descriptors) in which the local image regions corresponding to interest points are described. To further improve the Bags of Visual Phrases model, we propose to exploit the coherence between feature spaces (i.e., local descriptors) not only in the image representation step (e.g., using a two dimensional distribution of co-occurrence of visual words of codebooks corresponding to two different feature spaces), 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, which leads to a more distinctive image representation. Differently than Hu et al. [73], we do not obtain the final codebooks corresponding to the different feature spaces separately, but we generate the final codebooks taking into account the correspondence of the clusters of the involved spaces of descriptors to further enforce feature correspondence. Specifically, the partitions obtained through the clustering procedure on each descriptor space are further analyzed with respect to the involved local regions in order to find correspondence between clusters of different features spaces. This alignment allows to further improve the Bag of Visual Phrases Model by adding the coherence of different feature spaces also during codebooks generation phase. The approach is formalized in the following. Let I an image, and M the number of local regions extracted by making use of a local detector [104] or through dense sampling [10,91]. Each extracted local region r_i, $i = 1, \dots, M$, is described by H different local descriptors $\boldsymbol{\phi_{ih}}$, $h = 1, \dots, H$. Each region r_i is then associated to a set of local descriptors [105] $\boldsymbol{\phi_i} = \{\boldsymbol{\phi_{i1}}, \boldsymbol{\phi_{i2}}, ..., \boldsymbol{\phi_{iH}}\}$. A vocabulary V_h is built for each type of local descriptor, and the different local descriptors $\boldsymbol{\phi_{ih}}$ of a region r_i are hence associated to visual words $\boldsymbol{v_h}$ belonging to the codebook V_h as in the classic BoVW paradigm [134]. This produces a H-tuple $\boldsymbol{p_i} = \{\boldsymbol{v_h} | h \in [1, 2, ..., H]\}$, called "visual phrase", which contains visual words of different feature spaces for each $\boldsymbol{\phi_i}$, $i = 1, \dots, M$, corresponding to the M local regions detected into the considered image I. Each image is then described by the frequency distribution of visual phrases, called "Bags of Visual Phrases". This model, called "coherent phrase model" [73], incorporates the coherence across multiple descriptors in order to describe different aspects of the appearance of a local region detected within an image. Our approach augments the coherent phrase model by improving the vocabulary generation step. In [73], H codebooks (one per local descriptor type) are generated separately and independently by using a classical clustering approach on each descriptor space. Then the images are described with a normalized multidimensional histogram in which each bin is related to a visual phrase (e.g., a 2-D distribution by considering two different local descriptors). The underlying rationale is that, although different descriptors encode different properties of a local region, they represent the same local region, hence the clustering, and the visual words belonging to different feature spaces, are in "some way" related. Hence, the coherence among different local descriptors should be exploited also in the vocabulary generation step. The main schema of the proposed approach is summarized in Fig. 4-1. First, the H different local descriptor spaces are clustered separately and K visual words (cluster centroids) are obtained for each vocabulary V_h (one visual vocabulary per local descriptor) as in the classic BoVW paradigm [134]. The relative ordering of cluster labels in all of the clustering are hence rearranged according to the first one. A $K \times K$ similarity matrix between pairs of clusters belonging to the two partitions is obtained by counting the number of elements (local image regions) they share. The Hungarian algorithm [114] is then used to find the best assignment for the cluster correspondence problem which is encoded in the computed similarity matrix. Therefore, the alignment between clusters of different partitions is thought as a classical resources assignment problem to be solved by a combinatorial optimization algorithm. We choose to exploit Hungarian algorithm since it has been successfully used in Computer Vision to solve different problems which can be seen as a resources assignment problem (e.g., cluster correspondence [123], feature matching [25]). By using the Hungarian method the alignment of the different vocabularies can be done in $O(K^3)$ time. Despite we have used the Hungarian algorithm in our experiments, there are more efficient algorithms that can be used to solve the same problem [81]. The obtained cluster correspondences are used to create H novel vocabularies where visual words are generated considering the centroids relative to both common and uncommon elements between aligned clusters (Fig. 4-1). Hence three new visual words (cluster means) per descriptor space are generated from two aligned clusters considering the operations of intersection (shared local image regions belonging to the overlap among aligned clusters) and difference (local image regions belonging to the non-overlapped parts of the aligned clusters). Notice that, although Hungarian algorithm aligns all the clusters, some of them can have no common elements (Fig. 4-1). If two clusters are fully separated, only two new cluster centers will be computed from individual ones. In this last case the two obtained visual words are equal to the original ones. After building the vocabularies separately on each feature space, these are aligned (as described above) to find coherence between the different spaces based on shared keypoints. After that, each cluster of each vocabulary (which define a visual word in the considered feature space) is splitted in subclusters (defining more than one visual word if the overlap of the aligned clusters is not empty). In this way, the quantization of a descriptor space is refined by taking into account of the quantization obtained in the other feature space. So, the refinement of each vocabulary encodes also information induced from the other vocabulary. This allows to make stronger the discriminativeness of the original Bags of Visual Phrases approach [73] as empirically demonstrated by the experimental results reported in Section 4.4. The algorithm described above generates a multidimensional representation of the image under analysis. In particular, starting from the original image, it extracts a set of local feature points, associates them to different descriptors and, by using a pre-computed set of vocabularies, creates the final multidimensional normalized histogram. Considering two descriptors with the associated codebooks consisting of K_1 and K_2 elements respectively, the final image representation is a matrix ($2D$ normalized histogram) of $K_1 \times K_2$ values[1].

4.2 Image Representation Compression

The compactness of the image representation impacts both in terms of memory storage and computational complexity of the near duplicate detection task [79]. The cost per single image query becomes a critical feature of the overall system as the number of the images stored in the dataset increases. It is then extremely useful to study some approximations of

[1] Note that at this stage other encoding methods can be used starting from the aligned vocabulary [39].

the original representation able to reduce the amount of data to be stored and used during retrieval, without considerably reduce the performance of the overall system. The analysis of the $2D$ histogram representations of the training images shows that the $K_1 \times K_2$ matrices are pretty sparse (see Section 4.4), hence only a limited set of elements (visual phrases) are actually used to describe the image content. Based on this analysis, we propose a simple and effective compression technique. The most representative and discriminative T visual phrases (i.e., T bins of the $K_1 \times K_2$ matrix, with $T \ll K_1 \times K_2$), together with their IDs (i.e., the number of row and column they belong into the matrix), can be selected to represent the image under analysis. This selection considers all the images of the training dataset on which image queries are performed. To sum up, when a query is performed on the selected training dataset considering a generic test image I, the following steps are performed:

i) generate the multidimensional histogram Ψ_I of the image I;

ii) for each image J of the training dataset, select its matrix coordinates C_J relative to its most representative and discriminative T visual phrases;

iii) select the elements of the histogram Ψ_I at the coordinates C_J;

iv) for each image J compute the similarity between the compact representation of images I and J;

v) provide as output of the query the image $\widehat{J}$ belonging to the training dataset with the lowest distance from the image I.

It is worth noting that the effectiveness of the proposed approximation depends on the number of selected bins T. To obtain a satisfactory improvement in terms of memory storage and computational load this number should be considerably lower than $K_1 \times K_2$, where K_h is the dimension of the vocabulary V_h. On the other hand few visual phrases (i.e., bins) could be not able to properly discriminate the images belonging to the dataset. A smart selection strategy of the best T bins can be then useful in finding a good trade-off between compression and retrieval performance. Specifically, we employ the statistical measure TF-IDF (term frequency-inverse document frequency) [124] for the selection of the most representative and discriminative visual phrases (i.e., to select the best T bins within the representation matrix). In this way, the importance of the bin is not only related to its frequency in the image representation but also consider the frequency of the bin (i.e., the discriminativeness of the visual phrase composed by a pair of descriptors) with respect to the entire training dataset. In other words, for each image of the training dataset, we select the most representative and discriminative T visual phrases (e.g., bins) as indicated by the TF-IDF measure. During a comparison of an query image I with an image of the training dataset J only the T visual phrases of the image J which have been selected taking into account the TF-IDF measure are considered.

4.3 The Experimental Datasets

An image *I* is considered a near-duplicate of another image *J* if its content is "similar", according to some defined similarity measures, to the image *J*. So, the definition of a near duplicate image changes accordingly with the allowed photometric and geometric variations. As in [42], we consider an image *I* a near-duplicate of another image *J* if it contains the same scene of *J* with possibly different photometric and/or geometric variations (e.g., viewpoints changes, illumination and color variations, partial scene, occlusion, different compression and camera acquisition, etc.). The problem addressed here is hence the one to enumerate all the near duplicates of a given query image in a dataset. In order to test and compare different algorithms for near duplicate image retrieval, a representative dataset should be used. Despite different datasets have been employed in literature for testing purposes, most of them are synthetic [2] [82] or obtained taking into account keyframes of videos [73]. Although synthetic datasets are compliant with the definition of near duplicate given above, they aren't representative of the real variation that can be observed in real near duplicate images (see Fig. 4-2). On the other hand, datasets built by collecting

[2] We consider a dataset as synthetic when the near duplicates are generated from a set of images (or frames of videos) by using transformations typically available on image manipulation software (e.g., ImageMagick [68]), such as colorizing, contrast changing, cropping, despeckling, downsampling, format changing, framing, rotating, scaling, saturation changing, intensity changing, shearing. To generate near duplicates the basic transformations are usually applied changing the different involved parameters and/or making combination of them.

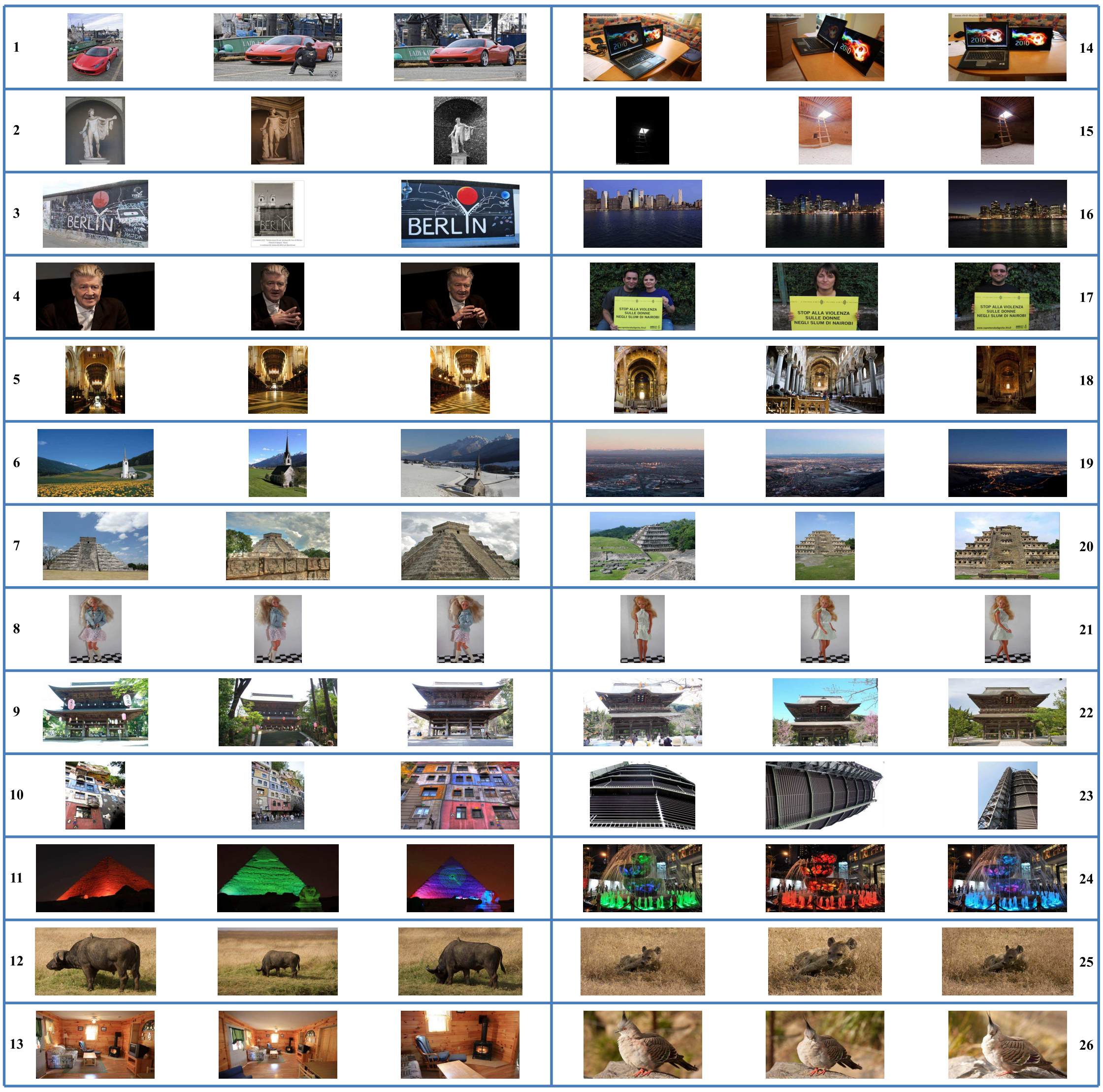


Figure 4-2: Examples of 26 different scenes belonging to the considered dataset. For each scene three near duplicates are shown.

frames of videos contain near duplicates with no variability in terms of resolution and compression factor. The classic datasets used for image retrieval testing purposes (e.g., CBIR task), such as the one introduced in [75], are not compliant with the aim of near duplicate image retrieval, where the problem is to search for the same scene with possibly different photometric and/or geometric variations, given an image as query. The above motivations induced us in building and using a new representative dataset for the problem under consideration. In this way we can properly test and compare the proposed augmented version of Bag of Visual Phrase model with respect to the original one [73]. Specifically, a dataset with images acquired by different cameras, in different conditions (e.g., viewpoint, scale, illumination, distance from the subjects, etc.), and high content variability (indoor, outdoor, object, natural scenes, etc.), has been collected from Flickr [61] and from private collections. To this aim, 525 different keywords (e.g., New York, Animal, Car, Church, Computer, Mountains, Landscape, etc.) have been chosen. Each keyword has been then used to retrieve images from Flickr. From the retrieved images a set of near duplicates have been hence manually sampled. Each specific set corresponding to a keyword contains from 3 to 34 near duplicates. The whole dataset contains 3148 images. In Fig. 4-2 some of the images belonging to the built dataset are shown. Specifically, in the figure are reported three near duplicates of 26 different scenes. As evident by visual inspection, there is a high variability in terms of scenes (outdoor, indoor, close up objects, portraits, archeological sites, buildings, animals, open scenes, etc.) as well as a high variability in terms of geometric and photometric characteristics among near duplicates of the same scene (different point of view, luminance and color variation, zoom, rotation, background variation, etc.). Moreover, different scenes have regions with similar appearances, such as in the case of the scenes with animals (see images of the scenes with number 12 and 25 in Fig. 4-2) and the ones with Japanese buildings (see images of the scenes with number 9 and 22 in Fig. 4-2). Differently than classic content based image retrieval task in which, for instance, given an image of the scene numbered as 9 in Fig. 4-2 all the images of the the scene numbered as 22 are acceptable in terms of visual similarity, in the context of near duplicate image detection this become an unacceptable error. The database was hence built to properly test the challenging task under consideration. Since near duplicate image detection techniques are usually tested on datasets used in the context of object recognition [73], we have performed tests also considering the UKBench 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 [108].

4.4 Experimental Results

In this section the effectiveness of the proposed approach is demonstrated through a number of experiments and comparisons. A first test, conducted on the dataset we built (see previous section), compares our method with respect to the coherent visual phrase model described in [73] and the technique proposed by Zhao et al. in [152, 165]. Note that both [73] and the classic BoVW approaches have been reimplemented at the best of our knowledge whereas the original code provided by the related authors has been used for [152, 165]. To properly evaluate the different methods, the experiments have been repeated three times. At each run the different approaches are executed on the same training and test sets. To this purpose, at each run we have built training and test sets by selecting images at random. Specifically, we have randomly selected one image per each set of near duplicates to build a training set with 525 different scenes, whereas two images per each set of near duplicates have been randomly selected to build the test set. All the parameters involved in the experiments have been learned from the corresponding training sets for each method. The results presented in the following are obtained by averaging the results of all three runs. For every run, training images have been used for the generation of codebooks. First, local interest points have been detected (Hessian-Laplace [104]). Afterward, two different descriptors have been extracted on each interest point: SIFT [96] and SPIN [80]. Since these descriptors are extracted considering different image properties (gradient orientation (SIFT) and intensity distribution at different distance from the center (SPIN)), they are somewhat complementary, hence can be fruitfully combined. K-means algorithm (K=500 in our tests) has been then used to produce the two independent codebooks corresponding to the two involved descriptors. The two obtained partitions have been aligned with the Hungarian algorithm to generate the new codebooks (see Section 4.1). Finally, training images have been represented by visual phrases (with a 2D histogram) by considering the new aligned codebooks. It is worth noting that the proposed procedure for codebook generation creates two novel vocabularies (one for each type of descriptor involved in the experiment) with a higher number of elements with respect to the original ones. Considering, as example, two codebooks of 500 elements, the alignment procedure will produce, in the worst case, two

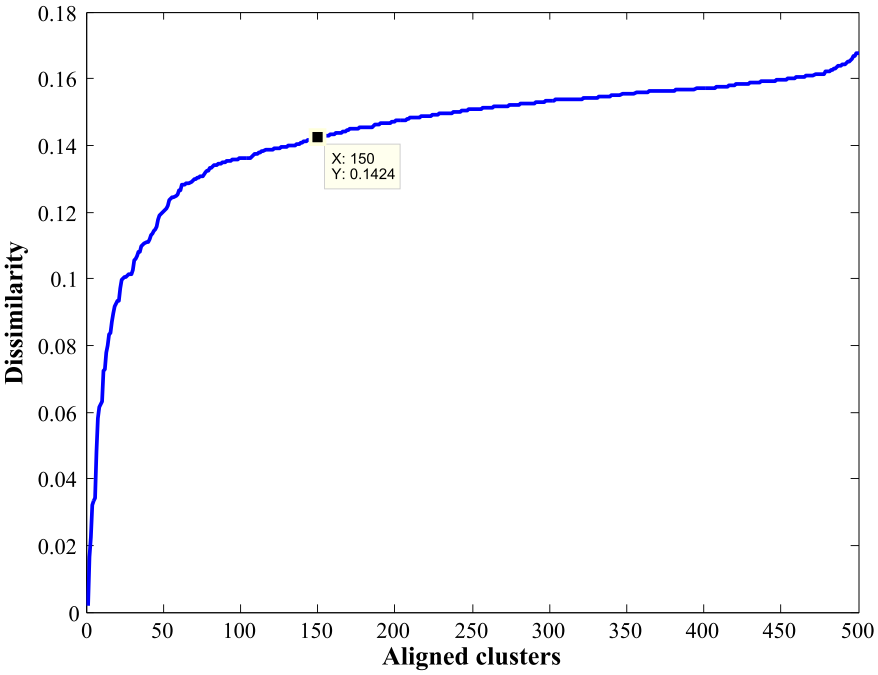


Figure 4-3: Sorted dissimilarity of aligned vocabularies. The first 150 aligned pairs of clusters can be considered "similar" in terms of shared keypoints, whereas the others are "dissimilar".

novel vocabularies of 1500 elements for each type of descriptor. In order to reduce the dimension of the final image representation maintaining at the same time good performance, analysis and tests have been performed. In particular, useful hint can be derived from the analysis of the degree of similarity between the clusters associated in the alignment procedure performed through the Hungarian algorithm. As reported in Fig. 4-3, which have been obtained sorting the aligned clusters with respect to their dissimilarity, after a certain threshold, the aligned clusters cannot be considered "similar". This means that after a given value the aligned clusters share only few keypoints (or nothing at all) and hence there is not too much coherence among these aligned clusters. The threshold imposed on cluster dissimilarity is chosen taking into account the gradient of the dissimilarity curve. At some point, the gradient of the curve starts to be very small and this fact can be used to set the threshold. Moreover, given a threshold, the number of elements of the aligned vocabulary is propery established. For example, in the case reported in Fig. 3, where a threshold which consider 150 aligned cluster is selected in correspondence of a small gradient, the final number of employed centroids is equal to $150 \times 3 + 350$ for each feature space. The cluster intersections produce 150×3 new visual words for each feature space, whereas the other not aligned 350 clusters produce 350 visual word for each feature space. So considering both, the gradient of the curve and the dimension of the final codebook, the threshold can be fixed. Taking into account the previous analysis, a more compact vocabulary can be hence generated performing the procedure for the generation of aligned codebooks (see Section 4.1) only for the aligned clusters having a high degree of similarity; for all other "dissimilar" clusters will be retained only the original centroids on the corresponding feature space. The analysis of the dissimilarity curves related to the different three training set considered in our tests, pointed out that the first 150 aligned pairs of clusters can be considered properly aligned (i.e., "similar" in terms of shared keypoints). In this way a final codebook of 800 visual words per descriptor (SIFT and SPIN) has been generated instead of one of 1500. To be fair, the comparisons with the other approaches (Hu et al. [73], BoVW SIFT and BoVW SPIN) have been performed considering codebooks with 800 elements per descriptor independently generated through K-means clustering. At each run,

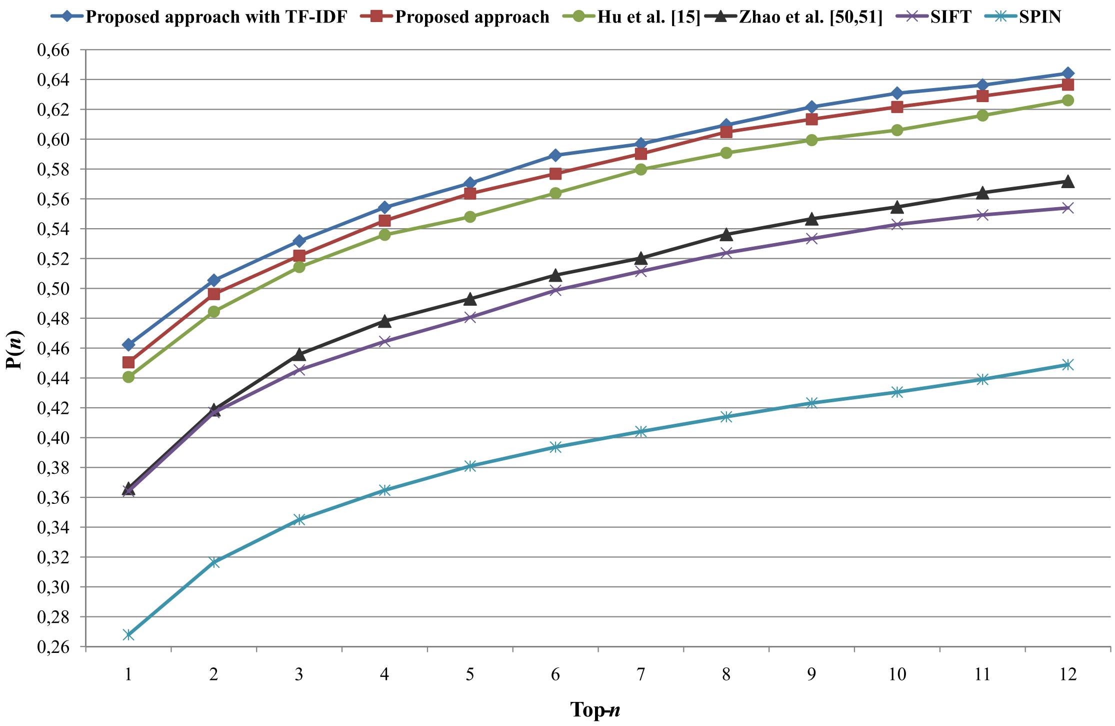


Figure 4-4: Top-n NDI retrieval performances comparison on the proposed dataset.

test images are used to perform queries on the related training dataset. Each test image is represented by a visual phrase histogram obtained considering the aligned codebooks (see Fig. 4-1). This representation is then used to retrieve images in the training dataset, by means of a similarity function between Bag of Phrases histograms. To cope with partial matching, we use the intersection distance τ defined as follows [67, 133]:

$$\tau(\Psi_I, \Psi_J) = \sum_{p=1}^{P} min(\Psi_p(I), \Psi_p(J)) \tag{4.1}$$

where Ψ_I, Ψ_J are two visual phrase histograms and $\Psi_p(.)$ is the p^{th} bin of the histogram. Both representation, with and without TF-IDF weighting scheme have been considered and compared. Each query image has been associated to a list of training images. The retrieval performance has been evaluated with the probability of the successful retrieval $P(n)$ in a number of test queries [73, 163, 166, 167]:

$$P(n) = \frac{Q_n}{Q} \tag{4.2}$$

where Q_n is the number of successful queries according to top-n criterion, i.e., the correct NDI is among the first n retrieved images, and Q is the total number of queries. The proposed approach has been compared with the original Bags of Phrases approach [73], with the approach proposed in [152, 165], as well as with respect to the classic BoVW approach considering both SIFT and SPIN descriptors. The obtained results are reported in Fig. 4-4.

Both proposed strategy, with and without TF-IDF, outperforms the original Bags of Visual Phrases, the approach proposed in [152, 165], and the classic BoVW model. We also show the precision/recall values at top-n=1 in Table 4.1. Note that the precision and recall for top-n=1 are equivalent because there is only one correct match for each query. Some visual examples of the first retrieved image on a specific query are reported in Fig. 4-5. Specifically, for some query images reported in the first column of Fig. 4-5, the first retrieved images obtained with the proposed approach and the method described in [73] are

Table 4.1: Precision/Recall values on the proposed dataset.

Method	Precision/Recall
Proposed approach with TF-IDF	0.4622
Proposed approach	0.4505
Hu et al. [73]	0.4406
Zhao et al. [152, 165]	0.3660
SIFT	0.3641
SPIN	0.2679

shown respectively in the second and third columns of Fig. 4-5. The proposed method is able to detect the corresponding near duplicate within the training set, whereas the technique proposed in [73] retrieves images which aren't a near duplicate of the queries and hence fail the aim. For completeness, further visual examples in which both approaches fail, as well as some examples in which the method of Hu et al. [73] outperforms our approach are reported respectively in Fig. 4-6 and Fig. 4-7. As evident from Fig. 4-6, often both approaches fail in the same way (i.e. selecting the same wrong image). As already stated in Section 4.2, some analysis and tests have been performed to compress the image representation in order to speed up the retrieval process and to reduce the amount of data to be stored. Although the image representation is based on a 2D histogram of 800×800 elements, only a limited number of bins are actually different from zero. Specifically, from the performed analysis we have observed that the training images are described, on average, by 1700 non-zero elements (with a standard deviation of 1039). This analysis motivated the compression strategy described in Section 4.2. Taking into account the number of non-zero elements it is possible to guess the number of bins to be used in the image representation. Several tests have been performed to validate the proposed compression strategy and to find a good trade-off between compression and retrieval performance. As shown by Fig. 4-8, the retrieval performance increases at increasing of the elements used for image representation. Moreover, the results obtained considering 1600 elements are comparable with the ones of the proposed approach without compression in which the overall 800×800 elements are involved. The considered number of bins (i.e., 1600) is very close to the number of the non-zero element computed during the aforementioned analysis (i.e., 1700). In this way we are able to obtain a compact image representation without sacrificing the retrieval performance.

It is worth noting that the built dataset is really challenging; in some cases even an human observer could have some difficulties in finding the correct near duplicate image (e.g., compare the scenes marked with number 9 and 22 in Fig. 4-2). Moreover, in the described near duplicate image retrieval system each of the 525 classes are described by only one image, a design choice that could limit the overall performance of the proposed method, but is realistic, for instance, in the context of forensic science where investigators have only one image example of a criminal scene. To further confirm the effectiveness of the proposed approach, additional experiments have been performed on the UKBench dataset [108]. This dataset, usually used for object recognition tasks, contains 10200 images of 2550 different objects. Specifically, there are four near duplicate images with photometric and/or geometric variations for each object. In our test, the training dataset has been built randomly selecting one image per class. The remaining images have been then used for testing purposes. The test has been repeated three times and the final results are obtained by averaging the results obtained on each test. As can be easily seen from Fig. 4-9 and Table 4.2, also considering this dataset, the proposed approach obtains satisfactory results. Also in this case the proposed approach outperforms the original Bag of Phrases approach [73] obtaining a good margin in terms of performances. The approach proposed in [152, 165] results worst than the original Bag of Phrases method and it is not reported in Fig. 4-9. As pointed out in Section 4.1, in terms of computational complexity the proposed approach has an additional cost due to the alignment of clusters during the codebook generation. On the other hand, this allows a richer description of the images which is reflected in the increasing of the performances with respect to the original Bag of Visual Phrases paradigm [73]. Moreover, the alignment procedure is performed just once during the vocabulary generation and it does not affect the retrieval step in terms of extra costs. Considering the computational complexity during the retrieval task, the description compression proposed in Section 4.2 helps to reduce both, space and time with to respect the original paradigm [73] by maintaining the performances of the proposed codebooks alignment framework. Moreover, regarding the retrieval task, the proposed technique is comparable with the one proposed in [152, 165] in terms of computational complexity. Indeed, considering vocabularies with size K for the different descriptors, the mapping of each image with M local regions to the related

Table 4.2: Precision/Recall values on the UKBench dataset.

Method	**Precision/Recall**
Proposed approach with TF-IDF (1600 bins)	0.7342
Hu et al. [73]	0.7003

visual vocabularies has computational complexity O(*MK*). The time needed to build the visual phrases distribution is O(*M*), whereas the compression of the image representation described in Section 4.2 takes time O(*T*). Finally, the similarity between the query and an image belonging to the training dataset has computational complexity O(*T*). Hence the overall computational complexity to represent and check a query image with to respect an image into the training dataset is O(*MK*)+O(*M*)+O(*T*). It is worth noting that by employing the compression strategy the complexity in terms of computational power as well as the one related to the memory usage, have been considerably reduced. Specifically, considering a simple 2D matrix representation (without optimized data structure such as sparse matrix) the complexity of the comparisons is O(K^2) instead of O(*T*) where $K^2 \gg T$. Hence the final complexity of the algorithm without compression is O(*MK*)+O(*M*)+O(K^2).

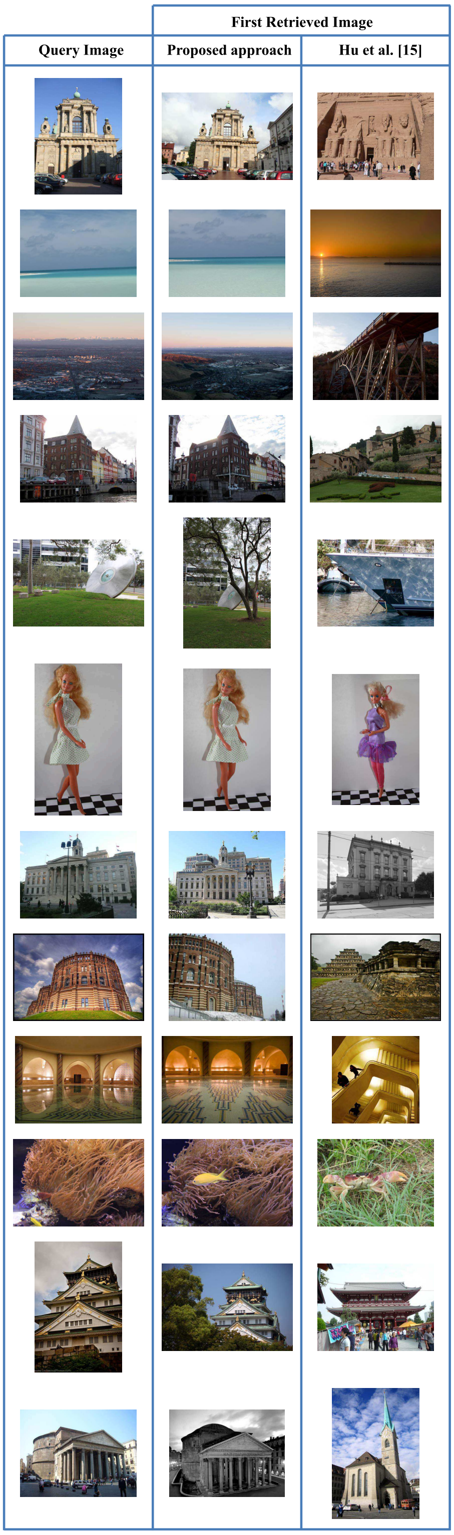


Figure 4-5: Some visual examples of the first retrieved image on a specific query. In these examples the proposed approach outperforms the method proposed by Hu et al. [73].

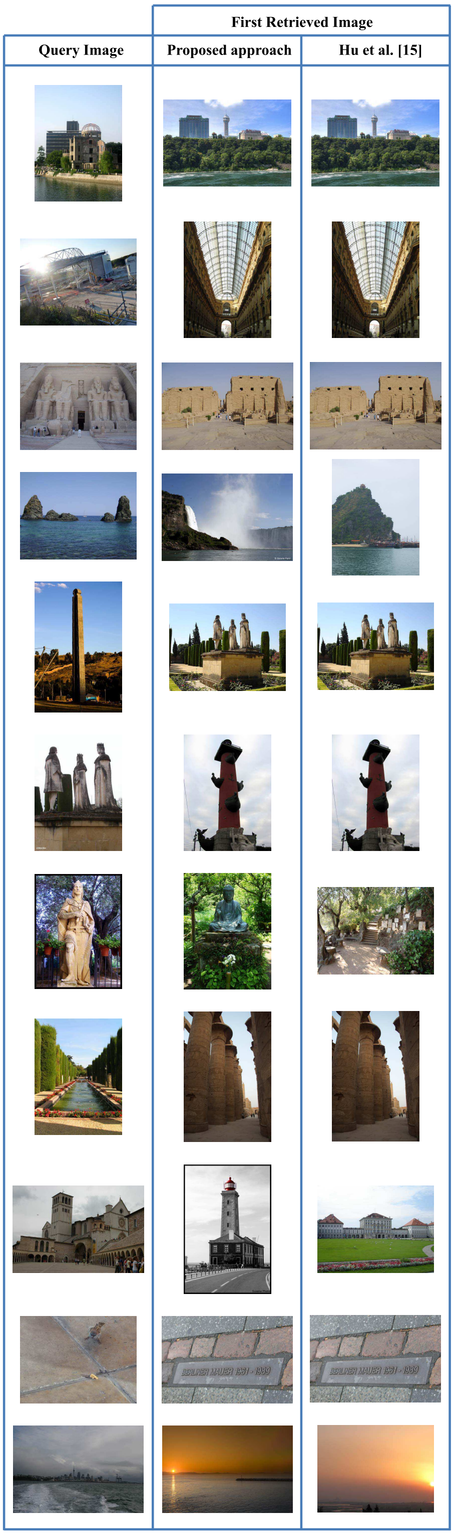


Figure 4-6: Some visual examples of the first retrieved image on a specific query. In these examples both, the proposed approaches and the one described in [73] fail.

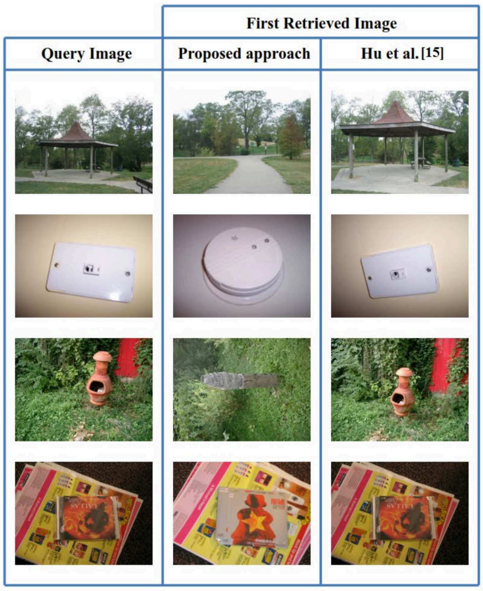


Figure 4-7: Some visual examples of the first retrieved image on a specific query. In these examples the method proposed by Hu et al. [73] outperforms the proposed approach.

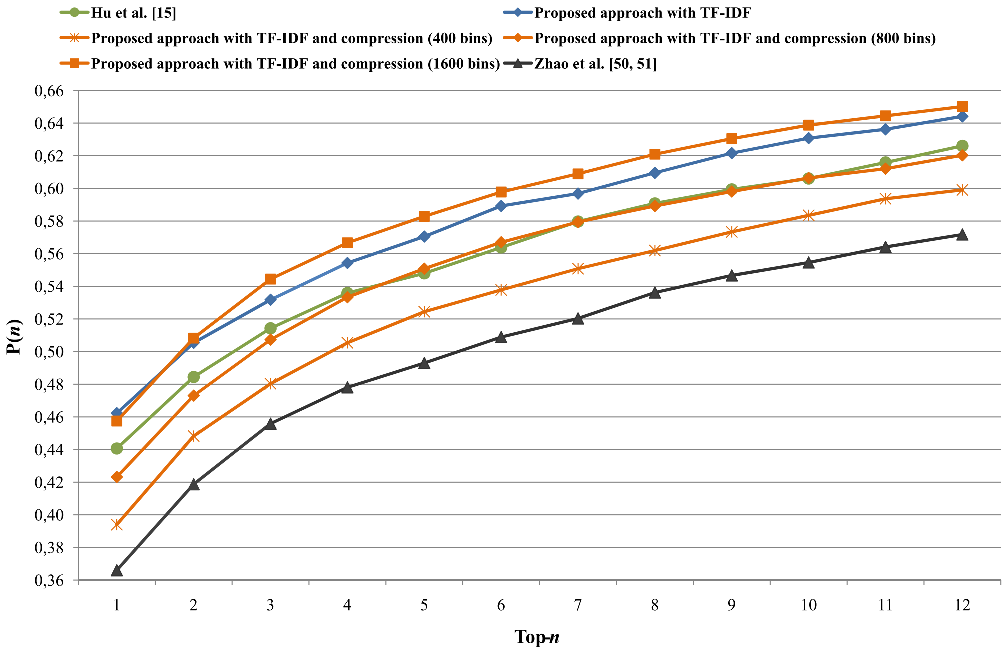


Figure 4-8: Top-n NDI retrieval performances of proposed approach with compression on the built dataset. Results are reported at varying of the number of elements involved into the image representation.

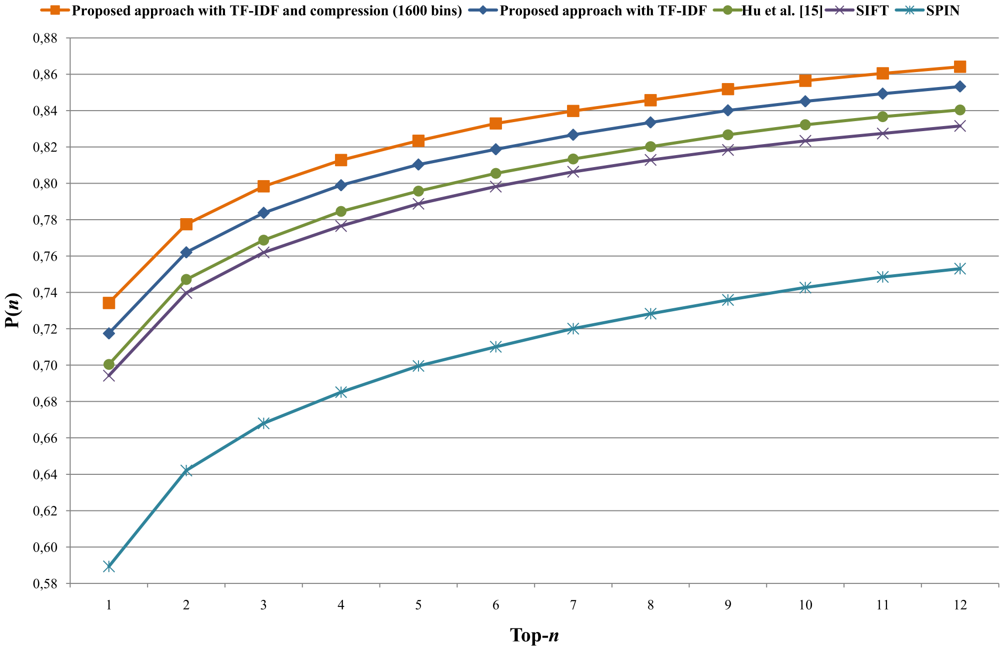


Figure 4-9: Top-n NDI retrieval performances of the proposed approach with compression on the UKBench dataset.

Chapter 5

Findings, Limitations and Perspective

In this thesis, we investigate the image understanding process from three different points of view.

Specifically, chapter 2 introduces an image representation to be exploited for scene context classification on mobile platforms. The proposed scene descriptor is based on the statistics of the DCT coefficients. Starting from the knowledge that the distribution of the AC DCT coefficients can be approximated by Laplacian distributions, and from the observation that different scene context present differences in the Laplacian scales, we proposed a signature of the scene that can be efficiently computed directly in the compressed domain (from JPEG format) as well as in the image generation pipeline of single sensor devices (e.g., smartphones, consumer digital cameras, etc.). The effectiveness of the proposed scene context descriptor has been demonstrated on representative datasets by comparing it with respect to the popular GIST descriptor [112] and the representation based on textons distributions on spatial hierarchy [10]. Moreover, the proposed scene context recognition architecture has been implemented and tested on a real acquisition pipeline of a mobile phone to demonstrate the real-time performances of the overall system. Differently than other state-of-the-art scene descriptors, the computation of the proposed signature does not need extra information to be stored in memory (e.g., visual vocabulary) or complex operation (e.g., convolutions, FFT, learning phase). The proposed holistic scene representation provides an efficient way to obtain information about the context of the scene which can be extremely useful as first step for object detection and context driven focus attention algorithms by priming typical objects, scales and locations [138, 142]. It can be also exploited to have priors for setting the parameters of the algorithm involved in the IGP (e.g., white balance) to improve the quality of the final acquired image [26].

Chapter 3 describes an approach for semantic segmentation of images. Two novel texture features based on DCT data are introduced in the Semantic Texton Forest model [128]. The proposed DCT features describe complex textures capable to recognize object and region with different frequencies characteristics. Our approach uses a limited amount of resources that allow good accuracy for real time applications. The effectiveness of the proposed semantic segmentation system has been demonstrated by comparing it with the STF and other state of the art approaches. In most of the case, our approach shows better performance overcoming the per-classes accuracy in the CAMVID database. Moreover, in a real scenario our system could shows further improvements since usually a large version of the image is available in the pipeline. This avoid to perform the proposed up-scaling block in the pipeline and generating a more reliable DCT data that are not affected by the interpolation.

In chapter 4, we propose an improvement of the coherent phrase model (Bags of Phrases) originally proposed in [73]. The main contribution of the presented approach is in augmenting the original paradigm by exploiting coherence between different feature spaces also during the codebook generation step. This is achieved through alignment of the feature space partitions obtained from independent clustering. Moreover, a method based on TF-IDF statistical measure to compress the proposed image representation for storage purposes is suggested. Experiments show the effectiveness of the described method on both, a novel and challenging near duplicate image database and a classic benchmark one. Future works will be devoted to extend the proposed alignment methodology to consider multiple ($h > 2$) types of descriptors.

Appendices

Appendix A

Saliency Based Selection for Content Aware Image Resizing

The extensive use of display devices with different resolution (e.g., on pc, tablet, smartphone, etc.) increases the demand of image resizing techniques which consider the visual content during the scaling process. Standard resizing techniques considering only geometric constraints, such as scaling, can be used only to change the image size (width and height) of a fixed percentage with respect to the original one. Scaling does not take into account the visual importance of pixels during image resizing (i.e., a resizing with respect to only one dimension introduce artifacts and distortions). Other standard operations in which outer parts of an image are removed (e.g., cropping), could produce images with loss of salient information (e.g., removal of objects or part of them).

In the last years, several techniques for content-aware image resizing (or content-based visual retargeting) have been proposed [4,8,11,41,63,117,122]. The main aim of a content-aware image resizing is the preservation of relevant visual information into the resized image. Intuitively, the goal is to remove unnoticeable paths of pixels that blend well with their surroundings, and retain the salient pixels which are important to generate the needed visual stimuli useful to correctly perceive the visual content. The algorithms should avoid distortion and changes of perspective of the image. Moreover, they should preserve edges, important textured areas belonging to the objects, size of the objects, and relevant details of the scene.

The Seam Carving, proposed by Avidan et al. in [8], is probably the most popular content-aware resizing approach. Such a technique reduces the image by removing connected path of pixels (called seams) having low-energy in the map related to the image to be resized. The authors of [8] compared different strategies to compute the energy map to be considered during the resizing process (e.g., the entropy energy computed for each pixel taking into account a fixed window, the magnitude of the gradient computed on each pixel, a saliency measure of each pixel computed as in [76], etc.). An interesting and powerful extension of standard resizing operators (i.e., scaling, cropping, etc.) and content-aware based algorithms (i.e., seam carving) can be obtained by their combination, as proposed by Rubinstein et al. in [122]. They propose an algorithm able to search for the optimal sequence of operators to be applied at each step of the resizing to get better results in terms of visual quality of the final reduced image. A drawback of this approach is that the computational complexity increases due to the use of different operators. Among others, patch-based methods have been also proposed for image retargeting or summarization. In particular, Cho et al. [41] suggested an algorithm to find an arrangement of patches of the original image that well fit in the resized image, whereas Pritch et al. [117] introduced a method to find the best Shift-Map which defines the pixel displacement useful to produce the output image. Gallea et al. [63] proposed a fast method for image retargeting based on the solution of a linear system. This model aims to find shift values for each line (row/column) preserving the distance among the relevant ones. The linearity of the considered model allows them to elaborate even large images in reasonable computational time. Building on this last technique, in our previous work [11] we have described different strategies to be employed for content-aware image resizing on mobile devices.

In this work we introduce a novel algorithm for content aware image resizing. The technique exploits the properties of Gradient Vector Flow (GVF) [157] to properly detect the seams to be removed, without introducing artifacts in the resized image. Specifically, GVF is used to produce a vector field useful to preserve objects by enhancing edges information during the generation of the possible paths to be removed. The vector field produced by GVF is also coupled with a visual saliency map [2] in order to refine the final selection of the paths to be removed. The proposed approach has been tested and compared, both qualitatively and quantitatively, with respect to state-of-the-art approaches on a representative dataset [2, 54]. Experimental results confirm the effectiveness of the proposed approach in terms of preservation of salient regions.

The rest of the appendix is organized as follows: Section A.1 and Section A.2 detail the proposed image resizing method with and without saliency exploitation. In Section A.3 the experimental phase and the results are detailed. Section A.4 discusses implementations details useful to speed up the proposed method during the resizing.

A.1 Proposed Method

One of the main issues of the content aware image resizing is the preservation of the salient information contained in the image under analysis. To this aim, our algorithm makes use of the properties of the Gradient Vector Flow (GVF) [157].

GVF is a dense force field [157] useful to solve the classical problems that affect snakes: sensitivity to initialization and poor convergence to boundary concavity. Starting from the gradient of an image, this field is computed through diffusion equations. Formally, GVF is the field **F** of vectors $\mathbf{v} = [u, v]$ that minimizes the following energy function:

$$E = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 \tag{A.1}$$

where the subscripts represent partial derivatives along x and y axes respectively, μ is a regularization parameter, and $|\nabla f|$ is the gradient computed from the intensity of the input image. Due to the above formulation, GVF field values are close to $|\nabla f|$ values in those areas where this quantity is large (energy E, to be minimized, is dominated by $|\nabla f|^2|\mathbf{v} - \nabla f|^2$), and are slow-varying in homogeneous regions (the energy E is dominated by the sum of the squares of the partial derivatives of GVF field). Hence, GVF is stronger close to the edges of objects within the image. An example of GVF field is shown in Fig. A-1. We exploit this vector field to effectively build the set of pixel paths (i.e., the seams) to be considered as candidate in the removal process. The relevance of each GVF path can be straightforward derived from the energy map obtained by the GVF magnitude associated

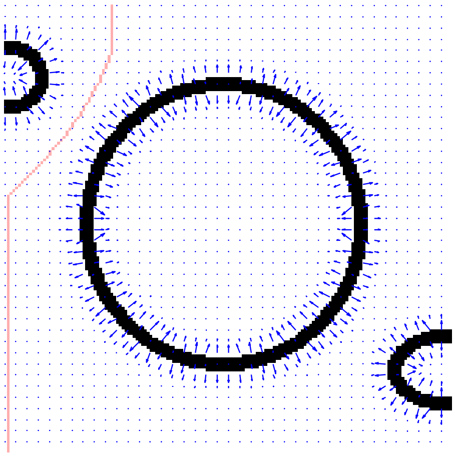


Figure A-1: Input image with its corresponding GVF field overimposed. GVF values are higher in correspondence of the edges information. The seam derived by the proposed resizing approach is shown in red. The gradient vector field forces the seams far from main contours of the objects.

Algorithm 1: Image Resizing Based on GVF

```
Input: $I$, $N$
Output: The resized image $\widehat{I}$
begin
    for $iteration \leftarrow 1$ to $N$ do
        $GVF \leftarrow ComputeGVF(I)$
        $\{s_1, \ldots, s_K\} \leftarrow SeamsComputation(GVF)$
        $\{c_1, \ldots, c_K\} \leftarrow SeamsCost(\{s_1, \ldots, s_K\}, GVF)$
        $\widehat{k} \leftarrow \text{argmin}_k\{c_1, \ldots, c_K\}$
        $I \leftarrow RemoveSeam(I, s_{\widehat{k}})$
    $\widehat{I} \leftarrow I$
end
```

to the image under consideration.

The proposed algorithm works as follows (see Algorithm 1). Let I be an image with H rows and W columns to be resized with respect to the width, and $0 < N < W$ the number of seams to be removed. First the GVF and its normalized version GVF_{norm} (i.e., each vector with norm one) are computed from the input image I considering the luminance channel (i.e., *ComputeGVF*). Several seams $\{s_1, s_2, \ldots, s_K\}$ are then built starting from the top of the image making use of the directions of the already computed GVF_{norm} (i.e., *SeamsComputation*). It is worth noting that the directions suggested by GVF_{norm} cannot be always followed. Specifically, considering a generic pixel p of co-

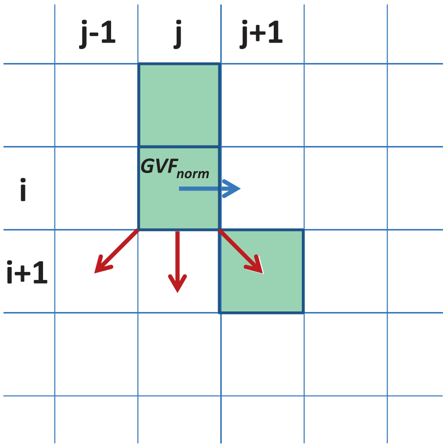


Figure A-2: An example of seam generation. Among the three possible directions (in red) the one with angle closest to the GVF_{norm} orientation (in blue) is chosen.

ordinates (i, j) belonging to a seam s_k, the next element of s_k has to be chosen among $(i+1, j-1)$, $(i+1, j)$, $(i+1, j+1)$. These pixels can be related to the following unit vectors $(-\sqrt{2}/2, -\sqrt{2}/2)$, $(0, 1)$, $(\sqrt{2}/2, -\sqrt{2}/2)$. Among the aforementioned unit vectors associated to a specific direction, the one making the smallest angle with $GVF_{norm}(i, j)$ is hence considered during the seam generation (see Fig. A-2). To this aim, a simple dot product between $GVF_{norm}(i, j)$ and the three considered unit vectors is employed. To sum up a generic seam s_k is built repeating $H-1$ times the aforementioned direction selection algorithm starting from a pixel p with coordinates $(1, w)$ at the top of the image ($w = 1, \ldots, W$ at the first iteration of the resizing). The proposed algorithm works similarly for the resizing with respect to the height.

After that the set of candidate seams $\{s_1, s_2, \ldots, s_K\}$ are computed, a cost is associated to each of them by considering the sum of the GVF magnitude $|GVF|$ related to the pixels belonging to the seam. Specifically the cost c_k of a seam s_k is computed as follows (i.e., *SeamsCost* in Algorithm 1):

$$c_k = \sum_{(i,j) \in s_k} |GVF(i, j)| \tag{A.2}$$

The seam with the lower cost c_k is hence removed from the image at each iteration (i.e., *RemoveSeam*). The GVF map is then updated and a new iteration of the seam removal algorithm is performed for each seam to be removed.

A.2 Saliency Based Selection of GVF Paths

The visual salience (or visual saliency) refers to the properties of the visual stimuli which are exploited by the human visual system in the tasks of visual attention [144] and rapid scene analysis [77]. The automatic detection of salient regions in images can be used in a broad scope of computer vision applications such as image segmentation [70], content-based image retrieval [100], object detection [38] and recognition [154].

Several saliency estimation methods have been proposed in literature. Some of them, such as the algorithm proposed by Itti et al. [77], originate from the biologically plausible visual architecture proposed by Koch and Ullman [83], whereas others, such as the method presented by Achanta et al. in [2], are purely computational and do not make any assumption on biological architecture. Finally, techniques based on combining both paradigms, biological and computational, have also been published, as in the work of Harel et al. [71]. All previously mentioned approaches estimate the visual importance of image pixels starting from information extracted in the uncompressed domain. Since most images (e.g., over internet) are stored in the compressed domain of joint photographic expert group (JPEG), Fang et al. [54] have proposed a method to extract saliency directly in the JPEG domain by exploiting information of intensity, color, and texture encoded by the discrete cosine transform (DCT) coefficients on each 8×8 block.

Visual saliency estimation algorithms have straightforward application in content based visual retargeting. Indeed, all the state-of-the-art retargeting algorithms (i.e., [8]) detect the paths to be removed (i.e., the seams) taking into account of an energy map which encodes the importance of each pixel in terms of content. A successful seam carving algorithm should ensure that the most important image regions pointed out by the energy map should not be removed. The algorithm we presented in Section A.1 makes use of the magnitude of the GVF as energy map to drive the selection of the seams to be removed. Despite this information is useful to take care of the saliency of the edges, it does not consider other saliency information.

In Achanta et al. [4] a visual saliency map able to uniformly highlight salient regions with well-defined boundaries [2] has been used for content aware image resizing purpose; the classic seam carving algorithm proposed by Avidan et al. [8] has been employed by replacing the energy map computed using the L_1-norm of the image intensity gradient, with the saliency map computed as proposed in [2]. Results presented in [4] and [54] emphasized the fact that by using the visual saliency better performances, with respect to the state-of-the-art methods, are achieved. This strongly motivated us to couple the proposed GVF based approach with saliency information for retargeting purpose.

Differently than [4] and [54] we propose to use visual saliency only for the selection of seams to be removed after that these paths are generated by exploiting the gradient vector flow as detailed in previous section. In this way we are able to combine different kinds of saliency information; the one related to the edges given by the GVF and the one related to the saliency objects within the image encoded by the saliency map. In our experiments we used the saliency map estimator proposed by Achanta et al. [4]. To include visual saliency information, we first generate the seams exploiting the GVF, and then perform the selection based on saliency. Referring to the Algorithm 1 in previous section, we need to simply replacing the function $SeamsCost$ defined by the equation (A.2) with the following one:

$$c_k = \sum_{(i,j) \in s_k} Saliency(i,j) \tag{A.3}$$

where $Saliency(i,j)$ is the value of visual saliency of the pixel (i,j) computed as described in [2]. The new resizing procedure is summarised in Algorithm 2. It is important to note that in our Algorithm 2 the saliency map related to the image is computed just one time independently from the seams to be removed.

A.3 Experimental Results

As pointed out in [4, 8, 54], the performance of a content-aware image resizing algorithm strongly depends on the adopted energy map which captures the salient regions of an im-

Algorithm 2: Image Resizing Based on Saliency Selection of GVF Paths

```
Input: $I$, $N$
Output: The resized image $\widehat{I}$
begin
    $Saliency \leftarrow ComputeSaliency(I)$
    for $iteration \leftarrow 1$ to $N$ do
        $GVF \leftarrow ComputeGVF(I)$
        $\{s_1, \ldots, s_K\} \leftarrow SeamsComputation(GVF)$
        $\{c_1, \ldots, c_K\} \leftarrow SeamsCost(\{s_1, \ldots, s_K\}, Saliency)$
        $\widehat{k} \leftarrow \mathrm{argmin}_k\{c_1, \ldots, c_K\}$
        $I \leftarrow RemoveSeam(I, s_{\widehat{k}})$
    $\widehat{I} \leftarrow I$
end
```

age. As described in previous sections, we propose to use GVF to build the seams during the resizing. The selection of the seams to be removed is then driven by GVF magnitude or by the saliency map. As estimation approach to build the visual saliency map we used the one proposed in [2][1]. In order to evaluate the results of our basic approach (i.e., the Algorithm 1 which exploits the equation (A.2)) and do not consider saliency information, we have compared it with respect to the classic Seam Carving algorithm proposed by Avidan et al. [8], and the approach recently proposed by Gallea et al. [63]. The approach in [8] has been re-implemented, whereas the original code of the method in [63] has been provided by the authors. While [8] proposes a local-based approach which takes into account the gradient of the image to select the seams to be removed, the approach in [63] is a global-based approach in which an objective function is considered to solve an optimization problem. In [63] the product of the gradient of the image and the saliency map proposed by Itti et al. [77] is taken into account as energy map during the resizing. Moreover, to underline the contribution of coupling GVF path extraction with saliency based selection (i.e., the Algorithm 2 in which equation (A.3) is employed and saliency map is computed as in [2]), we have compared the proposed saliency based selection approach with respect to the one proposed in [4]. Similarly to our approach, the one in [4] uses the saliency map proposed in [2] allowing a fair comparison.

[1] Note that other visual saliency maps can be used, such as the one proposed in [77] or in [54]. In our experiments we have used the map proposed in [2] since this has obtained good results both, in terms of saliency estimation and computational cost. The original code useful to compute this saliency map is available at the website of the authors.

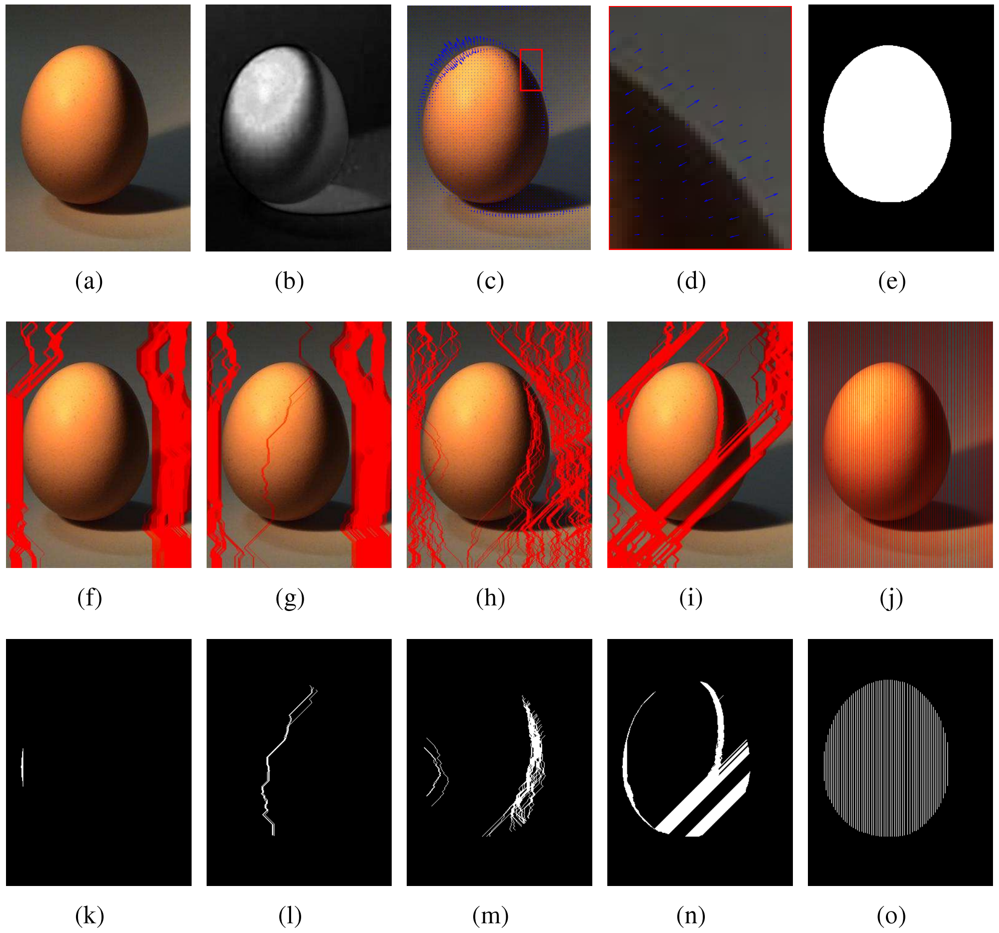


Figure A-3: Visual assessment of the involved algorithms by resizing the input image at 70% of the width. (a) Original image. (b) Saliency map [2] related to the image in (a). (c) Gradient Vector Flow map [157] of the image in (a). (d) Zoomed version of the zone marked with the red bounding box in the image in (c). (e) Ground-truth saliency mask related to the image in (a). (f), (g), (h), (i), (j) show in red the seams removed employing respectively the proposed Algorithm 2, the Algorithm 1, Avidan et al. [8], Achanta et al. [4] and Gallea et al. [63]. In (k), (l), (m), (n), (o) are shown the final maps obtained by combining the ground-truth mask shown in (g) and the maps of the removed seams which are reported in red in (f), (g), (h), (i), (j) respectively. These maps indicate the importance of the removed seams in terms of saliency. The values of these last maps are used to compute the corresponding saliency costs (i.e., the sum of values for each map) and hence employed to compare the different algorithms. As can be assessed by visual inspection of the image in (b), the saliency map alone is not able to capture some information about the edges (e.g., in correspondence of the shadow). On the other hand the GVF gives its contribution around the edges (see image in (c) and (d)). The combination of Saliency and GVF is hence able to exploits information from both sources.

In order to objectively assess the performances of the aforementioned methods, we have compared the different approaches on the dataset used in [2,54] for saliency detection. This dataset is composed by 1000 images labeled with corresponding accurate object-contour based ground-truth saliency segmentation. The dataset contains enough varieties of scenes and objects which also appear in multiple instances and in different locations (not only centered). For each image I of the dataset, the ground-truth map G_I denotes which pixels of the image are important in term of saliency. In Fig. A-3(a) and Fig. A-3(e) are shown respectively an image considered in the experiments and its corresponding ground-truth map (i.e., G_I). Since the aim of content-aware image resizing is to preserve salient regions, we used the following cost function in order to objectively evaluate the performances of a specific algorithm A involved in the comparison:

$$Cost(I, A, \lambda, d) = \sum_{p \in \psi_{A,\lambda}(I)} G_I(p) \tag{A.4}$$

where $\psi_{A,\lambda}(I)$ is the final set of pixels removed by employing the algorithm A during the resizing of the image I of a scale factor $\lambda \in \{95\%, 90\%, 85\%, 80\%, 75\%, 70\%\}$ with respect to the maximum dimension of the image (as defined by equation (A.5)), and $G_I(p)$ indicates the importance of the removed pixel p in the image I.

$$d = \operatorname*{arg\,max}_{\widehat{d} \in \{width, height\}} Size(I, \widehat{d}) \tag{A.5}$$

This cost can be used to fairly compare the performances of the different algorithms at varying of the scale factor. A lower cost value indicates better performances (i.e., more salient pixels are preserved in the resizing). We have measured the performances of the different algorithms on the aforementioned dataset at varying of the scale factor. The final results are obtained by averaging the results of all the executions for a specific scale factor λ.

Fig. A-3 reports an example of the seams removed by the different algorithms when resizing the original image at $\lambda = 70\%$ of the width. Red lines in Fig. A-3(f), (g), (h), (i), (j) correspond to the ones in the maps $\psi_{A,\lambda}(I)$ (i.e., the removed seams) obtained with the different algorithms, whereas Fig. A-3(k), (l), (m), (n), (o) depict the values $G_I(p)$ used to compute the cost function in equation (A.4) taking into account of one of the five compared algorithms. In this simple example containing a single object with non uniform illumination, it is clearly visible that the proposed approaches (Algorithm 1 and 2) are really powerful in preserving edges of objects and their original size. Fig. A-3(d) shows a zoomed area related to the GVF of the image in Fig. A-3(a) devoted to highlight the information exploited into the seam removal process in order to better preserve edges.

In Fig. A-4 are reported the results obtained by the three different algorithms which exploit the magnitude of the image gradient to select seams to be removed during the resizing: our Algorithm 1, the one proposed by Gallea et al. [63], and the original Seam Carving algorithm proposed by Avidan et al. [8]. The results are shown at varying of the percentage of the resizing.

Further experiments to test the robustness of the GVF based approach with respect to noisy input have been performed. Specifically, each image within the considered dataset has been corrupted with Gaussian noise $N(0, \sigma)$ and then the resizing has been performed considering a scale factor $\lambda = 80\%$ with respect to the maximum dimension of the input image. The results obtained by the three different algorithms which exploit the magnitude of the image gradient at varying of $\sigma \in \{0, 5, 10, 15, 20, 25, 30\}$ are reported in Fig. A-5. In all cases, the proposed approach outperforms the other content-aware based algorithms.

The proposed method based just on GVF information (i.e., Algorithm 1) achieves the best results demonstrating that the process of building seams by exploiting GVF more effectively preserves salient areas and hence removes less crucial pixels. Some visual results obtained with the aforementioned algorithms are shown in Fig. A-6.

By coupling the Algorithm 1 with a saliency estimator (i.e., the one in [2,4] in our experiments) the proposed strategy summarized by Algorithm 2 outperforms the approaches from which the solution originates. Fig. A-7 shows the results of the proposed approach based on saliency selection (i.e., Algorithm 2) with respect to the approach proposed in [4]. The results obtained by the Algorithm 1 are also reported as baseline. Although both approach based on saliency selection outperform the Algorithms 1, our proposal achieve the best margin in terms of saliency preservation performances.

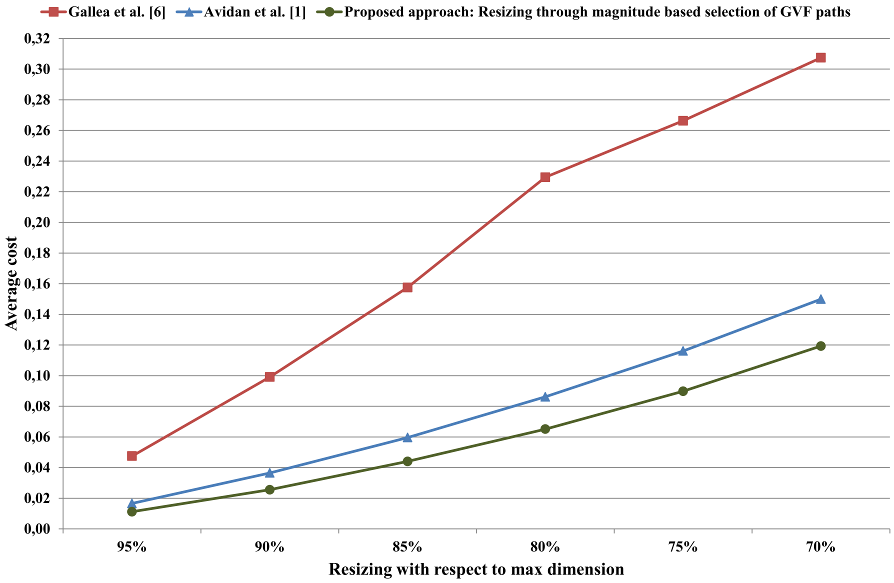


Figure A-4: Average cost computed over 1000 test images at varying of percentage of resizing. A lower value indicates that more salient pixels are preserved (i.e., better performances).

To visually assess the results obtained with the five compared algorithms, some visual results obtained by resizing images with a scale factor of 70% with respect to their original dimension (width or height) are shown in Fig. A-8 and Fig. A-9. A visual comparison reveals that the proposed approach with saliency based selection of GVF paths better preserves the main salient regions (i.e., the areas with objects).

In Fig A-10 and Fig. A-11 some examples of progressive resizing are shown with respect to the different compared algorithms. As can be easy assessed by visual inspection (Fig. A-10), already at 5% of the resizing some approaches remove information from the object (e.g., see the results at 4^{th} and 5^{th} rows), whereas the proposed Algorithm 2 works well in almost all cases. Comparing the results of 1^{st}, 2^{nd} and 4^{th} rows in Fig. A-11 it is straightforward to figure out that the exploitation of both, the GVF for seams generation and visual saliency for seams selection (as done by our Algorithm 2), more information about the salient object is retained.

To better highlight the peculiarities of the proposed approach, more visual examples are shown in Fig. A-12 and Fig. A-13. Specifically, first and second rows show examples

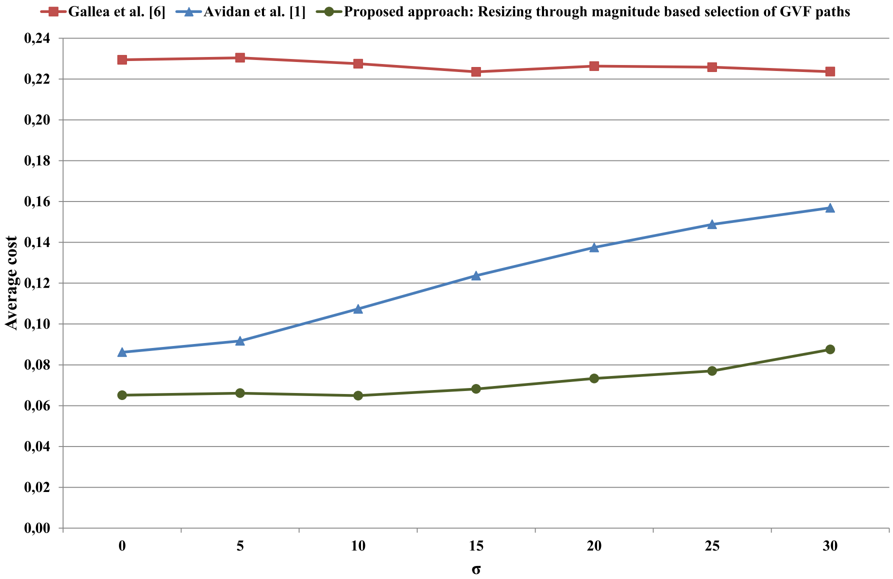


Figure A-5: Average cost computed at varying of noise by considering the 1000 test images resized at 80% of the maximum dimension. A lower value indicates that more salient pixels are preserved (i.e., better performances).

of scenes with edges and textures (i.e., the wall) and one saliency object. Our Algorithm 2 clearly preserves the visual content of the scene by maintaining both size of the object and the details related the visual stimuli of textures and edges. In the images in rows three and four, scenes with evident edges are shown. Also in this case the proposed algorithm produces the best results by maintaining the principal salient region (i.e., the plate with text) and the overall context (e.g., the fence and the background information). The example in the fifth row shows the preservation of perspective in the resizing, whereas the other examples are useful to assess the maintenance of size and details of objects as well as the other information which define the context of the scene.

All the aforementioned experiments clearly demonstrate that our proposal based on saliency selection of gradient vector flow paths outperforms both, the proposed basic strategy summarized by Algorithm 1 in which paths are selected just considering the GVF magnitude (i.e., equation (A.2)), as well as the method presented by Achanta et al. in [4], where the classic seam carving algorithm [8] is modified to consider the visual saliency of images.

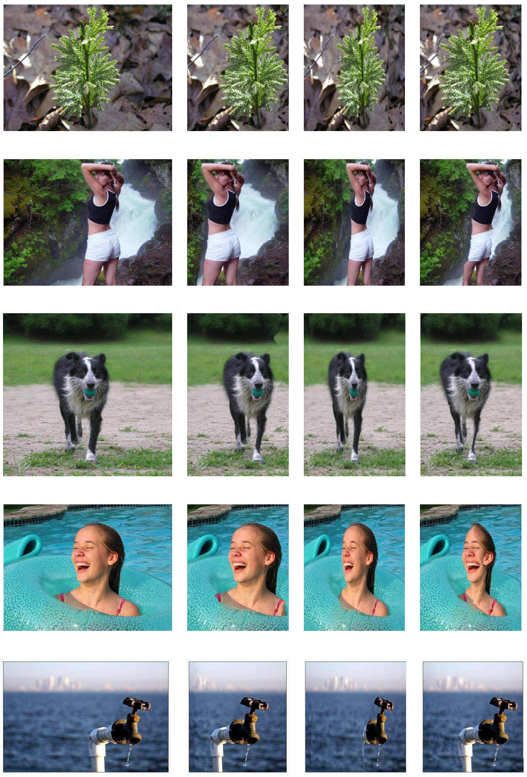


Figure A-6: Examples of content-aware image resizing. 1^{st} column: original image. 2^{nd} column: our Algorithm 1. 3^{rd} column: Gallea et al. [63]. 4^{th} column: Avidan et al. [8].

A.4 Implementation Details and Computational Complexity

In almost all approaches for content-based image resizing the computing of a seam consists in building the path of minimum cost from the top row (left column) of the image to the bottom (right) one. Typically state-of-the-art approaches use dynamic programming to this aim [4, 8, 11, 54, 122]; the algorithms consider all the possible row (column) paths to choose the seam to be removed at each iteration with computational time O(HW) for an image with size $W \times H$. Although the proposed approach has the same computational cost per iteration, we have exploited the properties of the GVF to reduce the number of paths to be considered. Indeed, the GVF is a vector field which is used by our algorithm to keep away the seams from the edges (see Fig. A-1). The rationale to reduce the number of paths to be considered at each iteration is that paths starting from neighbouring pixels (at the first

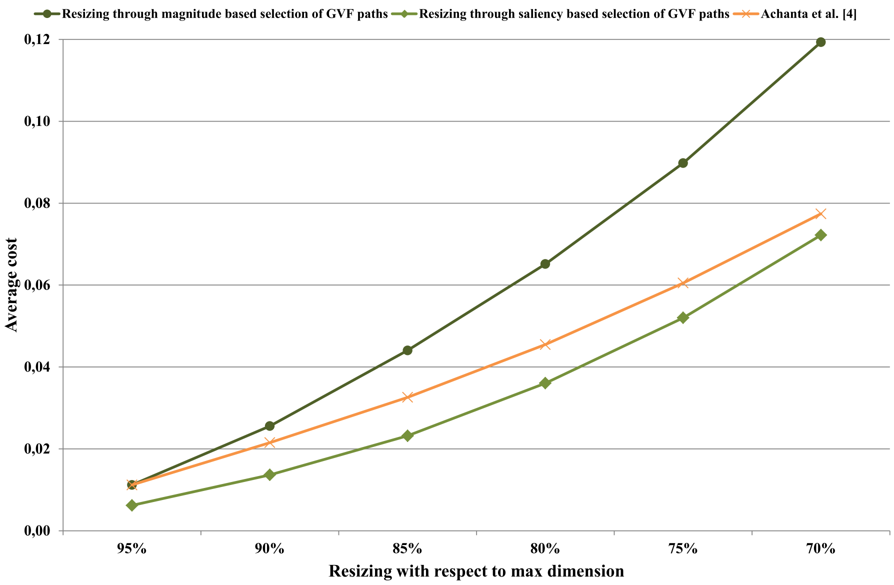


Figure A-7: Average cost computed over 1000 test images at varying of percentage of resizing. A lower value indicates that more salient pixels are preserved (i.e., better performances).

Table A.1: Average time in seconds needed to perform a resizing at 70% of the image dimension.

Method	Time
Proposed approach - all seams	92.937
Proposed approach - $\frac{1}{2}$ seams	58.832
Proposed approach - $\frac{1}{4}$ seams	41.909
Proposed approach - $\frac{1}{8}$ seams	33.357
Proposed approach - $\frac{1}{16}$ seams	28.971
Proposed approach - $\frac{1}{32}$ seams	26.858
Proposed approach - $\frac{1}{64}$ seams	25.594
Proposed approach - $\frac{1}{128}$ seams	25.136
Avidan et al. [8]	21.587
Achanta et al. [4]	21.464
Gallea et al. [63]	0.385

row or column) follow similar GVF flow in building the corresponding seams. Hence we have tested the proposed approach considering $\frac{1}{2^n} * W$ (or $\frac{1}{2^n} * H$), $n = 1, 2, \ldots, 7$, equally spaced starting pixels at each iteration during the resizing of the width (or height).

We report the experimental results obtained by reducing the number of seams to be considered on each iteration in order to decrease the computational cost of our algorithm as described above. The experiments have been done on a notebook equipped with a CPU intel core i7-2670QM 2.20GH with 8 Gb of Ram by using a Matlab implementation. To perform the test we have run the proposed algorithm by considering just $\frac{1}{2^n} * W$ (or $\frac{1}{2^n} * H$) equally spaced starting pixels at each iteration during the resizing of the width (or height). In Fig. A-14 the average cost indicating the accuracy of the resizing is reported at varying of the number of paths considered at each iteration, whereas in Table A.1 the average computational time in seconds is reported. The experimental results demonstrate the effectiveness of the proposal which reduces the computational cost during the resizing by maintaining almost the same performances in terms of saliency preservation (see also Fig. A-15).

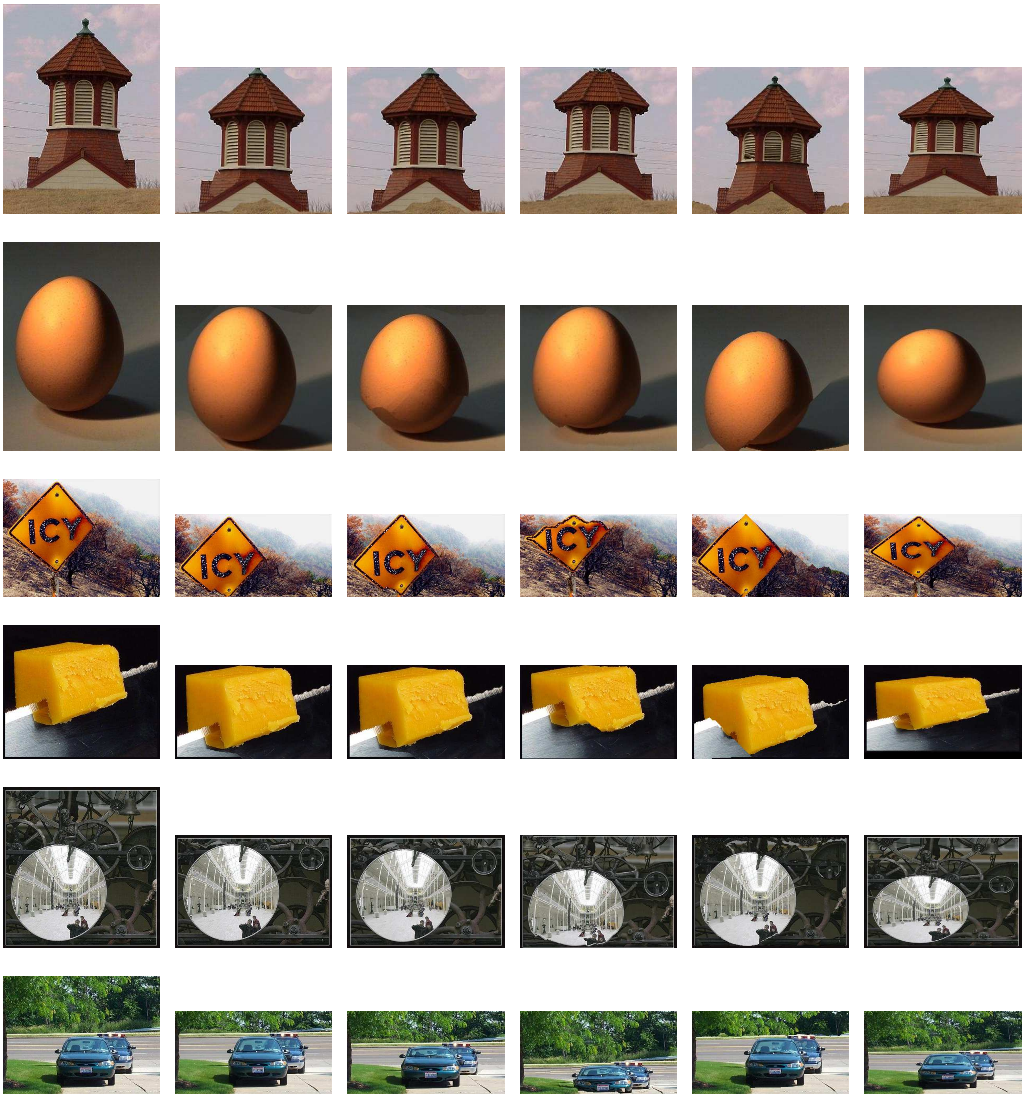


Figure A-8: Examples of content-aware image resizing at 70% of the height. 1^{st} column: original image. 2^{nd} column: our Algorithm 2. 3^{rd} column: our Algorithm 1. 4^{th} column: Avidan et al. [8]. 5^{th} column: Achanta et al. [4]. 6^{th} column: Gallea et al. [63].

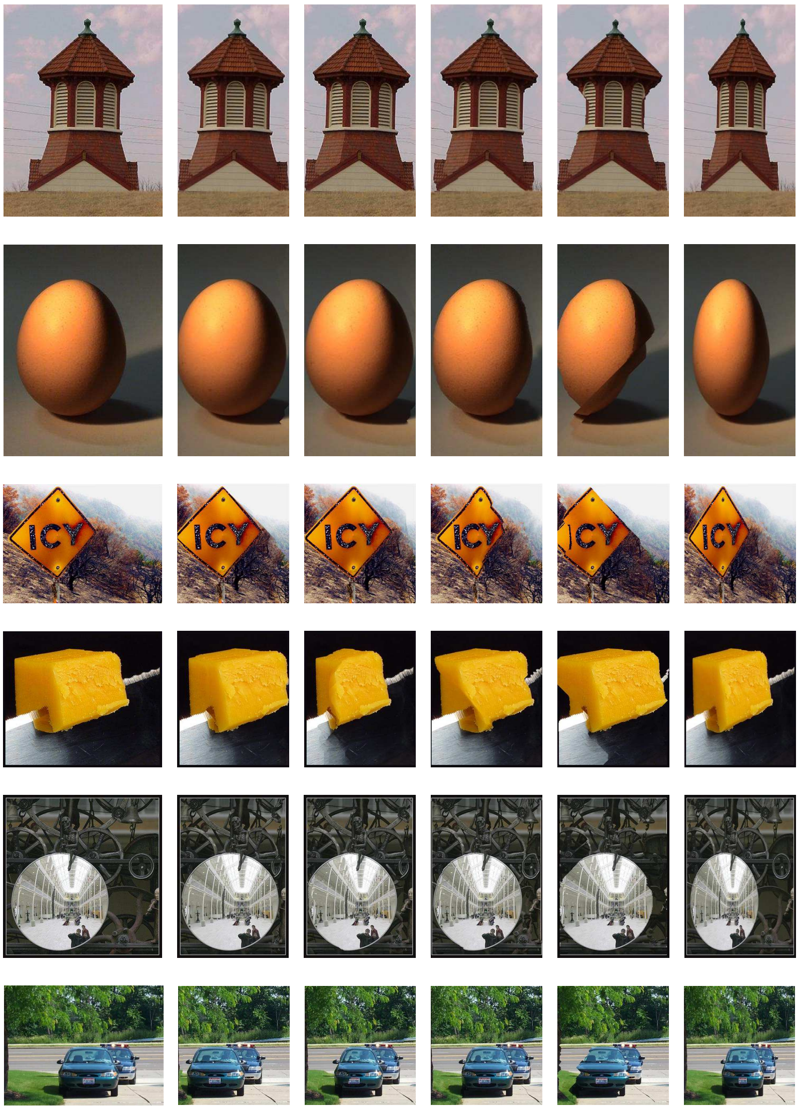


Figure A-9: Examples of content-aware image resizing at 70% of the width. 1^{st} column: original image. 2^{nd} column: our Algorithm 2. 3^{rd} column: our Algorithm 1. 4^{th} column: Avidan et al. [8]. 5^{th} column: Achanta et al. [4]. 6^{th} column: Gallea et al. [63].

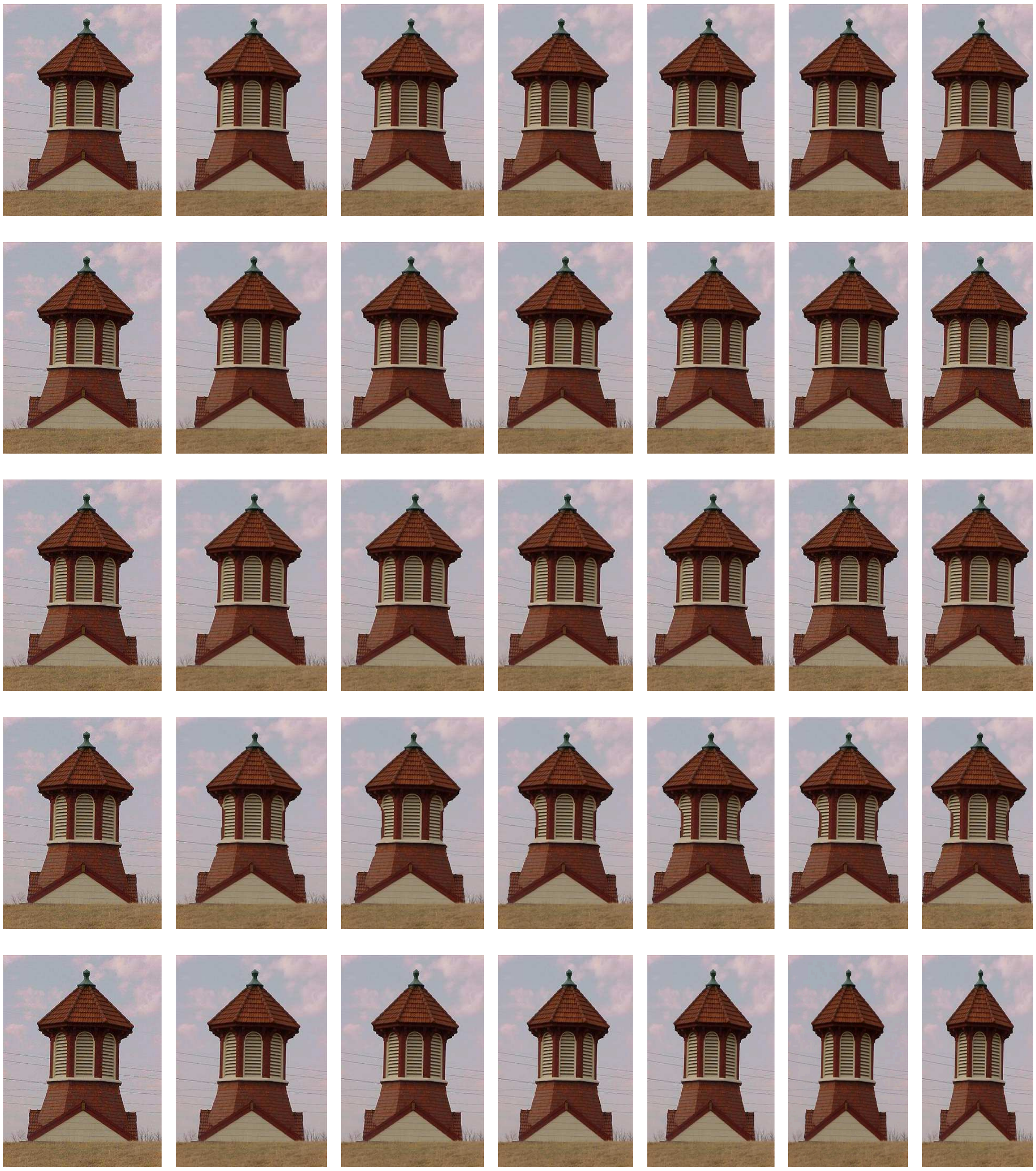


Figure A-10: Example of progressive resizing with respect to the width. Rows are related to the different algorithms: 1^{st} our Algorithm 2, 2^{nd} our Algorithm 1, 3^{rd} Avidan et al. [8], 4^{th} Achanta et al. [4], 5^{th} Gallea et al. [63]. Columns are related to the resizing factor with respect to the width: 1^{st} original image, 2^{nd} 5%, 3^{rd} 10%, 4^{th} 15%, 5^{th} 20%, 6^{th} 25%, 7^{th} 30%.

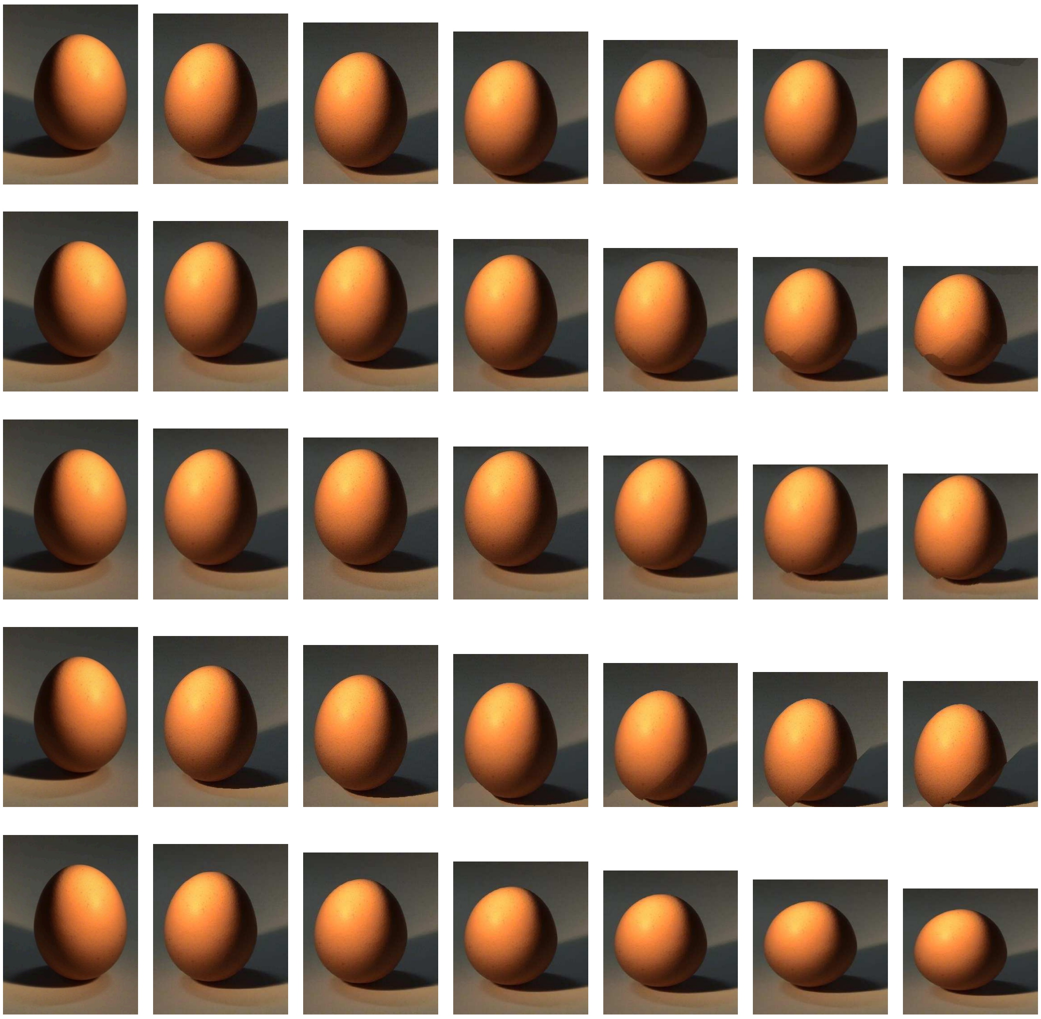


Figure A-11: Example of progressive resizing with respect to the height. Rows are related to the different algorithms: 1^{st} our Algorithm 2, 2^{nd} our Algorithm 1, 3^{rd} Avidan et al. [8], 4^{th} Achanta et al. [4], 5^{th} Gallea et al. [63]. Columns are related to the resizing factor with respect to the height: 1^{st} original image, 2^{nd} 5%, 3^{rd} 10%, 4^{th} 15%, 5^{th} 20%, 6^{th} 25%, 7^{th} 30%.

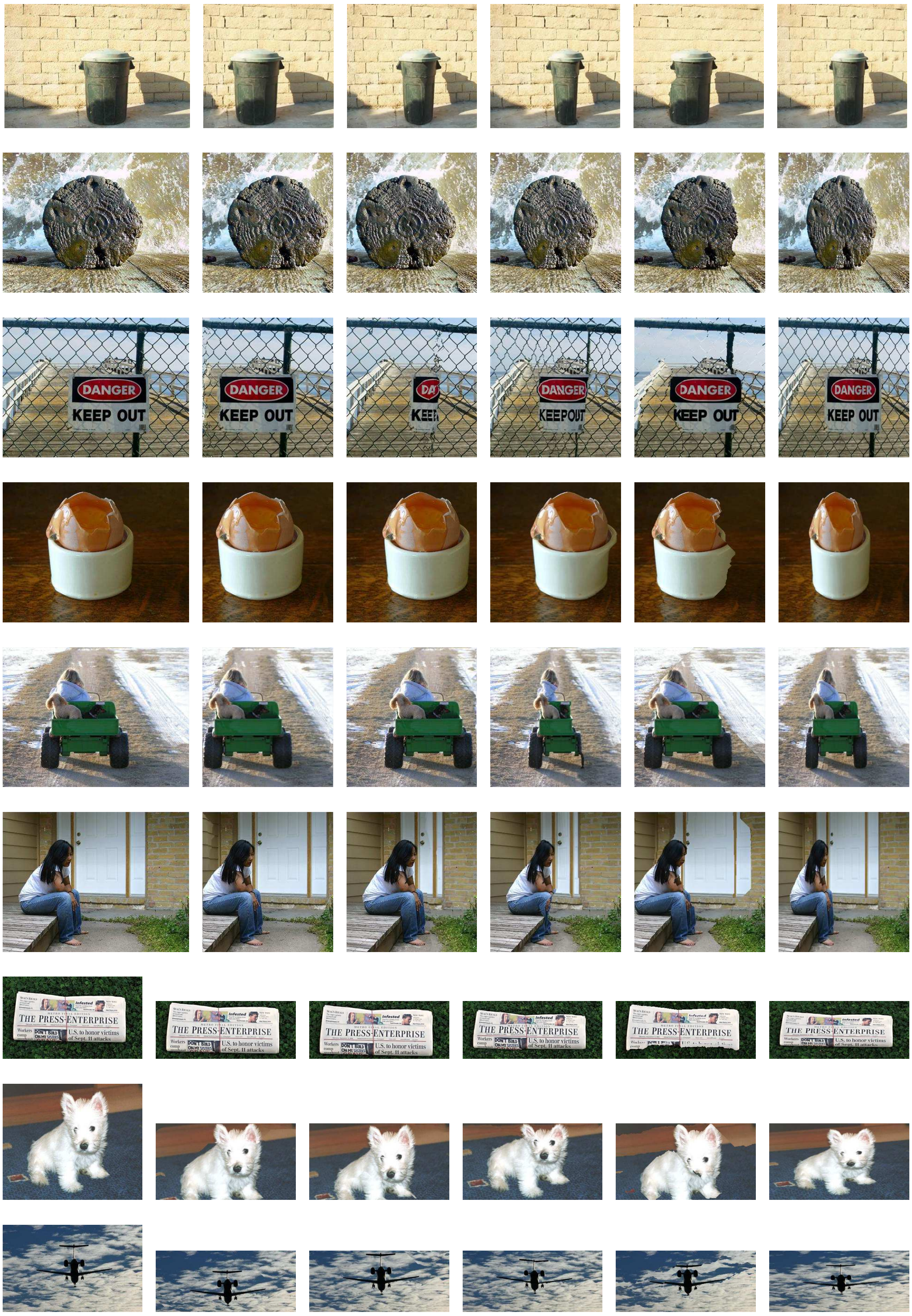


Figure A-12: Examples of content-aware image resizing of scenes containing objects and contexts with edges, textures, and different prospective. Images are resized at 70% of width/height. 1^{st} column: original image. 2^{nd} column: our Algorithm 2. 3^{nd} column: our Algorithm 1. 4^{th} column: Avidan et al. [8]. 5^{th} column: Achanta et al. [4]. 6^{rd} column: Gallea et al. [63].

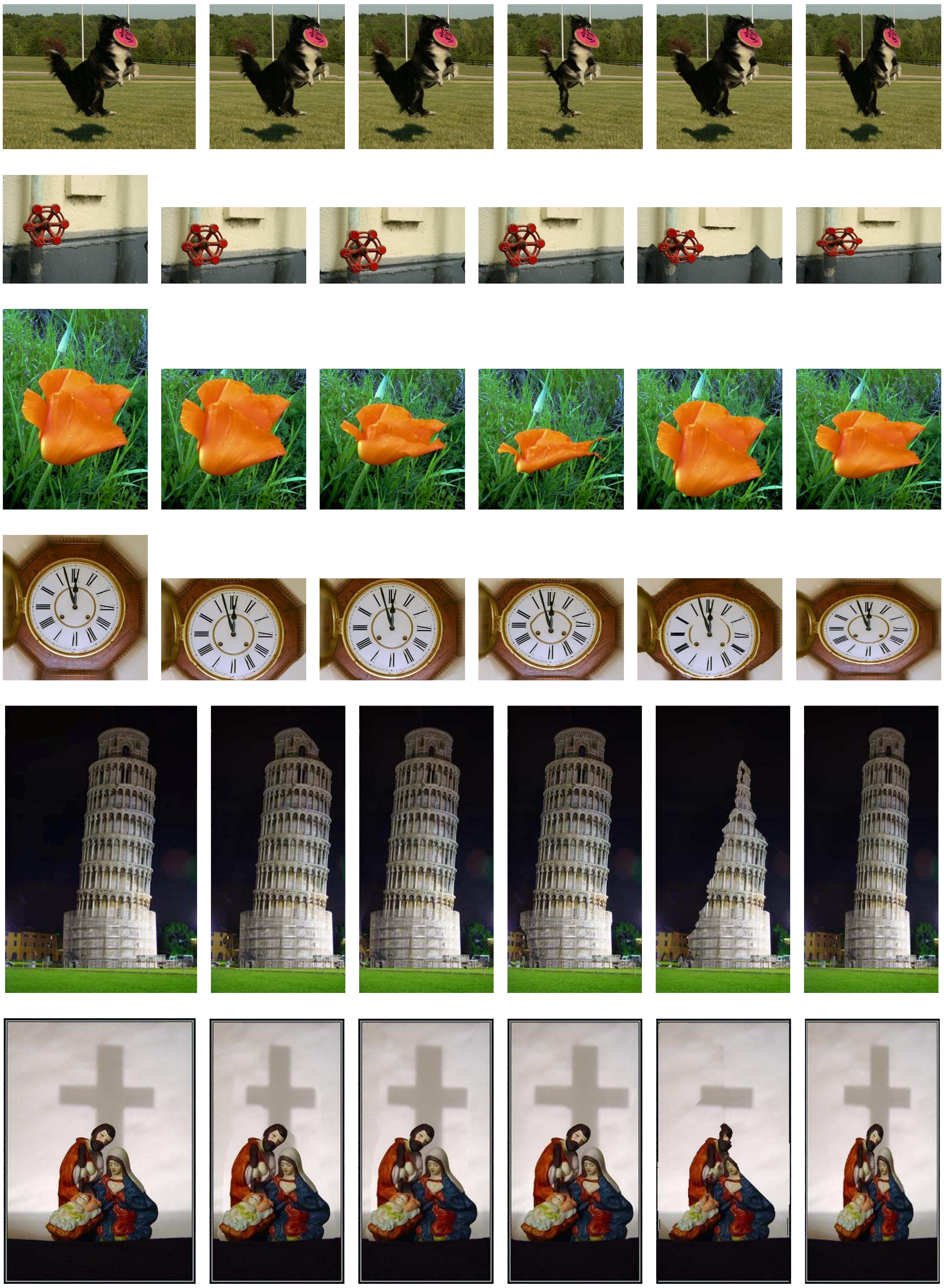


Figure A-13: Examples of content-aware image resizing of scenes containing objects and contexts with edges, textures, and different prospective. Images are resized at 70% of width/height. 1^{st} column: original image. 2^{nd} column: our Algorithm 2. 3^{nd} column: our Algorithm 1. 4^{th} column: Avidan et al. [8]. 5^{th} column: Achanta et al. [4]. 6^{rd} column: Gallea et al. [63].

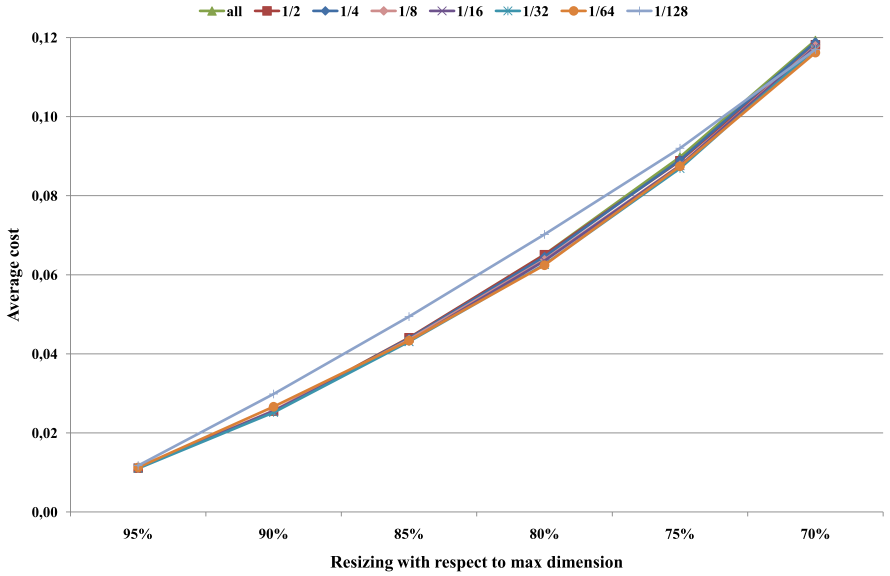


Figure A-14: Average cost of Algorithm 1 computed over 1000 test images at varying of the number of seams considered during the resizing.

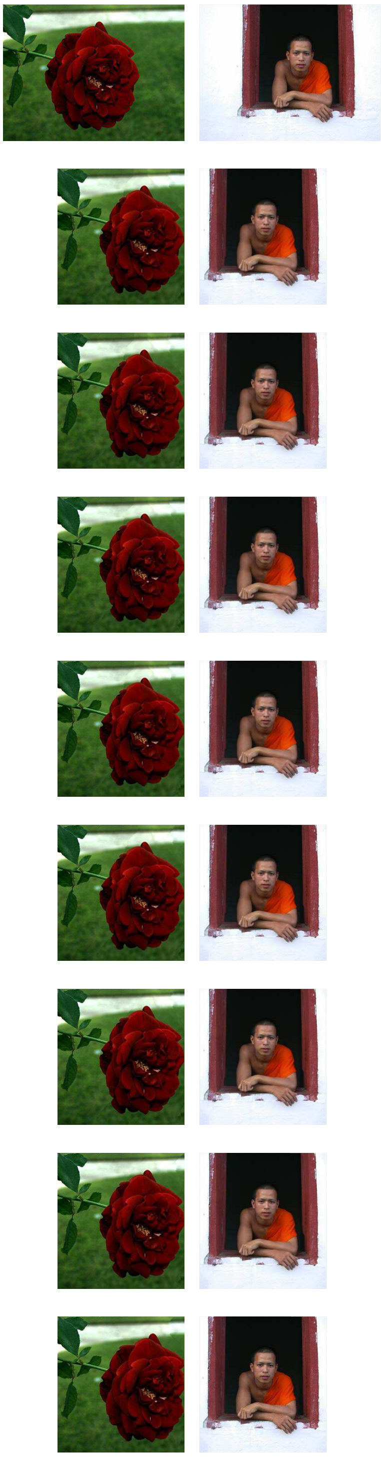


Figure A-15: Resizing images at 30% of the width by considering a reduced number of seams. Top: original images. 1^{st} row: resizing with all seams. From the 2^{nd} to the 9^{th} row are shown the results by considering respectively $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$ and $\frac{1}{128}$ of the total seams during the resizing.

Bibliography

[1] Photodna. http://www.microsoftphotodna.com/.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned Salient Region Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597 – 1604, 2009.

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. SÃijsstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274 – 2282, 2012.

[4] R. Achanta and S. Süsstrunk. Saliency detection for content-aware image resizing. In *Proceedings of the 16th IEEE International Conference on Image Processing*, pages 1001–1004, 2009.

[5] A. Adams, E.-V. Talvala, S. H. Park, D. E. Jacobs, B. Ajdin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek, M. Tico, H. P. A. Lensch, W. Matusik, K. Pulli, M. Horowitz, and M. Levoy. The frankencamera: An experimental platform for computational photography. *ACM Transaction on Graphics*, 29(4):1–12, 2010.

[6] Y. Amit and D. G. Y. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[7] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, pages 3378–3385. IEEE, 2012.

[8] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transaction on Graphics*, 26(3):1–10, 2007.

[9] S. Battiato, A. R. Bruna, G. Messina, and G. Puglisi, editors. *Image Processing for Embedded Devices*. Bentham Science Publisher, 2010.

[10] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravì. Exploiting textons distributions on spatial hierarchy for scene classification. *Eurasip Journal on Image and Video Processing*, pages 1–13, 2010.

[11] S. Battiato, G. M. Farinella, N. Grippaldi, and G. Puglisi. Content-based image resizing on mobile devices. In *International Conference on Computer Vision Theory and Applications*, pages 87–90, 2012.

[12] S. Battiato, G. M. Farinella, M. Guarnera, G. Messina, and D. Ravì. *A cluster-based boosting strategy for red-eyes removal.* Springer, 2012.

[13] S. Battiato, G. M. Farinella, E. Messina, G. Puglisi, D. Ravì, A. Capra, and V. Tomaselli. On the performances of computer vision algorithms on mobile platforms. In *SPIE Electronic Imaging - Digital Photography VIII*, 2012.

[14] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì. Computer vision on mobile devices: A few case studies. *STDAY*, 2011.

[15] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì. Content-aware image resizing with seam selection based on gradient vector flow. In *International Conference on Image Processing (ICIP)*, pages 1–4, 2012.

[16] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì. Saliency based selection of gradient vector flow paths for content aware image resizing (submitted). *IEEE Transactions on Image Processing*, 2013.

[17] S. Battiato, G. Farinella, G. Giuffrida, C. Sismeiro, and G. Tribulato. Exploiting visual and text features for direct marketing learning in time and space constrained domains. *Pattern Analysis and Applications*, 2009. http://dx.doi.org/10.1007/s10044-009-0145-2.

[18] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravì. Scene categorization using bag of textons on spatial hierarchy. In *IEEE International Conference on Image Processing (ICIP-08)*, pages 2536–2539, 2008.

[19] S. Battiato, G. M. Farinella, G. Giuffrida, C. Sismeiro, and G. Tribulato. Using visual and text features for direct marketing on multimedia messaging services domain. *Multimedia Tools Applications*, 42(1):5–30, 2009.

[20] S. Battiato, G. M. Farinella, G. C. Guarnera, T. Meccio, G. Puglisi, D. Ravì, and R. Rizzo. Bags of phrases with codebooks alignment for near duplicate image detection. In *Proceedings of the 2nd ACM workshop on Multimedia in forensics, security and intelligence*, MiFor '10, pages 65–70, New York, NY, USA, 2010. ACM.

[21] S. Battiato, G. M. Farinella, E. Messina, and G. Puglisi. Robust image alignment for tampering detection. *IEEE Transactions on Information Forensics and Security*, 7(4):1105–1117, 2012.

[22] S. Battiato, G. Farinella, M. Guarnera, D. Ravì, and V. Tomaselli. Instant scene recognition on mobile platform. In *European Conference on Computer Vision (ECCV) - Workshops and Demonstrations*, volume 7585 of *Lecture Notes in Computer Science*, pages 655–658, 2012.

[23] S. Battiato, G. Farinella, G. Puglisi, and D. Ravì. Aligning codebooks for near duplicate image detection. *Multimedia Tools and Applications*, pages 1–24, 2013.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[25] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, April 2002.

[26] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Improving color constancy using indoor-outdoor image classification. *IEEE Transactions on Image Processing*, 17(12):2381–2392, 2008.

[27] I. Biederman. Aspects and extension of a theory of human image understanding. In *Computational Processes in Human Vision: An Interdisciplinary Perspective*, 1988.

[28] X. Boix, J. M. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. S. Gual, and J. Gonzàlez. Harmony potentials - fusing global and local scale for semantic image segmentation. *International Journal of Computer Vision*, 96(1):83–102, 2012.

[29] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *European Conference on Computer Vision (ECCV-06)*, pages 517–530, 2006.

[30] A. Bosch, X. Muñoz, and R. Martí. Review: Which is the best way to organize/classify images by content? *Image Vision Computing*, 25(6):778–791, 2007.

[31] L. D. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV'09*, pages 1365–1372, 2009.

[32] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1, 2001.

[33] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1984.

[34] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recogn. Lett.*, 30:88–97, January 2009.

[35] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 44–57, Berlin, Heidelberg, 2008. Springer-Verlag.

[36] J. a. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *Int. J. Comput. Vision*, 98(3):243–262, July 2012.

[37] C.-C. Chang, J.-C. Chuang, and Y.-S. Hu. Retrieving digital images from a {JPEG} compressed image database. *Image and Vision Computing*, 22(6):471 – 484, 2004.

[38] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *IEEE International Conference on Computer Vision*, pages 914–921, 2011.

[39] K. Chatfield, V. Lemtexpitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[40] X. Cheng, Y. Hu, and L.-T. Chia. Exploiting local dependencies with spatial-scale space (s-cube) for near-duplicate retrieval. *Comput. Vis. Image Underst.*, 115(6):750–758, June 2011.

[41] T. Cho, M. Butman, S. Avidan, and W. Freeman. The patch transform and its applications to image editing. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[42] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference*, 2008.

[43] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, pages 17–24. IEEE, 2009.

[44] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. In *Proceedings of the 2010 international MICCAI conference on Medical computer vision: recognition techniques and applications in medical imaging*, MCV'10, pages 106–117, Berlin, Heidelberg, 2011. Springer-Verlag.

[45] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[46] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[47] R. de Oliveira, M. Cherubini, and N. Oliver. Looking at near-duplicate videos from a human-centric perspective. *TOMCCAP*, 6(3), 2010.

[48] D.Eastlake and P.Jones. RFC 3174. http://tools.ietf.org/html/rfc3174.

[49] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1964–1971, Washington, DC, USA, 2006. IEEE Computer Society.

[50] M. Douze, H. Jégou, S. Harsimrat, L. Amsaleg, and C. Schmid. Evaluation of GIST descriptors for web-scale image search. In *International Conference on Image and Video Retrieval*, Santorini, Greece, 2009.

[51] J. D. Eggerton. Statistical distributions of image DCT coefficients. *Computers & Electrical Engineering*, 12:137–145, 1986.

[52] I. Endres and D. Hoiem. Category independent object proposals. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 575–588, Berlin, Heidelberg, 2010. Springer-Verlag.

[53] T. Eude, R. Grisel, H. Cherifi, and R. Debrie. On the distribution of the DCT coefficients. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 365–368, 1994.

[54] Y. Fang, Z. Chen, W. Lin, and C.-W. Lin. Saliency detection in the compressed domain for adaptive image retargeting. *IEEE Transactions on Image Processing*, 21(9):3888 –3901, sept. 2012.

[55] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*. icml.cc / Omnipress, 2012.

[56] G. M. Farinella and S. Battiato. Scene classification in compressed and constrained domain. *Computer Vision, IET*, 5(5):320 –334, 2011.

[57] G. M. Farinella and D. Ravì. *Image Categorization*, chapter Chapter in Image Processing for Embedded Devices, Applied Digital Imaging ebook series -. Bentham Science Publisher, 2010.

[58] G. M. Farinella, D. Ravì, V. Tomaselli, M. Guarnera, and S. Battiato. Representing scenes for real-time context classification on mobile devices (submitted). *Pattern Recognition*, 2013.

[59] G. M. Farinella, S. Battiato, G. Gallo, and R. Cipolla. Natural versus artificial scene classification by ordering discrete Fourier power spectra. In *International Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR & SPR - 08)*, volume 5342 of Lecture Notes in Computer Science, pages 137–146. Springer-Verlag, 2008.

[60] 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*, 32(9):1627–1645, 2010.

[61] http://www.flickr.com/.

[62] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, September 1991.

[63] R. Gallea, E. Ardizzone, and R. Pirrone. Real-time content-aware resizing using reduced linear model. In *IEEE International Conference on Image Processing*, pages 2813–2816, 2010.

[64] F. Garage. Fcam api, 2012. http://fcam.garage.maemo.org/.

[65] K. R. Georgios Floros and B. Leibe. Multi-class image labeling with top-down segmentation and generalized robust p^n potentials. In *Proceedings of the British Machine Vision Conference*, pages 79.1–79.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.79.

[66] H. Grabner, F. Nater, M. Druey, and L. V. Gool. Visual interestingness in image sequences. In *ACM International Conference on Multimedia*, 2013.

[67] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision (ICCV-05)*, pages 1458–1465, Washington, DC, USA, 2005. IEEE Computer Society.

[68] G.Salton and C.Buckley. ImageMagick. 1988. http://www.imagemagick.org.

[69] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, pages 1030–1037, 2009.

[70] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang. Unsupervised extraction of visual attention objects in color images. *IEEE Transaction on Circuits and Systems for Video Technology*, 16(1):141–145, 2006.

[71] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 545–552. MIT Press, Cambridge, MA, 2007.

[72] T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society.

[73] 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*, 11(8):1434–1445, 2009.

[74] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 762–768, 1997.

[75] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, MIR '08, pages 39–43, New York, NY, USA, 2008. ACM.

[76] L. Itti and C. Koch. A comparison of feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10:161–169, 1999.

[77] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[78] R. Ji, L.-Y. Duan, J. C. 0006, L. Xie, H. Yao, and W. Gao. Learning to distribute vocabulary indexing for scalable visual search. *IEEE Transactions on Multimedia*, 15(1):153–166, 2013.

[79] R. Ji, H. Yao, W. Liu, X. Sun, and Q. Tian. Task-dependent visual-codebook compression. *IEEE Transactions on Image Processing*, 21(4):2282–2293, 2012.

[80] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433 – 449, May 1999.

[81] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, November 1987.

[82] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and subimage retrieval. In *In ACM Multimedia*, pages 869–876, 2004.

[83] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219 – 227, 1985.

[84] J. J. Koenderink and A. J. van Doom. Representation of local geometry in the visual system. *Biol. Cybern.*, 55(6):367–375, March 1987.

[85] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj cut. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 18–25, Washington, DC, USA, 2005. IEEE Computer Society.

[86] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 239–253, Berlin, Heidelberg, 2010. Springer-Verlag.

[87] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where and how many? combining object detectors and crfs. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 424–437, Berlin, Heidelberg, 2010. Springer-Verlag.

[88] P. Ladret and A. Guérin-Dugué. Categorisation and retrieval of scene photographs from JPEG compressed database. *Pattern Analysis & Application*, 4:185–199, June 2001.

[89] E. Y. Lam and J. W. Goodman. A mathematical analysis of the DCT coefficient distributions for images. *IEEE Transactions on Image Processing*, 9(10):1661–1666, 2000.

[90] E. Lam. Analysis of the dct coefficient distributions for document coding. *IEEE Signal Processing Letters*, 11(2):97–100, 2004.

[91] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR-06)*, pages 2169–2178, 2006.

[92] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(7):1294–1309, July 2009.

[93] H. Lejsek, H. Þormóðsdóttir, F. Ásmundsson, K. Daðason, r. r. Jóhannsson, B. r. Jónsson, and L. Amsaleg. Videntifier" forensic: large-scale video identification in practice. In *Proceedings of the 2nd ACM workshop on Multimedia in forensics, security and intelligence*, MiFor '10, pages 1–6, New York, NY, USA, 2010. ACM.

[94] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44, June 2001.

[95] T. K. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *ICCV*, pages 1010–1017, 1999.

[96] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[97] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2714–2721, 2013.

[98] J. Luo and M. R. Boutell. Natural scene classification using overcomplete ICA. *Pattern Recognition*, 38(10):1507–1519, 2005.

[99] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, June 2008.

[100] O. Marques, L. M. Mayron, G. B. Borba, and H. R. Gamba. On the potential of incorporating knowledge of human visual attention into CBIR systems. In *IEEE International Conference on Multimedia and Expo*, pages 773–776, 2006.

[101] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

[102] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press, 2002. doi:10.5244/C.16.36.

[103] G. Messina, M. Guarnera, G. M. Farinella, and D. Ravì. Method and apparatus for filtering red and/or golden eye artifact. *United States Publication Number US20110158511A1*, 2009.

[104] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, October 2004.

[105] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[106] M. M. Mokji and S. A. R. A. Bakar. Gray level co-occurrence matrix computation based on haar wavelet. In *Proceedings of the Computer Graphics, Imaging and Visualisation*, CGIV '07, pages 273–279, Washington, DC, USA, 2007. IEEE Computer Society.

[107] F. Müller. Distribution shape of two-dimensional DCT coefficients of natural images. *Electronics Letters*, 29:1935–1936, 1993.

[108] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[109] R. M. Norton. The double exponential distribution: Using calculus to find a maximum likelihood estimator. *The American Statistician*, 38(2):135–136, 1984.

[110] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, July 2002.

[111] A. Oliva and A. Torralba. http://people.csail.mit.edu/torralba/code/spatialenvelope/, 2001.

[112] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[113] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research*, 155:251–256, 2006.

[114] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[115] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[116] W. K. Pratt. *Digital Image Processing*. John Wiley & Sons, Inc., New York, NY, USA, 1978.

[117] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift - map image editing. In *International Conference on Computer Vision (ICCV)*, pages 151–158, 2009.

[118] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 10–, Washington, DC, USA, 2003. IEEE Computer Society.

[119] L. W. Renninger and J. Malik. When is scene recognition just texture recognition? *Vision Research*, 44:2301–2311, 2004.

[120] R. Rivest. RFC 1321. http://tools.ietf.org/html/rfc132.

[121] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[122] M. Rubinstein, A. Shamir, and S. Avidan. Multi-operator media retargeting. *ACM Transaction on Graphics*, 28(3):23:1–23:11, 2009.

[123] A. Saffari and H. Bischof. Clustering in a boosting framework, 2007.

[124] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *INFORMATION PROCESSING AND MANAGEMENT*, pages 513–523, 1988.

[125] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[126] G. Schaefer. Content-based image retrieval: Advanced topics. In T. Czachórski, S. Kozielski, and U. Stańczyk, editors, *Man-Machine Interactions 2*, volume 103 of *Advances in Intelligent and Soft Computing*, pages 31–37. Springer Berlin Heidelberg, 2011.

[127] R. E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5(2):197–227, July 1990.

[128] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, June 2008.

[129] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *In ECCV*, pages 1–15, 2006.

[130] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the International Conference on Computer Vision*, 2005.

[131] S. Smoot and L. A. Rowe. Study of DCT coefficient distributions. In *SPIE Symposium on Electronic Imaging*, volume 2657, pages 403–411, 1996.

[132] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *Proceedings of the British Machine Vision Conference*, pages 62.1–62.11. BMVA Press, 2009. doi:10.5244/C.23.62.

[133] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[134] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[135] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 338(6582):520–522, June 1996.

[136] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, pages 3001–3008, 2013.

[137] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, 101(2):329–349, 2013.

[138] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[139] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *IEEE Internation Conference on Computer Vision (ICCV-03)*, pages 273–280, 2003.

[140] A. Torralba and A. Oliva. Semantic organization of scenes using discriminant structural templates. In *IEEE International Conference on Computer Vision (ICCV-99)*, pages 1253–1258, 1999.

[141] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computing in Neural Systems*, 14:391–412, 2003.

[142] A. Torralba and S. Pawan. Statistical context priming for object detection. In *IEEE Internation Conference on Computer Vision*, 2001.

[143] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1226–1238, 2002.

[144] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78(1-2):507 – 545, 1995.

[145] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2010.

[146] M. Varma and R. Garg. Locally invariant fractal features for statistical texture classification. In *ICCV*, pages 1–8, 2007.

[147] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[148] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[149] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[150] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, April 2007.

[151] J. Vogel, A. Schwaninger, C. Wallraven, and H. H. Bülthoff. Categorization of natural scenes: Local versus global information and the role of color. *ACM Transactions on Applied Perception*, 4(3):19, 2007.

[152] X. W.-L.Zhao and C.-W. Ngo. Sotu: A toolkit for efficient near-duplicate image/video & retrieval/detection. *Manual for SOTU Version 1.06*, 2011.

[153] G. K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):18–34, 1991.

[154] D. Walther, U. Rutishauser, C. Koch, and P. Perona. On the usefulness of attention for object recognition. In *ECCV Workshop on Attention and Performance in Computational Vision*, pages 96–103, 2004.

[155] Y. Wang, Z. Hou, and K. Leman. Keypoint-based near-duplicate images detection using affine invariant feature and color matching. In *ICASSP*, pages 1209–1212. IEEE, 2011.

[156] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 0:25–32, 2009.

[157] C. Xu and L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[158] D. Xu, T. J. Cham, S. Yan, L. Duan, and S.-F. Chang. Near duplicate identification with spatially aligned pyramid matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 20:1068–1079, 2010.

[159] D. Xu and S.-F. Chang. Video event recognition using kernel methods with multilevel temporal alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1985–1997, 2008.

[160] G. S. Yovanof and S. Liu. Statistical analysis of the DCT coefficients and their quantization. In *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers*, 1996.

[161] G. Yu, G. Sapiro, and S. Mallat. Image modeling and enhancement via structured sparse model selection. In *IEEE International Conference on Image Processing (ICIP)*, pages 1641–1644, 2010.

[162] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 708–721, Berlin, Heidelberg, 2010. Springer-Verlag.

[163] D. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In H. Schulzrinne, N. Dimitrova, M. A. Sasse, S. B. Moon, and R. Lienhart, editors, *ACM Multimedia*, pages 877–884. ACM, 2004.

[164] W.-L. Zhao and C.-W. Ngo. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. *Trans. Img. Proc.*, 18(2):412–423, February 2009.

[165] W.-L. Zhao, X. Wu, and C.-W. Ngo. On the annotation of web videos by efficient Near-Duplicate search. *IEEE Transactions on Multimedia*, 12(5):448–461, August 2010.

[166] W. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Transactions on Multimedia*, 9(5):1037–1048, 2007.

[167] J. Zhu, S. C. Hoi, M. R. Lyu, and S. Yan. Near-duplicate keyframe retrieval by nonrigid image matching. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, pages 41–50, New York, NY, USA, 2008. ACM.