Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 00 (2015) 1–25

Pattern Recognition

Semantic Segmentation of Images Exploiting DCT Based Features and Random Forest

D. Ravì[a], M. Bober[b], G. M. Farinella[a], M. Guarnera[c], S. Battiato[a]

[a] *Image Processing Laboratory, Dipartimento di Matematica e Informatica, University of Catania, Italy*
[b] *Center for Vision, Speech and Signal Processing, University of Surrey, Uk*
[c] *Advanced System Technology - Computer Vision, STMicroelectronics, Catania, Italy*

Abstract

This paper presents an approach for generating class-specific image segmentation. We introduce two novel features that use the quantized data of the Discrete Cosine Transform (DCT) in a Semantic Texton Forest based framework (STF), by combining together colour and texture information for semantic segmentation purpose. 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 based features describe complex textures represented in the frequency domain and not just simple textures obtained using differences between intensity of pixels as in the classic STF approach. Differently than existing methods (e.g., filter bank, etc.) just a limited amount of resources is required. The proposed method has been tested on two popular databases: CamVid and MSRC-v2. Comparison with respect to recent state-of-the-art methods shows improvement in terms of semantic segmentation accuracy.

Keywords:
Semantic Segmentation, Random Forest, DCT, Textons

1. Introduction and motivations

Nowadays a wide range of applications including medical, robotics and automotive, require the ability to automatically understand the real world. Examples of these applications are a smart cars able to recognize and eventually help a careless driver, to detect a pedestrian crossing the street. Another example is a smart system that during a surgery operation is able to drive the surgeon on the localization of the tumour area and steer him in the removal process of that area. Last but not least a surveillance system that can analyze and recognize automatically what is going in the world from a recorded video. Electronic devices with the ability to understand the real world from images are called intelligent systems with semantic segmentation. The semantic segmentation, can be thought as an extension of the popular scene classification problem

Email addresses: ravi@dmi.unict.it (D. Ravì), m.bober@surrey.ac.uk (M. Bober), gfarinella@dmi.unict.it (G. M. Farinella), mirko.guarnera@st.com (M. Guarnera), battiato@dmi.unict.it (S. Battiato)

where the entity to classify is not anymore the whole image but single group of pixels [1]. It aims at pixel-wise classification of images according to semantically meaningful regions (e.g., objects). A precise automated image segmentation is still a challenging and an open problem. Among others, 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 images into regions [2, 3]. 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 supports rotation and scale invariance. When spatial organization is required a second order statistics can be used. An example is image correlograms [4] that 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 and allows to discriminate different objects. Finally texture is a visual cue that describes the luminosity fluctuations in the image, which let us interpret a surface as a whole part. Textures can be characterized using properties such as regularity, coarseness, contrast and directionality and contain also 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) [5], Gabor Filter [6], and Local Binary Pattern (LBP) [7] are examples of popular methods to extract texture features. Other methods to obtain texture features are the fractals representation [8] and Textons [9].

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, people, etc. The following criteria should be taken into account while considering the design of a system and the related features extraction method:

- Similar low-level features response can represent different objects as part of objects. Each single feature cannot be hence adequate for segmenting, in a discriminative way, the object that they belong to. A spatial arrangement of low-level features increases the object discrimination.

- A semantic segmentation approach cannot be generic because is strongly related to the involved application both in terms of requirements and input data types. Some examples of different domains include the segmentation of images obtained from fluorescence microscope, video surveillance cameras and photo albums. Another important parameter that is application dependent is for example the detail coarseness of required segmentation.

- 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 [10].

- The hardware miniaturization has reached impressive levels of performance stimulating the deployment of new devices such as smart-phones 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 related features should be designed to ensure good performance for computationally limited devices [11].

The first contribution of this paper is the design of new texture features pipeline, which combine colour and texture clues in more efficient manner with respect to other methods in literature (e.g., convolutional network). Secondly, we propose texture features based on DCT coefficients selected through a greedy fashion approach and suitably quantized. These DCT features have been exploited in [12] and successfully applied for the scene classification task making use of their capability to describe complex textures in the frequency domain maintaining a low complexity. Other approaches usually compute similar features using bank of filter responses that drastically increases the execution time. As in [12] our texture information is extracted using the DCT module that is usually integrated within the digital signal encoder (JPEG or MPEG based). The proposed features are then used to feed a Semantic Texton Forest [13] that has been showed to be a valid baseline approach for the semantic segmentation task.

The rest of the paper is organized as follows: Section 2 discusses the state-of-the-art approaches, whereas Section 3 describes the random forest algorithm and how to add the novel features in the STF system. Section 4 presents the pipeline of the proposed approach. Section 5 introduces the extraction pipelines for each proposed features. Section 6 describes the experimental settings and the results. Finally, Section 7 concludes the paper.

2. Related works

To address the challenges described above, different segmentation methods were proposed in literature. Some basic approaches segment and classify each pixel in the image using a region-based methodology as in [14, 15, 16, 17, 18, 19, 20, 21, 22]. Other approaches use a multiscale scanning window detector such as Viola-Jones [23] or Dalal-Triggs [24], possibly augmented with part detectors as in Felszenszwalb et al. [25] or Bourdev et al. [26]. More complex approaches as in [27, 28] unify these paradigms into a single recognition architecture, and leverage on their strengths by designing region-based specific object detectors and combining their outputs. By referring to the property that the final label of each pixel can

be dependent by the labels assigned to other pixels in the image, different methods use probabilistic models such as the Markov Random Field (MRF) and the Conditional Random Fields (CRF) that are suitable to address label dependencies. As example, the nonparametric model proposed in [29] requires no training and 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 [30], 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 for training a randomized decision forest. The formulation integrates multiple features in the MRF framework to segment and recognize different object classes. The results of this work highlight a strong dependency of accuracy from the density of the 3D features. In the TextonBoost technique [31] the segmentation is obtained by implementing a CRF and features that automatically learn layout and context information. Similar features were also proposed in [32], although Textons were not used, and responses were not aggregated over a spatial region. In contrast with these techniques, the shape context technique in [14] uses a hand-picked descriptor. In [33] 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 prior knowledge. An extended set of appearance-based features is used, which consists of Textons, colour, location and Histogram of Gradients (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 [34] 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 [34] 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 [35] 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

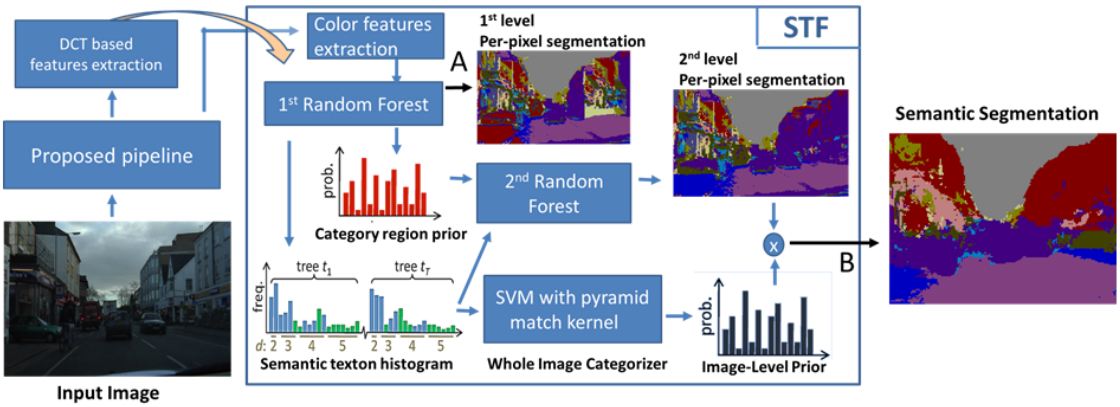


Figure 1. Integration of texture features in the STF system.

segmentation problem in terms of performance, they represent a bottleneck in the computation, because the inference is a highly resources consuming process. A powerful approach with good performance, while preserving high efficiency, is based on the random forest. For example in [13], the authors show that one can build rich Texton codebooks without computing expensive filter-banks or descriptors, and without performing costly k-means clustering and nearest-neighbour assignments. Specifically, the authors propose the bag of Semantic Textons that is an extension of the Bag of Word Model [36] obtained by combining a histogram of the hierarchical visual word with a region prior category. In this paper we build on the semantic texton forest (STF) approach (detailed in Section 3) by proposing a new semantic segmentation pipeline which exploits features extracted on the DCT domain. The exploitation of contextual and structural information have been recently proposed in [37, 38] to improve the performances of random forests for semantic image labelling. In particular, the random forest approach has been augmented in order to consider topological distribution of object classes in a given image. A novel splitting function has been also introduced to allow the random forest to work with the structured label space. Recent trends also consider Convolutional Neural Network for the semantic segmentation. In [39] the authors adapt state-of-the-art classification networks (i.e., AlexNet, the VGG net, and GoogLeNet) into fully convolutional networks and transfer their learned representations to the segmentation task. A novel architecture that combines semantic information of the different layers is proposed to produce the final semantic segmentation of images.

3. Random forests and Semantic Texton Forest

Before presenting our approach, we briefly review the randomized decision forest algorithms [40]. Random forests are an ensemble of separately trained binary decision trees. Decision trees are trained to solve the classification problem separately and the results are predicted combining all the partial results obtained by each tree. This process leads to a significantly better generalization and avoids overfitting 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 [41] and later more and more computer vision applications used an ensemble of randomly trained decision trees. Complex computer vision tasks exploiting random forests were presented in [42, 43, 44] for a shape classification system, automatic handwriting recognition and medical imaging. A Random Forest can solve different problems like predict class label, estimate value of a continuous variable, learn probability density function and manifold. The Random Forest uses weak classifiers in each node of the trees to solve the classification or regression problem. A weak classifier (called decision stump) is specialized on a sub problem and is significantly faster compared to a strong classifier (e.g., SVM [45]), which is usually designed to tackle complex problems. Every Random Forest can be described by the number T of the trees used, the maximum depth D and the type of weak learner model used in each node. The STF model is a complex system that ensembles 2 randomized decision forests in cascade. The randomized decision forests obtains semantic segmentation acting directly on image pixels with simple features (e.g., differences between pixels) and therefore do not need the expensive computation of filter-bank responses or local descriptors. They are extremely fast for both training and testing. Specifically, the first randomized decision forest in the STF uses only simple pixel comparisons on local image patches of size $d \times d$ pixels. The split function f_1 in this first forest can directly take the pixel value $p(x, y, b)$ at pixel location (x, y) in the colour channel b or computes some other functions defined on two different locations $p_1(x_1, y_1, b_1)$ and $p_2(x_2, y_2, b_2)$ selected 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 $P(c|r) = \frac{1}{|r|} \sum_{i=1}^{|r|} P(c|L_i)$ (see the STF block in Fig. 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 achieve an efficient and accurate segmentation. Specifically, the split node functions f_2 of the second forest evaluate either the numbers $H_{r+1}(n')$ of a generic semantic Textons n' or the probability $P(c|r + i)$ within a translated rectangle r relative to the i_{th} pixel that we want to classify. The categorization module determines 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:

$$P'(c|i) = \frac{1}{Z} P(c|i) P(c)^a \tag{1}$$

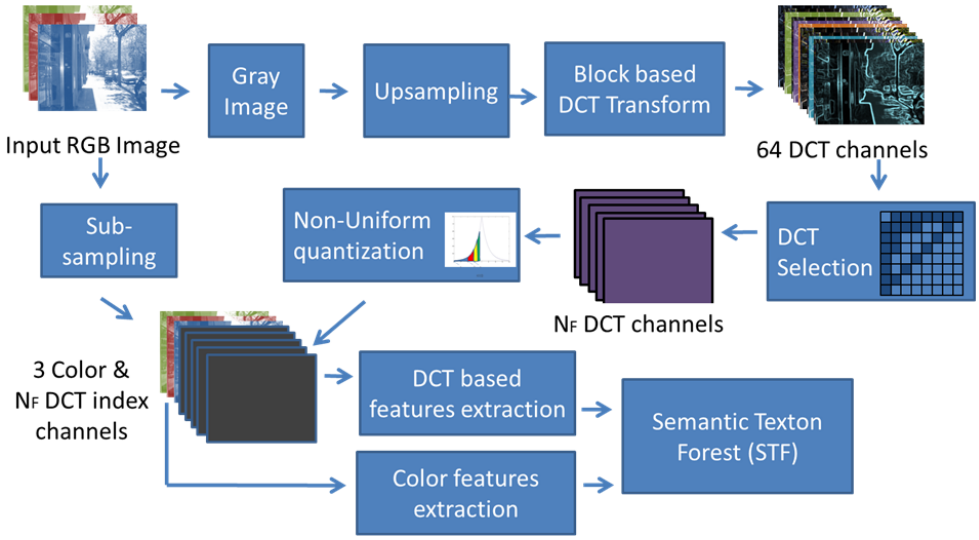


Figure 2. Pipeline of the proposed approach. For further details on the Semantic Texton Forest block refer to Fig. 1.

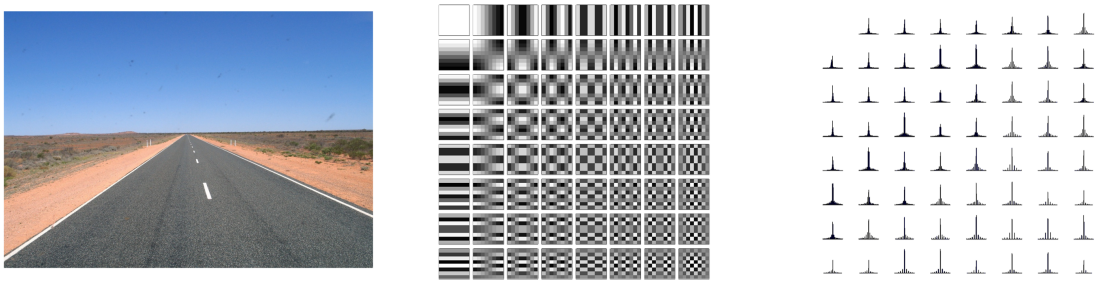
using parameter a to soften the prior and where $\frac{1}{Z}$ is a normalization constant such that $P'(c|i)$ sum up to one. As previous mentioned, our approach combines texture and colour clues within a STF (see Fig. 1). Adding the texture features in the first random forest allows us either to catch the semantic segmentation output after performing entirely the STF system (point B in the Fig. 1) or after perform just the first random forest (point A in the Fig. 1). The last solution is preferred for real time applications, when the execution time is crucial with respect to the accuracy. In Section 6, we show that including the proposed DCT features, the accuracy increases in both the semantic segmentation steps.

4. Proposed approach

The workflow of our method is shown in Fig. 2. Each image is first converted into a grayscale channel and then upsampled. A 8x8 block based DCT transformation is applied and just the most N_F discriminative DCT coefficients are selected (generating N_F different DCT channels). The DCT data are then quantized using a non-uniform clustering. The quantization extracts N_F new DCT index channels that will be aggregated with a subsampled version of the colour data. The 3 colour channels and the N_F DCT index channels are finally used to generate suitable colour and texture features for each node of the decision random forest in the STF system. Next Section explains the functionality of each block of the system, whereas Section 5 describes the detail of the "DCT based features extraction" block.

4.1. DCT transform and DCT frequencies selection

One of the most popular standard for lossy compression of images is JPEG [46]. JPEG is an hardware/software codec engine present in all the consumer devices such as digital cameras, smartphones, etc.



(a) (b) (c)

Figure 3. Laplace distributions of DCT coefficients for natural images. Left: image under consideration; Middle: the 64 basis related to the 8×8 DCT transformation; Right: the different DCT distributions related to the 64 DCT basis reported in the middle, obtained considering the image at left.

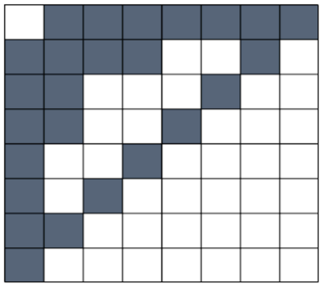
Moreover, the great majority of the images on Internet are stored in JPEG format. DCT features that can be extracted directly in the compressed domain reducing the features extraction cost. These are desirable features for the image segmentation engine. The JPEG algorithm divides the image into non-overlapping blocks of size 8×8 pixels, then 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. The coefficient that scales the constant basis function of the DCT is called the DC coefficient, while the other coefficients are called AC coefficients. Different statistical models for AC coefficients were proposed including Gaussian [47], Cauchy, generalized Gaussian and sum of Gaussian distributions [48, 49, 50, 51, 52]. 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 resembles the Laplace distribution (See Fig. 3). This guess has been demonstrated through a rigorous mathematical analysis in [53, 54]. The probability density function of a Laplace distribution can be written as:

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

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

$$\hat{b} = \frac{1}{N} \sum_{i=1}^{N} |x_i - \hat{\mu}| \tag{3}$$

A recent work [12] describes how to use these parameters to classify the scene in real time. In Section 4.2, instead, we show how to use the Laplace distribution 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



(a)

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

(b)

Figure 4. Left: Schema used to select the DCT frequencies. Right: Standard JPEG quantization table.

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(b)), whereas the others are more related to the reconstruction of the textured blocks. Moreover, high frequencies are usually affected by noise and could be not useful to 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 reduce 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 resources) is the one which considers the AC DCT components related the DCT basis of Fig. 4(a). According to this schema only 25 frequencies out of 64 are selected to compute features useful for the segmentation. We will refer to this set of frequencies as F and the related cardinality as N_F (see [12] for more details on the frequency selection) .

4.2. Quantization

Two important observations regarding the DCT data should be taken into account when these data are used as features. The first one has been disclosed in the previous paragraph: the DCT data can be summarized by Laplace distributions. The second one states that, in the real world, the human vision is more sensitive to some frequencies rather than others [55, 56, 57].

These observations convey the fact that before using the DCT data, they need to be properly processed. In our process, to take into account the HVS (human vision system) sensitivity to the different frequencies, we replace the uniform random function used to select the features in each node of the 1st random forest (see STF block in Fig. 1), with a probability selection function P_{DCT}. The P_{DCT} steers the learning process towards using more frequently the DCT coefficients that are more important for the human vision system. For this purpose we exploit the standard quantization Table (Fig. 4(b)) used in the JPEG compression [55]. This table has been developed to achieve good image compression and avoiding visible distortions. Due to

variations in the contrast sensitivity of the human visual system as a function of spatial frequency, different DCT coefficients are quantized with different factors [58]. More specifically, the standard JPEG suggests to use these perceptual criteria to quantize the corresponding DCT coefficients amplitudes that cause perceptible differences to a human observer. Equation (4) allows us to convert each quantization value into a selection probability that is high for the most important frequencies (i.e., low values in the quantization table) and low for the frequencies that are less important (i.e., high values in the quantization table) satisfying in this way our modelling. The standard quantization table is hence transformed in a probability table that we refer with the symbol P_{DCT} (see Table 1). Each element $P_{DCT}(i)$ in this table is formally defined as follows:

$$P_{DCT}(i) = \begin{cases} \frac{\frac{1}{q_i}}{\sum_{j \in F} \frac{1}{q_j}} & \text{if } i \in F \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where q_i, q_j are quantization values of the standard JPEG quantization table (Fig. 4(b)), and F is the set of selected DCT coefficients (see Section 4.1). These priors are used in the learning process to increase 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 [13].

Table 1. Probability table P_{DCT} 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
0.048	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

In order to cater our first observation stating that DCT data can be summarized by Laplace distributions, we propose a quantization step that is capable to generate more centroids in the DCT space where the data distribution is more dense (all the value that are near to the center of the Laplace distribution) and less in the areas where only a few DCT data fall in. The aim is to produce centroids that follow the natural distribution of the considered DCT data. Usually K-means is used to quantize the features space. The centroids provided by K-means codebooks are highly dependent on the sample distribution in the space, degrading as this becomes more non-uniform. K-means works well for data containing only uniform distribution since in the non-uniform case that K-means devotes most of its centres to the immediate neighborhood of the central peak, and the coding suffers [59]. The non-uniformity is essentially due to two effects: (i) certain data occur far more frequently than others; and (ii) the amount of any given feature that appears is extremely

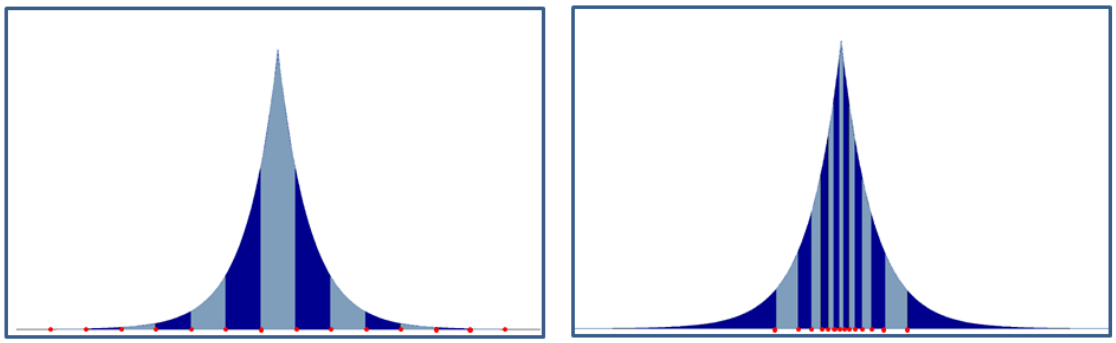


Figure 5. Laplace distribution representing a given AC frequency. The Laplace distribution is clustered in two different ways. In the left the centroids, represented by the red points, are obtained using a uniform quantization. In the right the centroids are instead obtained with the proposed analytic solution that takes into account the non-uniformity distribution of the data.

variable due to the multi-scale region structure of natural scenes. K-means centres drift towards high density regions owing to the "mean shift like" K-means update rule [60]. Asymptotically, they are distributed according to the underlying density (i.e., in the same way as random samples). This point is critical because the data that are most informative for classification tend to have intermediate frequencies. Over-frequent patches typically contain generic image structures like edges that occur often in all classes, giving little discriminant information. Conversely, rare patches may be highly discriminant when they occur, but they occur so seldom that they have little influence on overall performance. These effects are probably even greater in man-made environments, where large uniform uninformative regions are common. A clustering that takes into account the non-uniformity property of the data is essential to quantize the DCT space. 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 Laplace parameters (median and slope of each DCT coefficient). The cluster centroids are then computed performing integration on the area of each Laplace distribution. The points that divide this area in k equal spaces are the proposed quantization points (Fig. 5).

This process is repeated for all the N_F DCT coefficients, separately producing a vocabulary table with k x N_F entries. In each column of this table, the values are arranged in an ascending order that is important during the clustering process since it allows to implement an efficient stopping criteria. This vocabulary table is used to quantize the DCT channels and produce for each DCT value a corresponding DCT index value required in Section 5 to generate the proposed features. The new DCT index channels represent in a suitable way the visual data as discuss so far and they have also the advantage that can be stored in memory just using few bits per pixel. Comparison results using different number of clustering are provided in Section 6.

4.3. Upsampling and subsampling

The design of the proposed pipeline is aimed to be as generic as possible and capable to efficiently segment any image regardless of its resolution. In order to obtain this capability, we propose to process a subsampled version of the image. The only consequence of using the subsampled image is a less precise segmentation boundaries output. On the other hand, one should also consider that the DCT data are obtained by a block based process that produces not pixel specific information. For this reason, to obtain DCT information required for each subsampled pixels we need to use an enlarged version of the image as input of the DCT transformation block. The enlarged version can be obtained either using an interpolation process or can be already available if applied on a multi-resolution sensor. The relation that links the upsampling factor U_s and the subsampling factor S_s with the DCT block size S_{DCT_block} is described by the following equation:

$$U_s * S_s = S_{DCT\_block} \tag{5}$$

When the subsampling factor S_s is equal to the size of the DCT block S_{DCT_block} no enlarging process is required before the DCT transformation block. At the end of this process, colour and texture data are available and ready to generate features through the colour and the DCT based features extraction blocks (see Fig. 2).

5. DCT-based features extraction

We propose two novel DCT based features which are computed after the quantization step detailed in Section 4.2. The first one, that we call feature f_1, is aimed to capture the different textures distribution for each image region. The feature f_1 is defined as a tuple $[r(x, y, h, w, t), s]$ where r is an image region with size $h \times w$ associated to the i_{th} pixel in the DCT layer t, and s is the quantization index used for the statistical evaluation. The vector of coordinates (x, y) indicates the offset of the considered region with respect to the i_{th} pixel to be classified. A set R of candidate rectangle regions are chosen at random, such that their top-left and bottom-right corners lie within a fixed bounding box B_1. The details about the extraction process used to obtain the feature f_1 are shown in Fig. 6. Specifically, the 3D array with N_F 2D maps of size $h \times w$ is extracted with an offset vector (x, y) respect to the i_{th} pixel to be segmented (the i_{th} pixel is represented with the red cross in Fig. 6). In the step 3) of Fig. 6, one of the N_F available DCT index layers, is selected using the probability table P_{DCT} defined in Section 4.2, whereas in the step 4) and 5) a statistical measurement is performed on the selected region r. By fixing the value s as one of the index of the quantization process (selected randomly when the features are generate) and using the region r, we propose the three following measurements:

$$Stat_1(r, s) = \frac{\sum_{c \in r} |c - s|}{|r|} \tag{6}$$

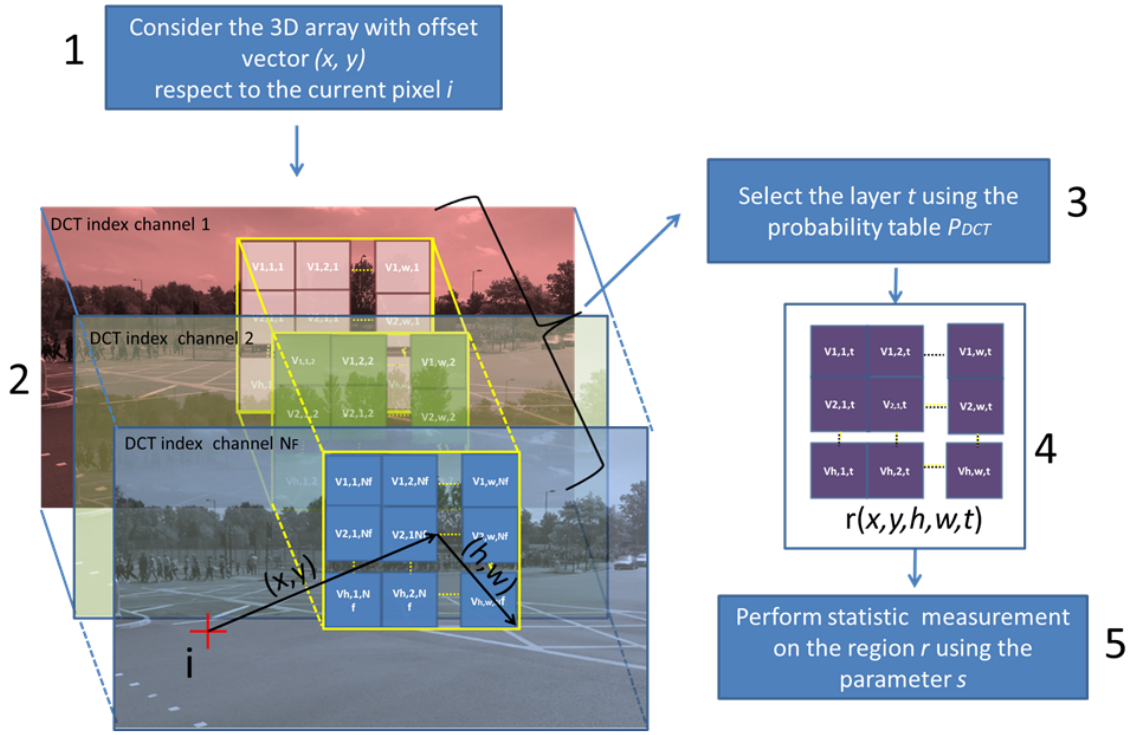


Figure 6. Extraction pipeline for feature f_1.

$$Stat_2(r, s) = \frac{\sum_{c \in r}(c - s)^2}{|r|} \tag{7}$$

$$Stat_3(r, s) = \frac{\sum_{c \in r} \delta_{cs}}{|r|} \tag{8}$$

where $|r|$ is the area of the region r and c is the result of the quantization process applied at each pixel in the region r. The quantization process used in the proposed approach is further detailed in Section 4.2. In equation (8), δ_{cs} is the Kronecker's delta function applied to the variables c and s. The performances obtained with the different measures are reported in Table 7 and will be analysed later in Section 6. These aforementioned measurements can be efficiently computed over a whole image by exploiting the integral histogram [61].

The second extracted feature, called feature f_2 detailed in Fig. 7 , is designed to compare two generic pixels P_1 and P_2 related to the i_{th} pixel to be classified in the semantic segmentation pipeline. Specifically, considering the pixels P_1 and P_2 obtained adding the offsets (x_1, y_1) and (x_2, y_2) to the i_{th} pixel, two 1D arrays are generated by the N_F channels extracted in the step 2) of the pipeline (see Fig. 7). This produces the feature vectors $V = V_1..V_{N_F}$ and $T = T_1...T_{N_F}$. In the step 4) of Fig. 7 two indexes w and z are extracted using the table P_{DCT}, and the elements V_w and T_z are selected. Such two values are finally combined using a mathematical operation (e.g., sum, log, pow, etc.) to generate the final feature value. The performances of each mathematical operation involved in the extraction of the feature f_2 are reported in Table 4 and will

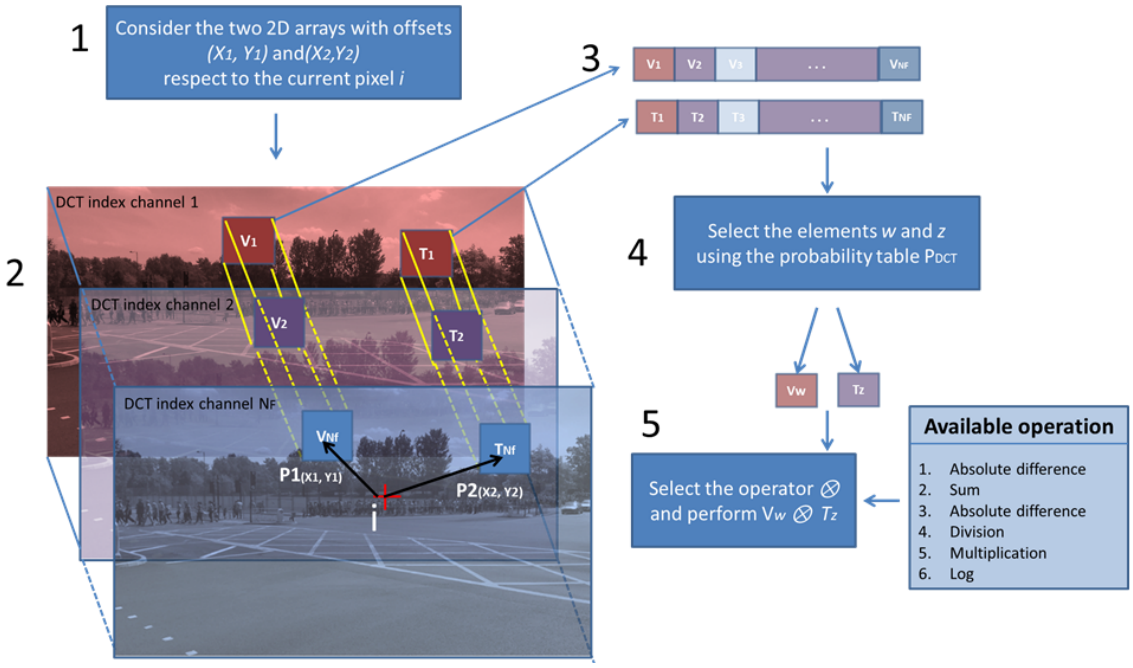


Figure 7. Extraction pipeline for feature f_2.

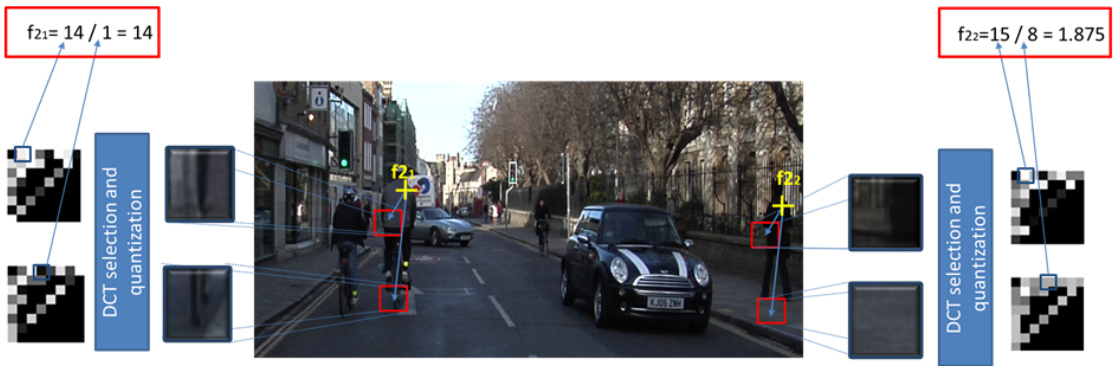


Figure 8. Example of features f_2.

be discussed later in Section 6.

Fig. 8 shows a toy example where the feature f_2 is computed in two different points specified by the yellow crosses. Just considering, the offsets (x_1, y_1) and (x_2, y_2) (represented by the blues arrows) the DCT blocks centred in the points denoted by the red square are picked and the N_F coefficients selected. For each block the two coefficients in the blue squares are selected and a division operation (in this example) is performed between them. The two analysed pixels are related to a bicyclist and a pedestrian. Due to the vertical high frequencies correlated to the wheel under the human the features value of f_{2_1} and the f_{2_2} are sensibly different allowing the STF system to properly perform the semantic segmentation. From this example we can observe that although the feature f_2 is very simple, it allows to recognize complex visual

Table 2. Number of operations required to compute f_1 for an image of 640x480 pixels

	Us=4	Us=2	Us=1
$Stat_1$	7692	1950	501
$Stat_2$	11538	2924	751
$Stat_3$	3846	974	250

cues inside the image.

5.1. Complexity of the proposed features

In this Section we describe the computational cost required to extract each proposed feature. For the feature f_1 the complexity is strictly related to the use of the integral histogram [62]. If the available memory is enough to store and use the integral histogram (N_F new layers of integer are required), the features f_1 can be computed in constant time. Otherwise, assuming that the bounding box of the region r cannot be more then B_1, the average number of operations required is:

$$\frac{\sum_i^{B_1} i^2 * s_{Op}}{B_1} = \frac{(B_1 + 1)(2 * B_1 + 1) * s_{Op}}{6} \tag{9}$$

where s_{Op} is a value that depends by the statistic measure used, and specifically it is 2 for $Stat_1$, 3 for $Stat_2$ and 1 for $Stat_3$. Table 2 summarizes the number of required operations when different statistics and different upsampling factors Us are used. Table 2 is obtained using an input image of 640x480 pixels and running the system with the a maximum bounding box B_1 equal to $width/3 = 213$ pixels.

On the other hand, since the feature f_2 is the result of just one operation between two numbers it can be computed always in constant time.

6. Experimental setting and results

To analyse the proposed solution we have performed experiments employing the Cambridge-driving Labeled Video Database (CamVid) [63, 64] and the MRSC-v2 dataset [13, 31]. In the following subsections are reported the experimental settings and the results obtained considering the aforementioned datasets.

6.1. CamVid Dataset

CamVid 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 as suggested in [28], we used only the 11 object categories, forming a majority of the overall labelled pixels (89.16%). Data were captured from the perspective of a driving automobile. The driving scenario increases the number and heterogeneity of the observed object classes. The parameters of the system are summarized

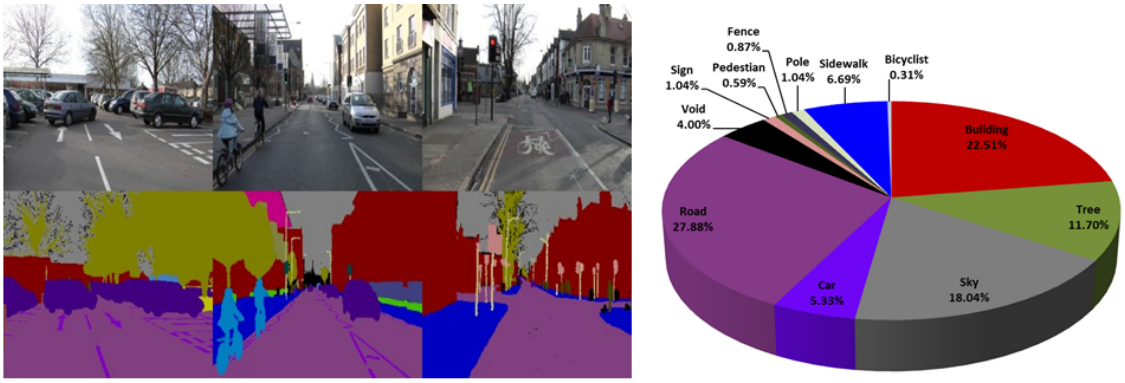


Figure 9. CamVid database and per-class distributions.

Table 3. System parameters

Related to	Name	Description	Default value
Proposed features	M	Modality for different features setting	M_4
	Us	Upsampling factor used to enlarge the image	2
	S	Type of statistic used to generate the feature f_1	$Stat_2$
	Qp	Number of quantization points used to quantize the Laplace distribution area	32
	B_1	Box size used to generate the feature f_1	$W^*/2$
	B_2	Box size used to generate the feature f_2	$resF^* \times 15$
STF system	D_1	Depth for the 1st forest	12
	D_2	Depth for the 2nd forest	15
	N_1	Number of the features randomly chosen to generate the nodes in the 1st forest	800
	N_2	Number of the features randomly chosen to generate the nodes in the 2st forest	800

* See Section 6 for the definition of W and $resF$

in Table 3.

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 as suggested in [28]. A validation step is applied to obtain the best configuration for each parameter. Specifically the validation process divides the training images in 2 sub-groups of equal size. The first group is used for training the validation and the remaining for test them. Moreover we have fixed a default value for all of the free parameters. These initial values are reported in the last column of Table 3. In the final test phase, the configuration set that has obtained the best performance is used to train the 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 mean-class accuracy, and the overall segmentation accuracy. Table 4 shows the results obtained when the novel features are introduced in the STF system and when different operations are used to compute the features f_2. The first 4 rows of Table 4 shows the classification results obtained by the system when each of the proposed features f_1 and f_2 are included in the STF system. Some tests use also a feature called "unary" that is obtained when the point P_1 and P_2 are the same and the selected DCT channels W and Z are equal. The next 6 rows of Table 4 show the results obtained using the different type of operations to compute feature f_2. From the results, we can see that adding both the features f_1 and f_2 to the STF system, improves the classification performance. Specifically the best results are obtained

Table 4. Results for different configuration of M

M	Overall	MeanClass
M1= STF	74.40	68.99
M2= STF & f_2 unary	74.86	69.90
M3= STF & f_1	75.55	70.46
M4= STF & f_1 & f_2 unary	74.84	70.42
M5= STF & f_1& f_2 unary & f_2 |diff|	75.12	70.52
M6= STF & f_1 & f_2 unary & f_2 sum	**75.51**	**71.01**
M7= STF & f_1 & f_2 unary & f_2 diff	75.24	70.94
M8= STF & f_1 & f_2 unary & f_2 div	**75.50**	**71.21**
M9= STF & f_1 & f_2 unary & f_2 mol	75.19	70.75
M10= STF & f_1 & f_2 unary & f_2 log	75.00	70.75

Table 5. Analysis 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 6. Results for different values of Us

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

when the "division" and the "sum" operations are considered to generate the feature f_2.

Tables 5(a), 5(b) and 5(c) analyse the performance related to the forest parameters, specifically the depths D_1 and D_2 of the 2 random forests involved into the system and the number of the features N_1 and N_2 randomly selected to generate each node. Increasing the values of these parameters gives in general a better accuracy.

Table 6 analyses the behaviours of the system when different upsampling factor Us are used. The best results are obtained when the image is upsampled by a factor of 4. To have an acceptable efficient system, it is recommended to use an upsampling factor of 4 only when the integral histogram is used in the system, otherwise, reminding the computational analysis proposed in Section 5 and according to the Table 2, an adequate trade-off between performance and high efficiency is obtained using an upsampling factor equal to 2. Table 7 shows the system accuracy obtained using each of the proposed statistics when different number of clusters are computed for quantizing the DCT data. The best results are obtained when

Table 7. 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**
$Stat_1$	75.12	70.52	75.57	70.99	74.43	71.00	74.46	71.30	74.90	70.95
$Stat_2$	75.48	71.21	75.41	71.11	74.89	71.63	74.90	71.06	**75.17**	**71.25**
$Stat_3$	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 8. Results for different values of B_1. W is $wt * Us/DCTblockSize$ where wt is the width of the image, Us is the upsampling factor and $DCTblockSize$ 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 9. Results for different values of B_2. $ResF$ is $DCTblockSize/Us$ where $DCTblockSize$ is the size of the DCT transformation block and Us the upsampling factor

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

Table 10. 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**
Shotton [13]	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 [28]	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 [29]	**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 [63]	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 [33]	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 [30]	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 [34]	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[35]	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

the statistic $Stat_2$ is selected. Moreover, only 8 clusters are enough to quantize the DCT data. We use clustering with 8, 16, 32 and 64 centroids. Table 7 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 clusters for each DCT coefficient we have 8xN_F different DCT Textons (in our case with 25 frequencies selected there are 200 DCT Textons). Furthermore, with this configuration, each DCT index data, can be saved in memory employing only 3 bits.

Table 8 and 9 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 $wt * Us/(3 * DCTblockSize)$ pixels is used for the feature f_1 and equal to $DCTblockSize * 15/Us$ pixels is used for the feature f_2 (where wt is the width of the image, Us is the upsampling factor and $DCTblockSize$ is the size of the DCT transformation block). Table 10 compares the results obtained by the state-of-the-art approaches with respect our proposal when the best configuration set of parameters is used. Instead, Table 12 shows the confusion matrix obtained by our solution.

Table 11. Comparison to state-of-the-art on the MSRCv2 dataset

	Classification for each class																						
Approach	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bike	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	Overall	MeanClass
Proposed	41.9	91.5	76.8	**87.6**	91.7	**92.4**	**85.1**	**62.5**	**90.5**	**72.0**	**77.3**	**72.7**	33.7	**29.8**	**92.3**	44.9	79.8	**78.0**	**36.7**	56.6	**24.3**	**74.0**	**67.5**
Shotton [13]	45.9	92.6	75.0	86.9	**91.7**	87.2	84.9	53.4	88.5	58.6	73.0	65.6	**40.4**	24.5	86.1	**48.8**	75.4	68.5	29.7	52.7	16.8	71.5	64.1
TexBoost [31]	**61.6**	**97.6**	**86.3**	58.3	50.4	82.6	59.6	52.9	73.5	62.5	74.5	62.8	35.1	19.4	91.9	15.4	**86.0**	53.6	19.2	**62.1**	6.6	72.2	57.6

Fig. 9 represents the distribution of the pixels per class. One can note a widely varying class prevalence in this dataset with the first two majority classes (Building and Road) alone containing more than 50% of the pixels. Obviously, the learning step for the small classes is in general more complex since they are not well represented in the database. The approaches in [28, 29, 30, 33, 34, 35] obtain better performances for the two classes most represented in the CamVid dataset (i.e., Building and Road - see Fig. 9 and Table 10) and hence show a better overall accuracy at the cost of losing the mean-classes performance (since they are less accurate in discriminating less represented classes in the dataset, such as Sign, Pedestrian, Bicyclist, etc. - see Fig. 9 and Table 10). In our approach instead, the challenging classes with small percentage of samples in the dataset have got a significant improvement in the per-class accuracy and hence the proposed approach obtains the better results in the mean-classes score.

Since the optimization is done over all the classes we have a significant improvement in the per-class accuracies and a more balanced performance at the cost of losing on the overall results. As a result, our approach obtains always better accuracy in the small classes (i.e., Pedestrian, Fence, Pole, Sign) with respect to all the considered approaches. In conclusion, one should note that, in the case of unbalanced dataset, the mean-class metric is a more reliable measure than the overall accuracy measure since it applies equal importance to all 11 classes. Fig. 10 shows some classification errors of our approach. 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.

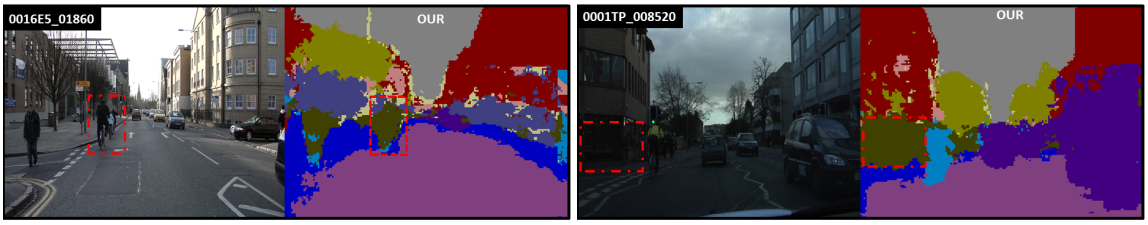
Fig. 11 shows an example of visual segmentation outputs, of our approach in comparison with the STF system. In this case, our approach has the ability to segment properly an area containing a bicyclist while the STF approach is failing. In Fig. 12 are compared the computational time obtained by our approach and the STF during the two semantic segmentation levels. These tests are performed on a PC with a processor i73930k 3.20 Ghz (6 cores) and with 32 Gb of memory RAM. Both approaches use the best parameters configuration (i.e., number of trees, depth, number of features analysed, bounding box, etc.). Moreover, features f_1 are computed without the support of the integral image. As we can see from Fig. 12, the proposed

Table 12. 11x11 Confusion Matric obtained on the CamVid database

	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**

Table 13. 21x21 Confusion Matric obtained on the MSRCv2 database

	Buil.	Grass	Tree	Cow	Sheep	Sky	Aerop.	Water	Face	Car	Bike	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Buil.	**41.9**	2.25	10.46	0.58	0.19	5.27	2.37	1.46	4.02	2.90	7.92	0.00	0.90	0.20	2.26	2.17	11.82	0.44	0.50	1.84	0.46
Grass	0.16	**91.5**	0.77	3.36	1.43	0.01	0.60	0.01	0.06	0.00	0.12	0.19	0.00	0.08	0.00	0.17	0.22	0.00	0.25	1.06	0.00
Tree	1.49	10.81	**76.8**	0.54	0.00	3.38	1.42	0.90	0.82	0.13	1.18	0.07	0.58	0.11	0.02	0.19	0.39	0.00	0.13	0.82	0.14
Cow	0.01	7.22	0.37	**87.6**	2.27	0.00	0.00	0.11	0.08	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.01	1.64	0.60	0.00
Sheep	0.05	4.91	0.02	1.34	**91.7**	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	1.50	0.00	0.08	0.39	0.00	0.02	0.00	0.00
Sky	1.63	0.12	1.30	0.00	0.00	**92.4**	0.74	1.50	0.00	0.01	0.00	0.00	0.06	1.84	0.00	0.02	0.22	0.00	0.00	0.00	0.05
Aerop.	8.67	1.30	0.06	0.26	0.00	2.52	**85.1**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.03	0.00	0.00	0.00	0.00
Water	6.18	6.48	0.69	0.30	0.13	6.34	0.01	**62.5**	0.01	2.18	1.61	0.04	0.25	1.41	0.01	0.14	9.70	0.33	0.24	0.42	1.01
Face	0.58	0.08	0.70	0.01	0.00	0.21	0.00	0.06	**90.5**	0.15	0.00	0.27	0.08	0.02	0.37	0.01	0.00	0.30	0.42	6.01	0.16
Car	13.29	0.00	0.31	0.00	0.00	0.01	0.00	0.93	0.00	**72.0**	0.08	0.00	0.96	0.02	0.00	0.07	11.99	0.19	0.02	0.06	0.01
Bike	4.23	0.01	0.37	0.00	0.00	0.00	0.00	0.00	0.14	1.40	**77.3**	0.00	0.12	0.00	0.00	1.37	14.17	0.86	0.01	0.00	0.00
Flower	0.31	3.45	2.80	1.17	1.61	0.01	0.00	0.22	0.83	0.01	0.00	**72.7**	8.35	2.52	2.20	0.01	0.25	0.20	0.38	2.88	0.00
Sign	26.42	0.83	3.20	0.08	0.24	2.58	0.00	2.36	0.56	0.36	0.71	1.82	**33.7**	3.68	12.11	0.99	6.02	1.63	0.42	1.88	0.35
Bird	11.63	9.68	0.98	0.58	7.90	3.35	0.00	8.74	0.10	0.24	2.07	0.01	0.20	**29.8**	0.00	4.31	14.40	2.86	2.71	0.22	0.14
Book	3.03	0.03	0.16	0.00	0.00	0.00	0.00	0.01	1.28	0.10	0.00	0.08	0.34	0.00	**92.3**	0.00	0.31	0.25	0.04	1.97	0.02
Chair	1.21	9.76	9.98	9.49	0.89	0.03	0.00	0.06	0.04	0.64	5.45	0.00	1.07	0.70	0.35	**44.9**	9.33	2.53	2.90	0.64	0.00
Road	3.65	0.70	0.47	0.00	0.89	0.31	0.32	6.18	0.61	0.89	3.18	0.00	0.23	0.04	0.00	0.32	**79.8**	1.48	0.47	0.44	0.04
Cat	1.42	0.00	0.72	0.00	0.00	0.00	0.00	0.71	0.14	0.03	7.33	0.00	0.07	0.28	0.18	3.04	7.80	**78.0**	0.24	0.00	0.00
Dog	17.91	3.07	6.09	0.25	0.07	0.47	0.91	0.74	4.57	0.01	0.00	0.07	3.13	4.66	0.04	3.59	11.95	1.80	**36.7**	3.95	0.00
Body	4.87	4.18	2.52	3.31	0.00	1.64	0.38	3.46	8.28	1.13	0.01	0.47	0.88	0.49	3.29	0.31	2.98	1.68	1.75	**56.6**	1.72
Boat	21.44	0.10	0.80	0.00	0.00	2.51	0.00	22.31	0.00	5.78	10.50	0.00	1.36	1.50	0.00	1.17	7.41	0.00	0.00	0.80	**24.3**



(a) (b)

Figure 10. Examples of classification error obtained using our approach. In 10(a) a region containing a bicycle is confused with a pedestrian; in 10(b) an area belonging to a building is confused with the class pedestrian.

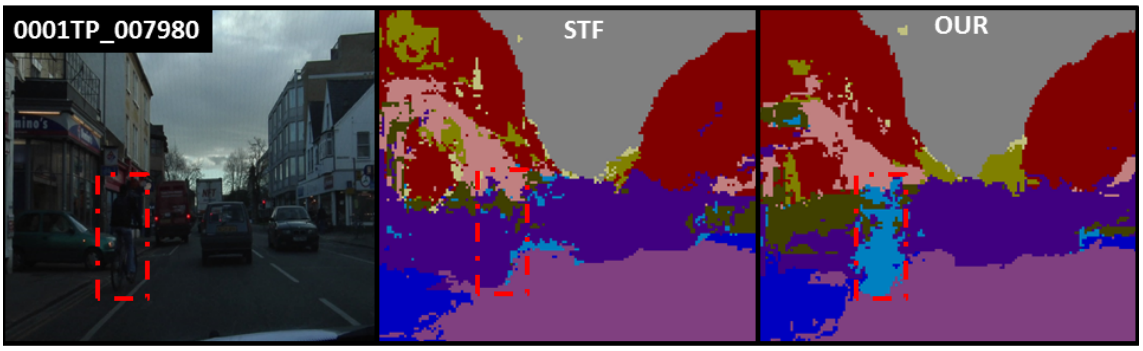


Figure 11. Example of visual segmentation improvement, obtained using our approach with respect to the STF.

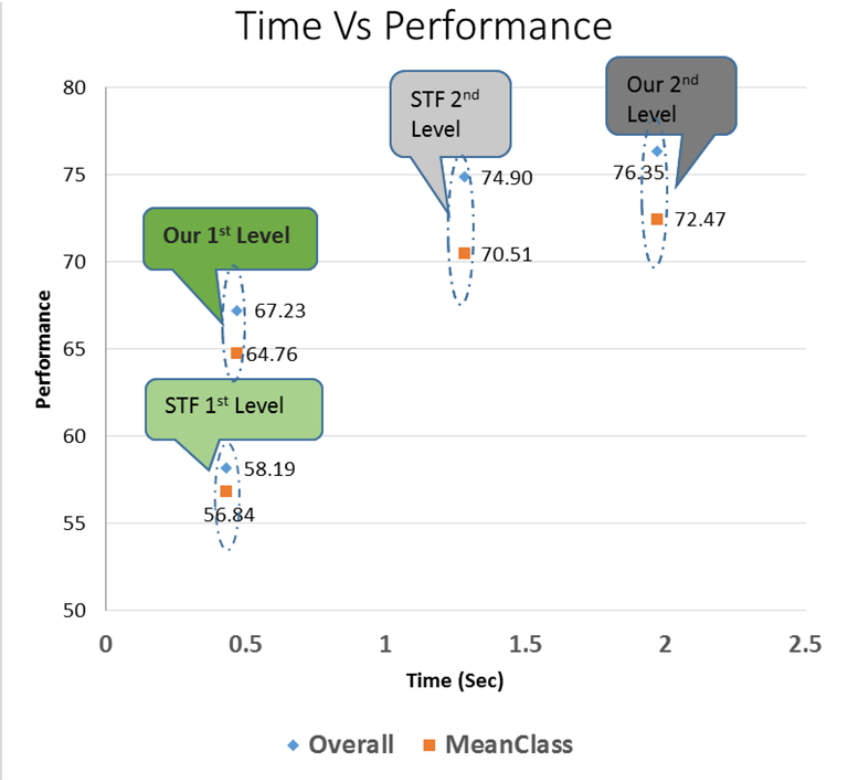


Figure 12. Computational time obtained by our approach and the by the STF [13] during the two segmentation phases.

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 has 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.

6.2. MSRC-v2 Dataset

This Section presents results of image segmentation on the database MSRC-v2 that contains photographs of real objects viewed under general lighting conditions, poses and viewpoints, for a total of 591 images. In this experiment we have used, for all the parameters, the same configuration obtained in the previous Section (except for the two depth levels that we have changed into 15 for D_1 and 17 for D_2). We have again compared our approach with state-of-the-art and the results are showed in Table 11. As we can see from the table, also in this case results are in favour of the proposed approach. Specifically, our method achieve the highest segmentation accuracy of 74.0% (1.8% more than Shotton [13] and 2.5% more than TexBoost [31]) for the overall pixel accuracy. Whereas regarding the average across categories we obtaining 67.5% (3.4% more than Shotton [13] and 9,9% more than TexBoost [31]). Still better performance could likely be achieved by a complete experimental validation on this database. Finally, in Table 13 we can analyse in more details the confusion matrix obtained by the proposed approach on the MSRC-v2 database.

7. Conclusion

This paper describes an approach for semantic segmentation of images. Two novel texture features based on DCT data are introduced in the Semantic Texton Forest framework [13]. The proposed DCT features describe complex textures capable to recognize object and region with different frequencies characteristics. Our approach makes use of 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 considered databases. Moreover, in a real scenario our system could show further improvements since usually a large version of the image is available in the pipeline. This avoids to perform the proposed upsampling block in the pipeline and generating a more reliable DCT data that are not affected by the interpolation.

8. Acknowledgments

This research has been supported by STMicroelectronics [65].

References

[1] M. Johnson, Semantic segmentation and image search, Ph.D. thesis, University of Cambridge (April 2008).

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11) (2012) 2274–2282.

[3] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images, in: IEEE International Conference on Computer Vision, Vol. 1, 2001, pp. 105–112.

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

[5] M. M. Mokji, S. A. R. A. Bakar, Gray level co-occurrence matrix computation based on Haar wavelet, in: Computer Graphics, Imaging and Visualisation, CGIV '07, Washington, DC, USA, 2007, pp. 273–279.

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

[7] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transaction on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[8] M. Varma, R. Garg, Locally invariant fractal features for statistical texture classification, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[9] S. Savarese, J. Winn, A. Criminisi, Discriminative object class models of appearance and shape by correlatons, in: IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, Washington, DC, USA, 2006, pp. 2033–2040.

[10] C. Farabet, C. Couprie, L. Najman, Y. Lecun, Scene parsing with multiscale feature learning, purity trees, and optimal covers, in: J. Langford, J. Pineau (Eds.), International Conference on Machine Learning, ACM, New York, NY, USA, 2012, pp. 575–582.

[11] S. Battiato, A. Bruna, G. Messina, G. Puglisi, Image Processing for Embedded Devices: From CFA Data to Image/video Coding, Applied Digital Imaging, Bentham Science Publishers, 2010.

[12] F. Giovanni Maria, R. Daniele, T. Valeria, G. Mirko, B. Sebastiano, Representing scenes for real–time context classification on mobile devices, Pattern Recognition 48 (4) (2015) 1086–1100.

[13] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[14] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transaction on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.

[15] X. Ren, J. Malik, Learning a classification model for segmentation, in: IEEE International Conference on Computer Vision, Vol. 2, Washington, DC, USA, 2003, pp. 10–.

[16] M. P. Kumar, P. H. S. Torr, A. Zisserman, Obj cut, in: IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 01, Washington, DC, USA, 2005, pp. 18–25.

[17] T. Malisiewicz, A. A. Efros, Recognition by association via learning per-exemplar distances, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2008.

[18] C. Gu, J. J. Lim, P. Arbelaez, J. Malik, Recognition using regions, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1030–1037.

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

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

[21] I. Endres, D. Hoiem, Category independent object proposals, in: European Conference on Computer Vision, Berlin, Heidelberg, 2010, pp. 575–588.

[22] L. Ladicky, C. Russell, P. Kohli, P. H. S. Torr, Graph cut based inference with co-occurrence statistics, in: European Conference on Computer Vision, Berlin, Heidelberg, 2010, pp. 239–253.

[23] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2004) 137–154.

[24] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: C. Schmid, S. Soatto, C. Tomasi (Eds.), IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, 2005, pp. 886–893.

[25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[26] L. D. Bourdev, J. Malik, Poselets: Body part detectors trained using 3d human pose annotations, in: IEEE International Conference on Computer Vision, 2009, pp. 1365–1372.

[27] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, J. Malik, Semantic segmentation using regions and parts., in: IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 3378–3385.

[28] J. Tighe, S. Lazebnik, Finding things: Image parsing with regions and per-exemplar detectors, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3001–3008.

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

[30] C. Zhang, L. Wang, R. Yang, Semantic segmentation of urban scenes using dense depth maps, in: European Conference on Computer Vision, Berlin, Heidelberg, 2010, pp. 708–721.

[31] J. Shotton, J. Winn, C. Rother, A. Criminisi, Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation, in: European Conference on Computer Vision, 2006, pp. 1–15.

[32] P. Dollar, Z. Tu, S. Belongie, Supervised learning of edges and object boundaries, in: IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, Washington, DC, USA, 2006, pp. 1964–1971.

[33] P. Sturgess, K. Alahari, L. Ladicky, P. H. S. Torr, Combining appearance and structure from motion features for road scene understanding, in: British Machine Vision Conference, 2009, pp. 62.1–62.11.

[34] K. R. Georgios Floros, B. Leibe, Multi-class image labeling with top-down segmentation and generalized robust p^n potentials, in: British Machine Vision Conference, 2011, pp. 79.1–79.11.

[35] L. Ladický, P. Sturgess, K. Alahari, C. Russell, P. H. S. Torr, What, where and how many? combining object detectors and CRFs, in: European Conference on Computer Vision, Berlin, Heidelberg, 2010, pp. 424–437.

[36] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: European Conference on Computer Vision - International Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.

[37] S. Rota Bulò, P. Kontschieder, Neural decision forests for semantic image labelling, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 81–88.

[38] P. Kontschieder, S. Rota Bulò, M. Pelillo, H. Bischof, Structured labels in random forests for semantic labelling and object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (10) (2014) 2104–2116.

[39] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[40] A. Criminisi, J. Shotton, Decision Forests for Computer Vision and Medical Image Analysis, Advances in Computer Vision and Pattern Recognition, 2013.

[41] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, 1984.

[42] Y. Amit, D. G. Y, Shape quantization and recognition with randomized trees, Neural Computation 9 (1997) 1545–1588.

[43] T. K. Ho, Random decision forests, in: International Conference on Document Analysis and Recognition, Vol. 1, Washington, DC, USA, 1995, pp. 278–.

[44] A. Criminisi, J. Shotton, D. Robertson, E. Konukoglu, Regression forests for efficient anatomy detection and localization in CT studies, in: International conference on Medical computer vision: recognition techniques and applications in medical imaging, MCV'10, Berlin, Heidelberg, 2011, pp. 106–117.

[45] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

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

[47] W. K. Pratt, Digital Image Processing, New York, NY, USA, 1978.

[48] J. D. Eggerton, Statistical distributions of image DCT coefficients, Computers & Electrical Engineering 12 (1986) 137–145.

[49] T. Eude, R. Grisel, H. Cherifi, R. Debrie, On the distribution of the DCT coefficients, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 5, 1994, pp. 365–368.

[50] F. Müller, Distribution shape of two-dimensional DCT coefficients of natural images, Electronics Letters 29 (1993) 1935–1936.

[51] S. Smoot, L. A. Rowe, Study of DCT coefficient distributions, in: SPIE Symposium on Electronic Imaging, Vol. 2657, 1996, pp. 403–411.

[52] G. S. Yovanof, S. Liu, Statistical analysis of the DCT coefficients and their quantization, in: Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers, Vol. 1, 1996.

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

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

[55] ITU, Iso/iec 10918-1 : 1993(e) ccit recommendation t.81 (1993).

[56] V. C. Smith, J. Pokorny, Spectral sensitivity of the foveal cone photopigments between 400 and 500 nm, Vision Research 15 (2) (1975) 161 – 171.

[57] S. Battiato, M. Mancuso, A. Bosco, M. Guarnera, Psychovisual and statistical optimization of quantization tables for DCT

compression engines., in: ICIAP, IEEE Computer Society, 2001, pp. 602–606.

[58] H. A. Peterson, H. Peng, J. H. Morgan, W. B. Pennebaker, Quantization of color image components in the DCT domain, in: B. E. Rogowitz, M. H. Brill, J. P. Allebach (Eds.), Human Vision, Visual Processing, and Digital Display II, Vol. 1453 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1991, pp. 210–222.

[59] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: IEEE International Conference on Computer Vision, Vol. 1, 2005, pp. 604–610 Vol. 1.

[60] A. Likas, N. Vlassis, J. J. Verbeek, The global k-means clustering algorithm, Pattern recognition 36 (2) (2003) 451–461.

[61] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001, pp. I–511–I–518.

[62] F. M. Porikli, Integral histogram: A fast way to extract histograms in cartesian spaces., in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2005, pp. 829–836.

[63] G. J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla, Segmentation and recognition using structure from motion point clouds, in: European Conference on Computer Vision, Berlin, Heidelberg, 2008, pp. 44–57.

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

[65] STMicroelectronics, Advanced System Technology - Computer Vision group. URL http://www.st.com/.