SEGMENT-BASED OBJECT DETECTION AND RECOGNITION

by

Rabia Gökçe Erdem

B.S., Electrical and Electronics Engineering, Pamukkale University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2018

SEGMENT-BASED OBJECT DETECTION AND RECOGNITION

APPROVED BY:

Prof. H. Işıl Bozma (Thesis Supervisor)	.....
Prof. Yağmur Denizhan	.....
Assoc. Prof. Sanem Sarıel	.....

DATE OF APPROVAL: 08.01.2018

ACKNOWLEDGEMENTS

First of all, I would like to express my deep gratitude to my thesis supervisor Prof. H.Işıl Bozma for the continuous support during my study and research, for her patience, motivation and immense knowledge. Without her guidance this thesis would have never been accomplished. I could not have imagined having a better supervisor and mentor for my master degree.

I would also like to thank to my friends who supported and motivated me all the time. Their motivation always kept me fresh and determined.

Finally, I want to thank my parents. I always felt their support and love during my life. Without them, it would be impossible for me to realize my achievements.

ABSTRACT

SEGMENT-BASED OBJECT DETECTION AND RECOGNITION

This thesis is concerned with object detection and recognition in complex scenes. Both are necessary if robots are to be capable of interacting with their surroundings. Object detection is based on color based segmentation. Features of these segments are used in order to detect and recognize objects. A set of SVM model is constructed in the appearance space and object recognition is done based on these models. The proposed approach is evaluated experimentally in varying scenarios.

ÖZET

BÖLÜTLEME TABANLI NESNE BULMA VE TANIMA

Bu çalışmada, karmaşık sahnelerde nesneleri bulma ve tanıma konuları ele alınmıştır. Yapay zeka ve bilgisayarla görü alanında yoğun olarak çalışılan konulardan biri olan karmaşık sahnelerde nesne tanıma ile ilgili yeni bir yaklaşım burada önerilmektedir. Yapay zeka uygulamalarının büyük bir kısmı gerçek zamanlı olarak çalışacak sistemlere yöneliktir. Bu gerçek zamanlı sistemlerde çalışacak olan nesne bulma programlarının hızlı, maliyet etkin ve hafif olması beklenmektedir. Bu çalışmada önerilen method da gerçek zamanlı sistemlerde kullanılmaya uygundur.

Önerilen method, renk temelli sahne bölütleme konusunda yeni bir yaklaşım içermektedir. Sahnelerden elde edilen bölütlerin öznitelikleri kullanılarak nesnelerin bulunması ve tanınması sağlanmaktadır. Öznitelikler, destek vektör makineleri yöntemiyle modellenerek, elde edilen model kaydedilmiştir. Bu model kullanılarak bölütlerin sınıflanması ve tanınması yapılmaktadır. Model, artımlı tipte olmadığından kaydedildikten sonra tekrar tekrar kullanılabilmektedir. Bu da gerçek zamanlı sistemlerde süre açısından verimlilik sağlamaktadır.

Deneysel sonuçlar, farklı ışık koşulları ve dinamik, değişken sahneler kullanılarak elde edilmiştir. Bu çalışmada amaç, renk temelli sahne bölütleme kullanılarak güdümlü bir öğrenme methodu geliştirmek olmuştur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . vii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. OBJECT DETECTION . . . 3
2.1. Related Literature . . . 3
2.2. Proposed Approach . . . 5
2.2.1. HSV Based Quantization . . . 6
2.2.2. Labeling Connected Components . . . 7
2.2.3. Final Merging of Connected Components . . . 8
2.3. Experimental Results . . . 10
3. SEGMENT-BASED RECOGNITION . . . 23
3.1. Introduction . . . 23
3.2. Related Literature . . . 23
3.3. Proposed Approach . . . 27
3.4. Segments - Objects Database . . . 27
3.4.1. SVM Based Learning . . . 38
3.5. SVM Based Recognition . . . 40
3.6. Experimental Results . . . 43
3.6.1. Validation Experiments . . . 45
3.6.2. Integrated Object Detection & Recognition . . . 46
4. CONCLUSION . . . 49
REFERENCES . . . 50
APPENDIX A: COLOR QUANTIZATION . . . 58

LIST OF FIGURES

Figure 1.1. Proposed approach 1

Figure 2.1. Segmentation flow 5

Figure 2.2. Mapping between HSV color regions and color quanta 6

Figure 2.3. HSV-Based color quantization. 7

Figure 2.4. 8 connectedness 8

Figure 2.5. Color quantization example 10

Figure 2.6. Effect of variations of parameter σ_D 11

Figure 2.7. Effect of variations of parameter N_1 12

Figure 2.8. Effect of variations of parameter N_2 13

Figure 2.9. COLD - Freiburg Portion A, Extended Path Cloudy2 14

Figure 2.10. Comparative results for sample images from Area 1 (Printer area) scene 15

Figure 2.11. Comparative results for sample images from Area 2 (Corridor) scene 16

Figure 2.12. Comparative results for sample images from Area 3 (Kitchen) scene 17

Figure 2.13. Comparative results for sample images from Area 5 (Two person office) scene 19

Figure 2.14. Comparative results for ISL scenes 20

Figure 3.1. Recognition flow 27

Figure 3.2. A complex scene image given with three proportions 29

Figure 3.3. Major axis and minor axis 30

Figure 3.4. Segment ratio example 31

Figure 3.5. Segment ratio for different objects 31

Figure 3.6. Normalized hue values for different objects 32

Figure 3.7. 8 bin histogram values for chair 35

Figure 3.8. 8 bin histogram values for jaguar 35

Figure 3.9. Spatial filters 36

Figure 3.10. Example filter responses for different objects 36

Figure 3.11. Example filter responses for chair 37

Figure 3.12. Segments and the objects they represent 44

Figure 3.13. Example images used for auto-search test. 47

Figure A.1. Colors 58

LIST OF TABLES

Table 2.1. Color adjacency matrix 9

Table 2.2. Comparative results for COLD areas 21

Table 2.3. Object-segment analysis on COLD scenes 22

Table 3.1. Segment center 28

Table 3.2. Average Segment Locations 29

Table 3.3. Effect of histogram size on validation test data 33

Table 3.4. Example filter responses for different objects represented with normalized histogram values 34

Table 3.5. Features 38

Table 3.6. Probabilities 41

Table 3.7. Decision example for low recognition probability, rule # 4 41

Table 3.8. Decision example for rule # 2 42

Table 3.9. Comparative recognition performance 45

Table 3.10. Our method vs CNN 47

Table 3.11. Specific Definitions 47

Table 3.12. Statistical results for experiments. Precision, recall and accuracy definitions are given in Equation 3.8, 3.9 and 3.10. 48

Table A.1. HSV ranges 59

LIST OF SYMBOLS

A	Segment area
a_{ij}	Color similarity representation
b	Offset constant
H	Hue range
$\mathcal{H}$	Summation of hue values in segment
h_n	Hue value
h_r	Normalization parameter of EDISON method for range domain
h_s	Normalization parameter of EDISON method for spatial domain
k	Threshold function of FH algorithm
min_reg	Minimum size of a class in EDISON method
min_size	Minimum component size of FH algorithm
N_1	Threshold for small segments
N_2	Threshold for large segments
N_q	Number of color quanta
$\mathcal{P}$	Color of a pixel in HSV color space
$\mathcal{Q}$	Color of a pixel in quantization color space
S	Saturation range
x	Descriptor of a segment
V	Value range
w	Weight vector
y	Object classes
ξ	Color quantization map
ξ_t^{ij}	Slack variable
σ	Smoothing parameter of FH algorithm
σ_D	Dilation parameter

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
CIE	Commission Internationale de l'Eclairage
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
COLD	COsy Localization Database
EDISON	Edge Detection and Image Segmentation
FH	Felzenszwalb-Huttenlocher
HIS	Hierarchical Image Segmentation
HBS	HSV-Based Segmentation
HOG	Histograms of Oriented Gradients
HSV	Hue Saturation Value
ICA	Independent Component Analysis
ISL	Intelligent Systems Laboratory
JSEG	J Image Segmentation
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LUV	Lightness Uniform chromaticity scale Valence
PCA	Principal Component Analysis
RGB	Red Green Blue
RGB-D	Red Green Blue Depth
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SRG	Seeded Region Growing
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

1. INTRODUCTION

Understanding a scene is crucial if robots are ever to interact with their surroundings. It requires recognizing the objects that are in the scene. This thesis is concerned with object recognition. The proposed approach consists of three stages: object detection, supervised learning and object classification. Object detection is based on

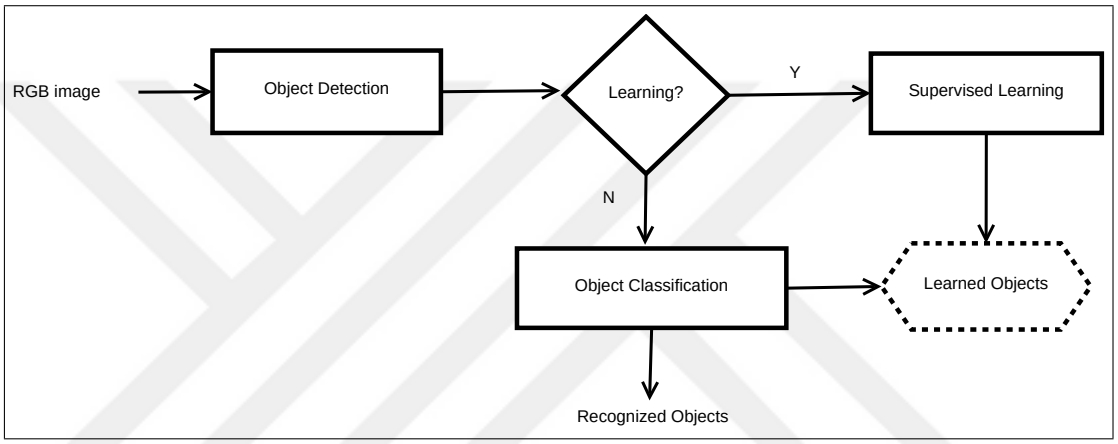


Figure 1.1. Proposed approach

segmentation. In the first stage of the proposed approach, the image is transformed into a color-quantized image. The second step is to find the color-uniform regions via a modified version of a connected components algorithm - taking into account the quantization of Hue Saturation Value (HSV) color space. At this stage, color quantas are obtained according to the appearance of colors - thus enabling a smoother quantization scheme and the integrity of the segments. In previous HSV color space quantization studies, the hue channel is mostly used as a reference to partition the color space into bins. This results in a loss in the integrity of segments. This study, however differs in that it uses the color appearance to for this. In this approach, resulting regions are merged depending on their size, adjacency and color. It is observed that even small segments are preserved, which may be important for object recognition.

After this object detection stage, in the object classification stage, the resulting segments are used to learn and classify objects. First, learning a set of object classes is done in a supervised manner based on a set of features as extracted from the segments.

This is based on the support vector machine (SVM). Once learning is completed, object recognition performance is tested via considering segments associated with the learned object classes.

This thesis is organized as follows: In Chapter 2, the object detection method is presented along with experimental results using the COsy Localization Database (COLD) benchmark. Object learning and classification is explained in Chapter 3. The thesis concludes with a brief summary. The representation of color quantas is given in Appendix A.

2. OBJECT DETECTION

Object detection is based on image segmentation. First, we discuss related literature. This is followed by the presentation of the proposed approach.

2.1. Related Literature

Image segmentation is an extensively studied area in machine vision. While a plethora of approaches with very good segmentation capabilities have been developed, real-time applicability and robustness of the developed algorithms still remain to be challenging issues [1]. Sometimes the size of the input images is so big, that we need to down-sample them in order to shorten the processing time. Most of the algorithms are proposed for binary images and gray scale images [2, 3].

Different image segmentation techniques have been proposed in literature. Some of them are for general purpose and some of them are for specific classes of images [4]. These methods can be grouped into four categories : pixel-based, region-based, edge based, graph based and artificial neural networks [5].

In pixel-based approaches, segmentation is based on operations such as thresholding [6] or histogram [7]. In color image segmentation process, color quantization is an essential step. For example, in histogram based segmentation, HSV color space divided into 36 colors [8], 166 colors [9] and 300 bins [10]. Commission Internationale de l'Eclairage Lightness Uniform chromaticity scale Valence (CIE LUV) color space is quantized in [11]. A simple thresholding operation can be used to recognize a specific object on same background for each time. Binary images are used mostly for fast processing. Boundaries of objects are extracted and each connected component is labeled with same number. The complexity of the algorithm depends on the source images.

Currently, Deep Convolutional Neural Networks (DCNNs), eCognition, J Image Segmentation (JSEG), Felzenszwalb-Huttenlocher (FH), Seeded Region Growing

(SRG), Edge Detection and Image Segmentation (EDISON) and Hierarchical Image Segmentation (HIS) are the most known methods for image segmentation.

In region-based approaches, methods are further sub-categorized into split and merge approach and region growing approach [12]. SRG is region growing type of method which takes individual points as seeds and grows the seed regions in an iteratively [13]. JSEG algorithm is also region growing type of method that applied on J-images [14, 15] and overall performance is very good. However JSEG has two limitations due to the color quantization parameter which will affect the segmentation results. This algorithm is suitable for segmentation of satellite digital images and outdoor nature images. A form of best merge region-growing segmentation method contained in eCognition which is mostly used for airborne data and landscape labeling with three parameters for shape, compactness and scale [16]. The eCognition approach grows regions with the goal of minimizing image object heterogeneity by accounting for both local image texture and the size of groups of pixels [17]. For indoor scene labeling it is not suitable but it is successful on urban environments. This is due to the color quantization parameter and the varying shades due to the illumination [18].

In edge based approaches, segmentation is based on edge flow or deformable contours such as snakes. EDISON [19] combines the mean-shift algorithm with image edge information. EDISON is sensitive to parameter selection and has more than one parameter.

In [20, 21] graph based approaches represented. HIS is a graph based algorithm which uses hierarchy of regions for performing segmentation [22]. FH is also a graph based algorithm able to capture perceptually important non-local image characteristics. Even though graph cut is a popular technique for interactive image segmentation, it has problems with segmenting thin elongated objects [23]. For crowded scenes it is hard to segment objects. A comparative study has shown that a hybrid method based on FH and EDISON yields best results [24].

Artificial neural networks yield best results among other methods. DCNNs have recently shown state of the art performance in semantic image segmentation. However, although they provide satisfactory results in image classification and object detection, they learn both features and noise equally well, increasing the potential for over-fitting. Also, the DCNN knowledge cannot be fully utilized in early stages of training which make DCNNs good for high level tasks [25, 26].

2.2. Proposed Approach

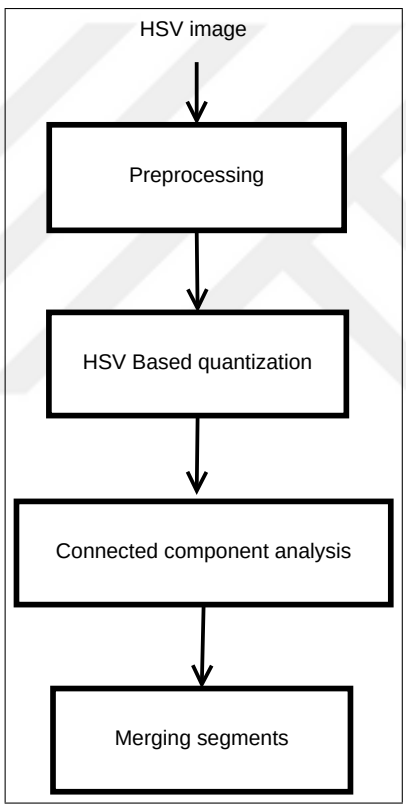


Figure 2.1. Segmentation flow

In this thesis we propose a novel region-based segmentation algorithm called HSV-Based Segmentation (HBS) that attempts to overcome the problems mentioned in the previous chapter.

For segmentation, the first step is to divide the incoming image into meaningful regions via a modified version of the connected component labeling algorithm that exploits the properties of the HSV color space. Following, the resulting smaller segments are merged with their adjacent regions. In the final step, larger segments are merged

again based on their adjacency and similarity of color associations. It is observed that the resulting segments are more robust against changes in perceived colors due to illumination changes.

As mentioned above, segmentation is done according to color. HSV color space is divided into bins and each bin is assigned to a color. In this way, HSV color space is quantized to 25 colors. Each pixel of the image has color information. This color information is used for segmentation. Similar colors are grouped together and they correspond to an object for indoor applications.

2.2.1. HSV Based Quantization

Color quantization transforms the incoming image via a map $\xi : \mathcal{P} \rightarrow \mathcal{Q}$ where $\mathcal{Q} = \{1, \ldots, N_q\}$ that assigns each pixel with a label corresponding to a color quantum. The quanta are determined offline via partitioning the HSV space into $N_q = 25$ quanta based on 172 regions and defining the map ξ accordingly.

Color space manually divided into 25 quanta based on visual appearance of colors while avoiding drastic changes within each color quanta. Each color channel is treated as controlled variable in a test scenario. Changes are analyzed accordingly and 172 regions are extracted (Table A.1). Each region is assigned to only one color quanta as shown in Figure 2.2. The resulting colors and their indices are as shown in Figure A.1 in Appendix.

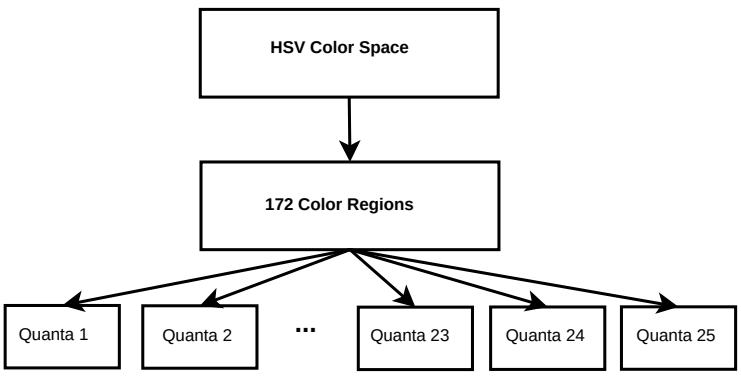
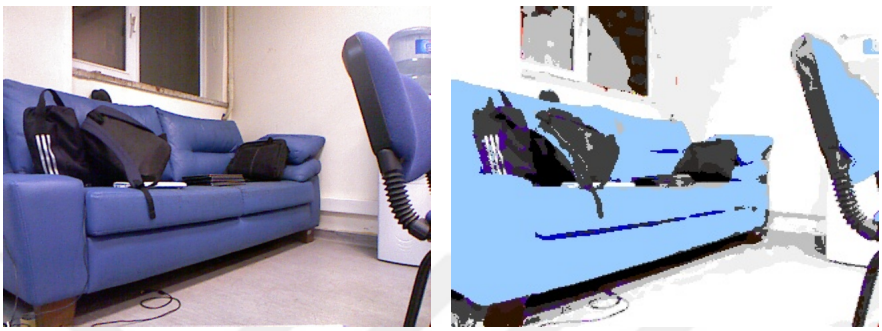


Figure 2.2. Mapping between HSV color regions and color quanta

A sample HSV color-based quantization is as shown in Figure 2.3. Illustration of color quanta are chosen close to real colors of the scene. General aspects of visual appearance are preserved and not all fine details but sufficient details remained after quantization.



(a) Original image (b) Quantized image

Figure 2.3. HSV-Based color quantization.

2.2.2. Labeling Connected Components

In the beginning of this thesis, there was another connected component labeling method which was then replaced by simple raster scan method. The reason for this change is that the first method was time-consuming. Both methods will be explained in this section.

The first method is based on region growing based approach. A seed (pixel) is taken from image and 8-neighbours of this pixel are candidates for being connected to the seed (Figure 2.4).

Color information is used as connectedness criteria. Color quanta given in the previous section are determined for each pixel. If candidate pixel has same color with the seed, it is labeled with the same label of the seed.

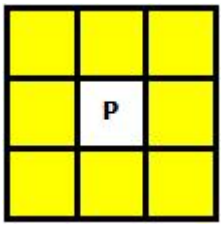


Figure 2.4. 8 connectedness

If a candidate is determined to be connected to the seed, this is added to the segment array of the seed and this added pixel also becomes a seed. For a specific segment, there is a seed array, a candidate array and a segment array. When a candidate becomes a seed, 8-neighbours of the candidate are checked again. Because this is a very time-consuming process, the process time is shortened considerably by changing this method.

In the new method, connected components labeling algorithm is applied on the color quantized image [27]. The goal of the connected component is to have segments that find regions with the same color label. A simple raster scan of the color labeled pixels is employed: left to right and from top to bottom.

2.2.3. Final Merging of Connected Components

The final step is to merge segments based on their size, labels and adjacency. The merging proceeds in two steps. First, all segments smaller than N_1 pixels are integrated with their adjacent segments where those adjacent segments have more than N_1 pixels. Segments smaller than N_1 pixels occupy extremely tiny areas in the scene. The second step is to merge larger segments - if that is found to be necessary based on size, adjacency and color. Here, the merging is applied only on components that have at least N_2 pixels. Each segment is processed one-by-one. While a segment is being processed, the pixels from the adjacent regions are determined using morphological dilation with parameter σ_D and xor operations. If the labels of these pixels are found to be similar to that of the current segment, then pixels are added to the current region.

This similarity is based on visual data and color adjacency matrix generated manually according to the similarity measure. For example, dark gray is similar to black and mixed black, mixed black is similar to dark gray. In Table 2.1 given similarities are represented. Each color quanta is represented with a number are given as shown in Appendix A. The same number representation is valid for Table 2.1. Color quanta $i \in \mathcal{Q}$ and $j \in \mathcal{Q}$ are deemed to be adjacent if the corresponding entry is $a_{ij} = 1$. Also in this matrix each color is marked similar with itself. The reason for such marking is, two segments belong to same object with same color next to each other can be extracted. We need to align them again in order to preserve integrity.

Table 2.1. Color adjacency matrix

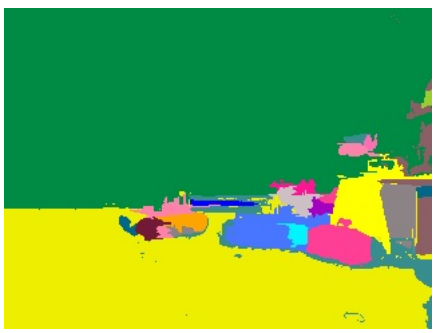
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

2.3. Experimental Results

The experiments are done - considering images from COLD dataset [28] and Intelligent Systems Laboratory (ISL) image database. Both databases contain complex indoor scenes acquired by a robot. COLD images belong to a map shown in Figure 2.9. Our proposed algorithm is compared with two previous approaches. A sample segmentation is shown in Figure 2.5. Without any segment merging, the original visual data as seen in Figure 2.5(a) is quantized into 1073 different color segments as seen in Figure 2.5(b). With the merging operation with parameters $\sigma_D = 3$, $N_1 = 70$ and $N_2 = 500$, there are only 26 resulting segments as seen in Figure 2.5(c). Next,



(a) original (b) Simple segmentation, 1073 segments

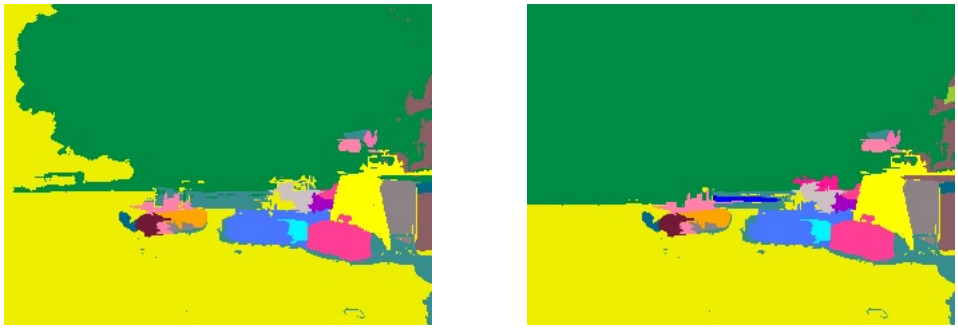


(c) $\sigma_D = 3$, 26 segments

Figure 2.5. Color quantization example with numerical results.

we consider the effects of the algorithm parameters. First, we consider the adjacency parameter σ_D . The effect of σ_D is demonstrated in Figure 2.6. If σ_D is too small, no merging expected. As σ_D gets larger, wrong mergings tend to occur as segments belonging to different objects may be merged. A sample case is presented in Figure 2.6. In this case, it is observed that the segment corresponding to the wall (green segment) is merged with that associated with the ground (yellow). As such, $\sigma_D = 3$ is deemed to

be a good value. N_1 parameter is used for involving small segments that doesn't have



(a) $\sigma_D = 7$, 24 segments (b) $\sigma_D = 3$, 26 segments

Figure 2.6. Effect of variations of parameter σ_D where $N_1 = 70$ and $N_2 = 500$.

any semantic meaning for object recognition task into a bigger neighbour segment. As $N_1 = 70$, this means that any segment with size less than 70 pixels will be merged with a bigger neighbour segment. Briefly, this choice means that any segment that occupies less or equal to $\approx 0.09\%$ of image will be ignored. In Figure 2.7, the effect of various changes in N_1 are given. Parameters used in Figure 2.7(a) and Figure 2.7(b) are the same except for the N_1 parameter. Numerical results show that in Figure 2.7(b) there are more segments than in Figure 2.7(a). In Figure 2.7(a) there are 27 segments and in Figure 2.7(b) 73 segments exist. Visual data supports what is claimed for both images. The most significant example is the light pink segment in Figure 2.7(a) which is a toolbox object and it is poorly merged in Figure 2.7(b). In Figure 2.7(c) and in Figure 2.7(d) over-merging example is given of the comparison of the two images. Numerical results are 26 segments and 73 segments with respect to Figure 2.7(c) and Figure 2.7(d). In Figure 2.7(d) a very large value is set for N_1, meaning that any segment which has an area less than 350 pixels will be merged with a neighbour segment. This caused an excessive merging between segments which resulted in loss of objects.

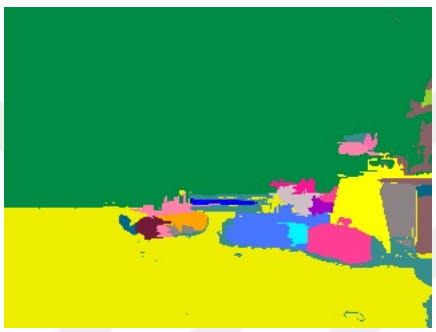
N_2 parameter effect can be seen in Figure 2.8. N_2 parameter directly affects the segment of interest. When it is greater than a certain value, parts that should have been merged may remain unmerged. The orange segment in Figure 2.8(a) is a good example for this. Parts of the object are not merged in Figure 2.8(b) because of the N_2 effect.



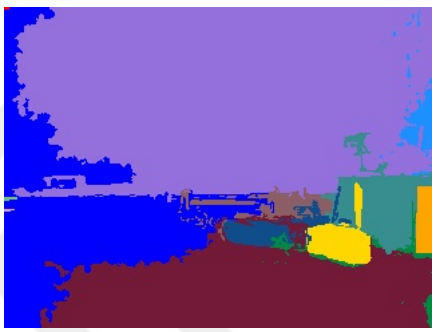
(a) $N_1 = 70$, $N_2 = 70$, $\sigma_D = 7$, 27 segments



(b) $N_1 = 10$, $N_2 = 70$, $\sigma_D = 7$, 73 segments



(c) $N_1 = 70$, $N_2 = 500$, $\sigma_D = 3$, 26 segments



(d) $N_1 = 350$, $N_2 = 500$, $\sigma_D = 3$, 13 segments

Figure 2.7. Effect of variations of parameter N_1 where $N_2 \in \{70, 500\}$ and $\sigma_D \in \{3, 7\}$.

A problem arises in segmenting images when the pixels which comprise an individual object in an image are not visually homogeneous. In our case the problem corresponds to color homogeneity. We partly solved this problem by parsing color space into quanta and by determining the color relationship (Table 2.1) in our method. As can be seen from parameter adjustment examples, we have a flexible method. For specific applications, it is an advantage to use a system which can be optimized by user. As this flexibility is an advantage for small applications, it is a disadvantage for larger ones. Optimized black box commercial methods for segmentation are much more suitable than our proposed algorithm for large-scale applications.

In the second group of experiments we compared our algorithm with two previous approaches - considering the images from COLD dataset as seen in Figure 2.10, Figure 2.11, Figure 2.12, Figure 2.13. The parameters of the proposed HBS algorithm



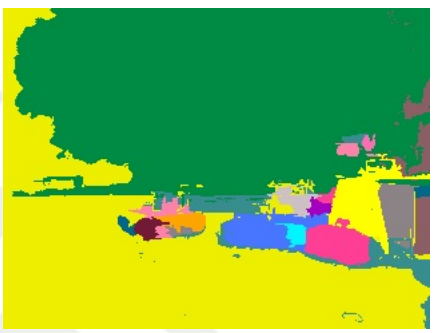
(a) $N_2 = 70$ and $\sigma_D = 3$, 31 segments



(b) $N_2 = 500$ and $\sigma_D = 3$, 26 segments



(c) $N_2 = 70$ and $\sigma_D = 7$, 27 segments



(d) $N_2 = 500$ and $\sigma_D = 7$, 24 segments

Figure 2.8. Effect of variations of parameter N_2 where $\sigma_D \in \{3, 7\}$ $N_1 = 70$.

are set as $N_1 = 70$, $N_2 = 500$ and $\sigma_D = 3$. In parallel, the parameters of the FH algorithm are set as $k = 150$, $min_size = 1000$ with $\sigma = 0.7$. Finally the parameters of the EDISON method are set as $h_r = 7$, $h_s = 6.5$, $min_reg = 20$. The results given in Figure 2.10, Figure 2.11, Figure 2.12 and Figure 2.13 are obtained with COLD images and results given in Figure 2.14 are obtained with ISL images. Timewise, the HBA algorithm runs in $1 - 2$ seconds. This is in comparison with FH that runs in 1 second or less than 1 second while the segmentation takes approximately 3.5 seconds with EDISON.

In Figure 2.10, images of the printer area room are used. If we look at Figure 2.10(a), we can see these objects : wall, ground, chair, bin, printer and a scanner. With the HBS algorithm, the wall, the ground and the chair are found and with the FH method, the bin, the scanner and the chair objects are determined. In Figure 2.10(b) window, table, ground, box, chair, closet, carpet and bin exist. The HBS method found a window, the chair, the carpet, the ground and a box where FH found a window, the

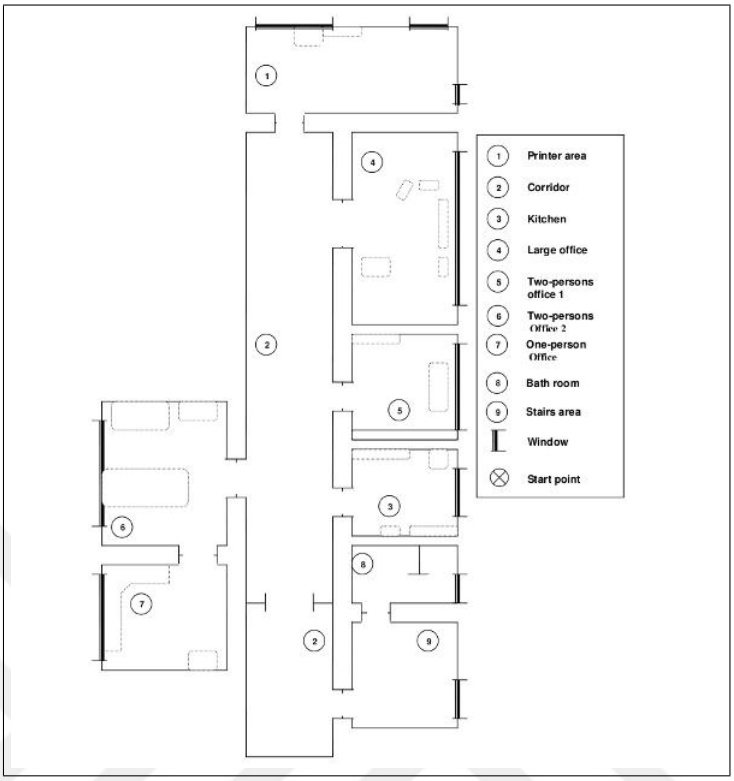
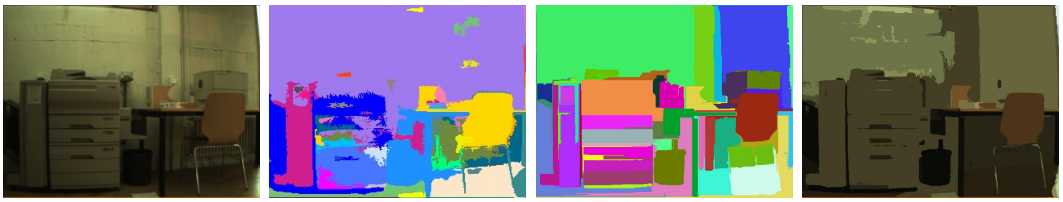


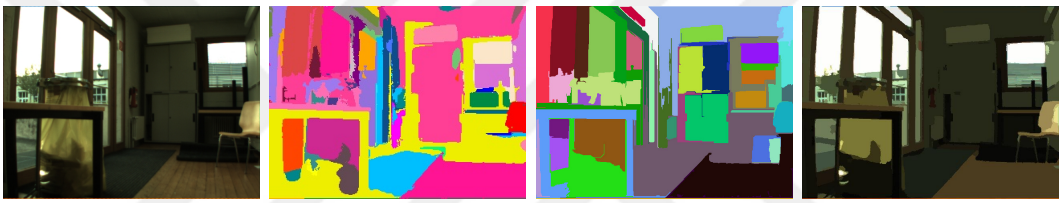
Figure 2.9. COLD - Freiburg Portion A, Extended Path Cloudy2

chair, the table, a box and a closet. Figure 2.10(c) contains window, ground, carpet, chair, box and closet. The HBS method was able to segment the window, the ground, the carpet, the chair and a box where FH was able to find the window, the chair, the box and the closet. In last scene of the printer area room (Figure 2.10(d)) a chair, a wall, a ground, a printer, a scanner and closets exist. The chair and the printer are determined successfully with our method where the chair and the wall are determined successfully with FH method. For this room, objects found with our algorithm and FH method mostly appeared to be common.

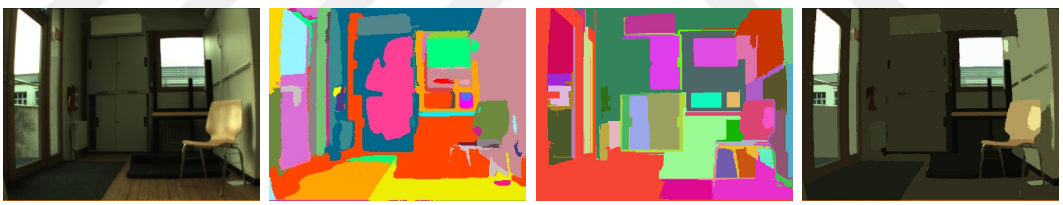
In Figure 2.11, corridor images are used. Since there are no significant objects in this scene, the overall performance of our method is expected to be low. Also, Figure 2.11(a), Figure 2.11(b) and Figure 2.11(c) are slightly different from each other. We see almost the same objects in these three images. A ceiling, a ground, doors, posters, pictures and a bin exist in those images. In Figure 2.11(a) two doors are successfully found with our method and two posters are successfully found with FH. In Figure 2.11(b) our method again found two doors and FH found a poster and a



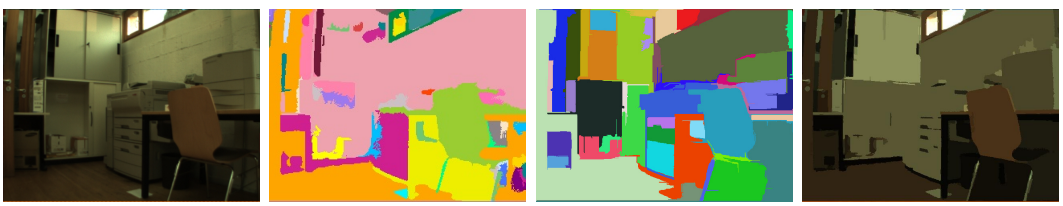
(a)



(b)



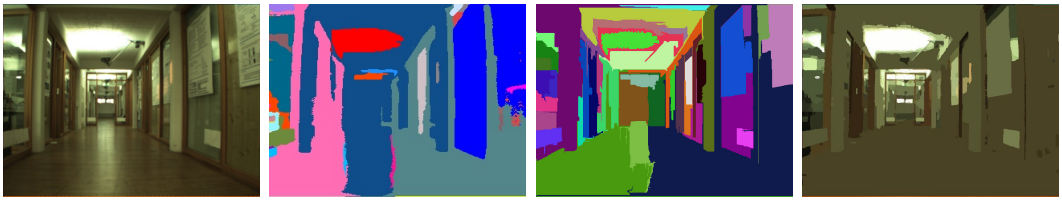
(c)



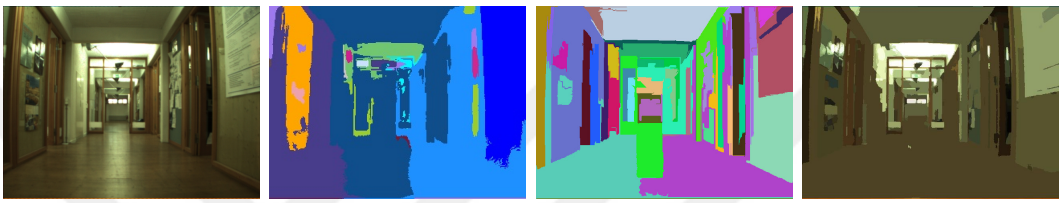
(d)

Figure 2.10. Comparative results for sample images from Area 1 (Printer area) scene: Left: Original image; Center-left: Proposed; Center-right: FH Method; Right: EDISON.

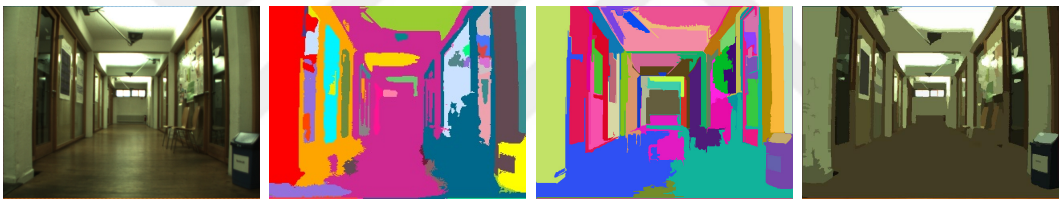
picture. A wall, a door and the bin found in Figure 2.11(c) with our method, whereas a wall, the bin and two posters are found with FH.



(a)



(b)



(c)

Figure 2.11. Comparative results for sample images from Area 2 (Corridor) scene: Left: Original image; Center-left: Proposed; Center-right: FH Method; Right: EDISON.

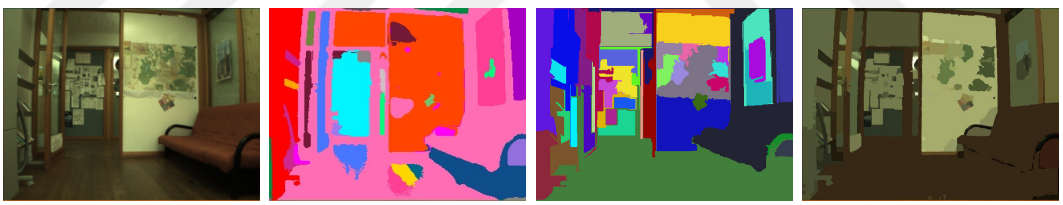
In Figure 2.12, images from the kitchen area are used. In Figure 2.12(a) and Figure 2.12(b) two windows, a floor, a wall, a closet, a box, a coffee machine, a poster and magazines exist. Our method found the box, the poster, a magazine and a window in Figure 2.12(a) and the FH approach found the box, the poster, the closet and the ground. In Figure 2.12(b) HBS found the box and a window where FH found the box, the coffee machine, the closet and two widows. Figure 2.12(c) contains a sofa, posters, a glass wall, a wall and a ground. Objects found with our method in this image are: the glass wall and the wall and objects found with FH are three posters and the wall. In Figure 2.12(d) our method found the glass wall and the wall where FH found three posters.



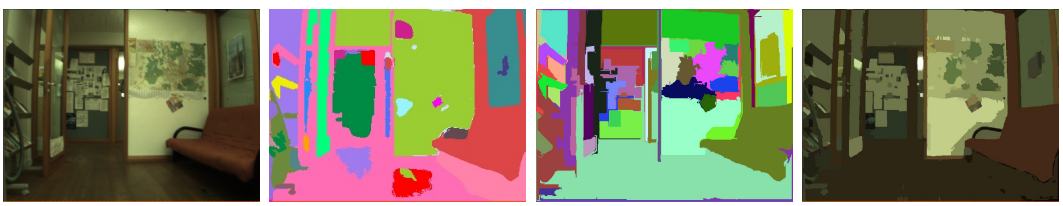
(a)



(b)



(c)



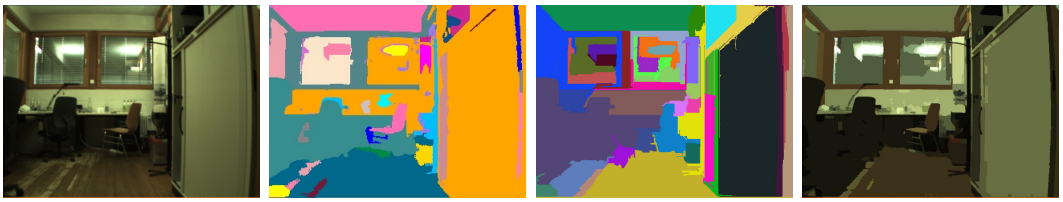
(d)

Figure 2.12. Comparative results for sample images from Area 3 (Kitchen) scene: Left: Original image; Center-left: Proposed; Center-right: FH Method; Right: EDISON.

Objects in the office images in Figure 2.13 are : office chairs, a monitor, posters, a lamp, a scarf, a closet, walls, a ground, a ceiling and windows. In Figure 2.13(a) three windows, the ceiling, the ground, the closet and one of the office chairs are found with our method and the ceiling, the scarf and the closet successfully found with FH. The ceiling, two windows and the scarf found in Figure 2.13(b) with our method where the scarf, the ceiling and one of the windows found with FH. In Figure 2.13(c) two windows, a wall and the lamp are found with HBS and objects found with FH are: the lamp and a wall. Only the monitor is found with our method in Figure 2.13(d) where the monitor and the lamp are found with FH in same image.

We used images from ISL database as shown in Figure 2.14 for the evaluation of our method and other methods in complex indoor scenes. If we look at Figure 2.14, the overall success of our method can be considered good in means of segmentation of object areas. EDISON gives rapid results but in means of object definition our results or FH algorithm results can be considered more useful. In segment reserving FH algorithm is better at some images and sometimes our method is better. If we consider the overall performance at complicated scenes, our method is superior to FH algorithm in means of object integrity.

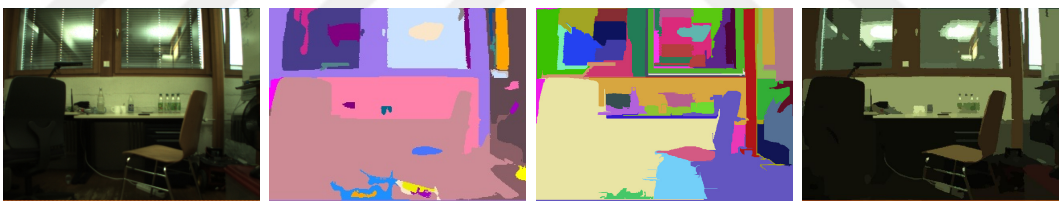
The results of segmentation analysis are shown in Table 2.2 and Table 2.3. The number of segments in an image is important as the segments indicate the objects in an image. As we use complex indoor scenes in our experiments we don't expect to find too many segments in an image. As shown in Table 2.2, the minimum number of segments found in an image is 226 with EDISON algorithm. This is a very large number for an indoor scene. The minimum number of segments found in an image with HBS and FH are 25 and 39 respectively. Those numbers are reasonable numbers for an indoor scene. The maximum number of segments found in an image is also shown in Table 2.2. Again, EDISON algorithm extracts too many segments, numerically 635 segments, which is far from being reasonable for an indoor scene. Such number would be reasonable only if there were around 400 objects in a scene. Number of segments found with our algorithm always very close to number of segments found with FH. Maximum number of segments found with HBS and FH are 65 and 53 respectively.



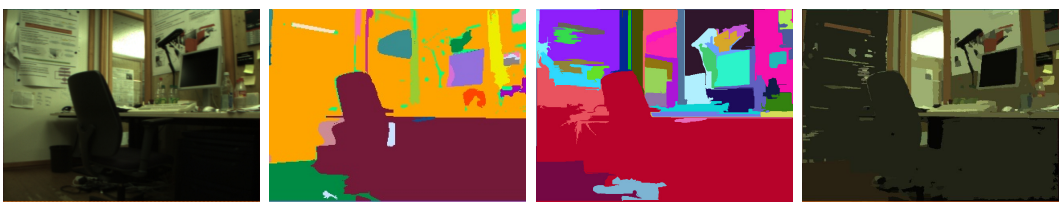
(a)



(b)



(c)



(d)

Figure 2.13. Comparative results for sample images from Area 5 (Two person office) scene: Left: Original image; Center-left: Proposed; Center-right: FH Method; Right: EDISON.

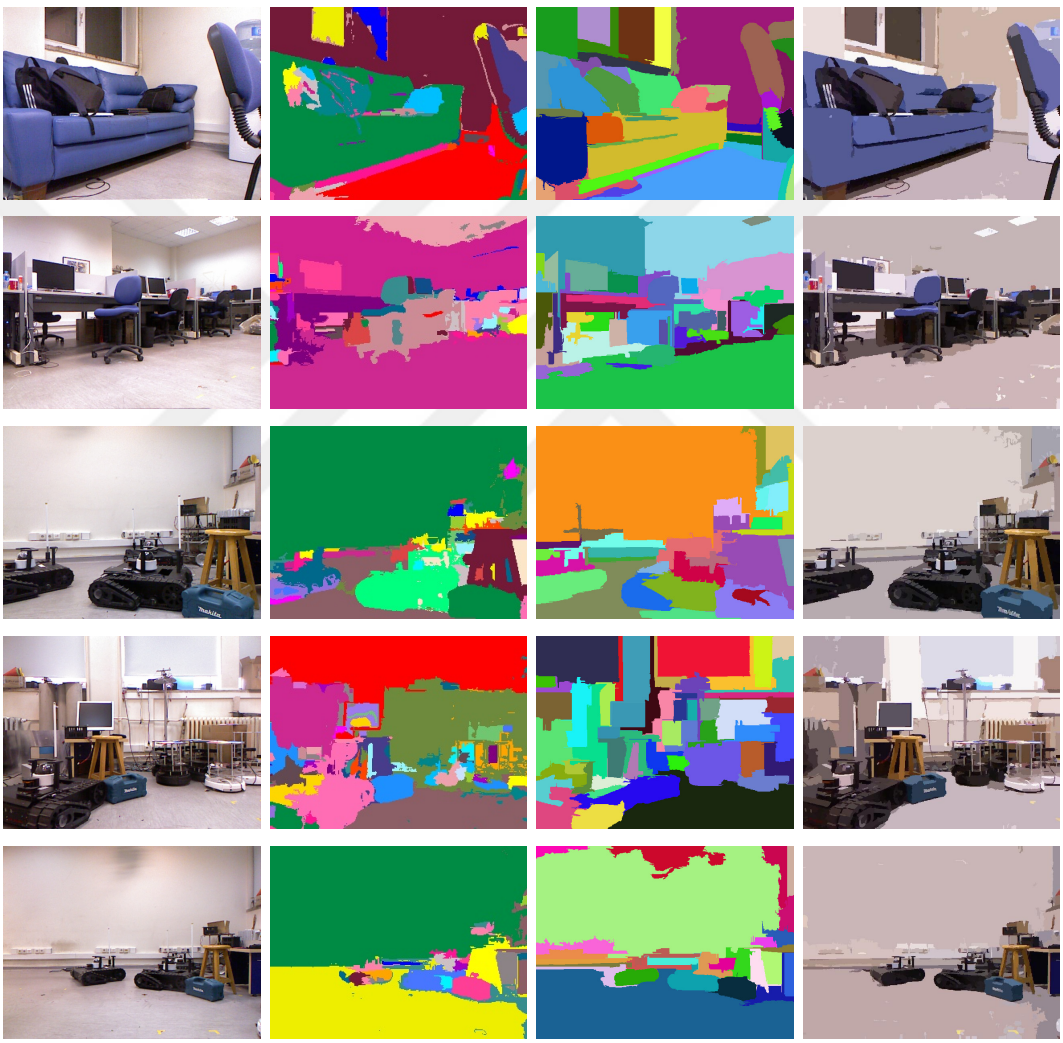


Figure 2.14. Comparative results for ISL scenes: Left: Original image; Center-left: Proposed; Center-right: FH Method; Right: EDISON.

According to those mathematical evidence, we can say that number of segments found with HBS and FH tend to be in a narrow range. The range of FH is narrower than HBS. Other measurements shown in this table are : mean, variance and standard deviation. The mean number of segments is found one by one for each algorithm. Although the number of segments range of FH is narrower than HBS, the means of both algorithms are very close : 43.80 for HBS and 45.26 for FH. EDISON has highest mean value of segments among comparative results: 454.73. Variance is described as the spread of the collection of segments from the mean number of segments. In order to evaluate results, we can use standard deviation which gives us insight about algorithms. The higher value means more variance in results. The standard deviation of number of segments in ascending sort are: 3.8, 11.18 and 104.27 regarding to FH, HBS and EDISON. According to these numbers we can say, FH, almost in every image has found the same number of segments and HBS in each image has found a segment in close numbers.

Table 2.2. Comparative results for Figure 2.10, Figure 2.11, Figure 2.12 and Figure 2.13

number of segments	HBS	FH	EDISON
total number of segments	657	679	6821
min	25	39	226
max	65	53	635
mean	43.80	45.26	454.73
variance	125.09	14.46	10874.06
stdev	11.18	3.80	104.27

In order to have a sense of accuracy of methods two methods, we consider object detection. Determining an object in complicated scenes is a tricky task. Even parts of an object cannot be defined precisely, you need to give a close definition. Object integrity refers to the ability to combine the segmented parts that belong to a single object in order to correctly identify the boundaries of the object. This ability to

Table 2.3. Object-segment analysis on Figure 2.10, Figure 2.11, Figure 2.12 and Figure 2.13

	Ground truth	HBS	FH
mean	7.86	3.2	3.2
stdev	1.88	1.61	1.08
variance	3.55	2.6	1.17

merge segments of a single object, called 'object integrity', is important as multiple segments may belong to the same object. Object integrity is measured based on visual confirmation and mathematical evidence (number of segments) in experiments. The results are represented in Table 2.3 is given. Ground truth column stands for notable object numbers in each image and in other columns meaningful segment numbers given for related method. Mean value shows us typical expected number of objects in this dataset. Standard deviation row indicates measure of the spread of object numbers within the dataset. As shown in Table 2.3, HBS and FH found same number of objects in average. However, both algorithms were not able to find every object in the scenes.

3. SEGMENT-BASED RECOGNITION

3.1. Introduction

The automatic recognition of objects of interest is an important task for the semantic understanding of a scene. Humans recognize objects in images easily, even they are partially obstructed from view. However, this task still poses a challenge for computer vision systems. One of the primary reasons is due to the significant variations in real-world images [29]. Ideal object recognition algorithms should be robust with respect to different appearances of the objects. Performance should be stable under changing object view point and changes in the lightning conditions. In this work, we consider recognizing objects in scenes. In particular, we use the detected segments for object recognition. They are learned using a supervised approach and the learned objects are used for object recognition.

The outline of this chapter is as follows: Related literature will be given in Section 3.2. In Section 3.3 proposed approach for object recognition will be explained in detail. Section 3.4 stands for giving general information about our segment-object relationship. In Section 3.4 segment representation will be given before SVM based learning and recognition. The closing section, Section 3.6, contains experimental results.

3.2. Related Literature

Object recognition is the task of processing an image to detect and to classify objects. Recognition of 3D objects from 2D images is a challenging task. A variety of different object recognition approaches have been proposed. Most share a common key concept. Mostly, the recognition of objects is performed using supervised classification techniques, requiring a set of labeled training samples, known as ground truth data. These are then used to extract different types of information based on the chosen algorithm [30]. Training samples are mostly labeled by a manual process, which is a very tedious task [31].

In most of the research, basic areas such as floor, ceiling and walls are tried to be found [32–36]. For the semantic relation between objects and environment, some other small objects are needed to be found. Indoor and outdoor objects vary as expected. For indoor applications mostly bed, table, chair, keyboard, monitor, sofa, stove, bathtub etc. need to be recognized as specific objects [34, 37, 38]. This is because they play a major role in interpreting semantic information about a room/environment. In outdoor applications some common objects like park, garden, road, grass, cow, cat etc. are related with the environment [39, 40]. Object recognition methods fall under six main headings : appearance based, model based, template matching, region based, feature based and neural networks.

Appearance based methods are probably the oldest methods among other methods. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA) are examples of appearance-based methods. Those approaches yield best results for scenes with unoccluded objects on black background [41]. Appearance based approaches do not require image features or geometric primitives to be detected and matched, hence they are desired. But those methods have some limitations. In order to obtain reasonable results with those methods, isolation of the complete object of interest from the background is required. Such requirement make them suitable mainly for certain applications under controlled environment. They are sensitive to occlusion and require good segmentation. Furthermore, keeping all possible object appearances of different backgrounds, occlusion and illumination, computationally inefficient.

In model-based object recognition, an object model is typically defined as to capture object's geometrical and appearance properties at the appropriate level of specificity [42]. Model matching algorithms recognize objects by finding a match between the models and images. For some applications where the goal is to find a specific shape, clustering methods are compared and the most accurate one is chosen. For RoboCup environment it is proved that Levenberg-Marquardt least squares circle fitting is more accurate and faster than circular Hough transform method for finding a tennis ball [43]. A good example of model matching and interpreting with semantic map results is done

in [33]. It describes an approach and an integrated robot system for semantic mapping. According to spatial and conceptual hierarchy, room names are interpreted from the object information [34, 37, 38]. Results show that interpreting room information like living room, bathroom, bedroom, kitchen is possible when you specify the most probable objects exist in there. Model-based object recognition methods are applicable to specific range of classes. Those methods cannot cope with the range of objects found in natural environments. Hence, they are robust among indoor scenes.

Template matching is done via sliding the template image across the scene image. It is a direct and simple process where template images for different objects are stored. When an image is given as input to the system, it is matched with the template images in database to determine the object in the input image. Templates are frequently used for recognition of characters, numbers, objects, etc [44]. For example, in [45], a mathematical morphological template matching approach for object detection is proposed. In this paper, the main purpose is to detect the objects on ground. In [45], with template matching method, the ground objects at different altitudes and orientation successfully recognized and detected. The disadvantage of this method is that it is only applicable if you know what you are looking for. You need to give accurate template in order to obtain good results. For example, template matching algorithm wouldn't be a good choice if the goal is to find small objects [32].

Region-based methods are basically segmentation based object detection methods. Regions can be segmented according to different properties. Those properties can be color, shape, texture etc. Studies also show that region-based object representations enable efficient and robust learning and inference of semantic information [46,47]. However, there is a cost trade-off between segments and objects [48]. Depending on the associated parameters, the number of segments may vary from a few to being very large. The former may result in all regions being merged together while with the latter, the scene might be split into too many segments.

Feature-based object recognition methods are extensively used methods in literature. Many state-of-the-art object recognition methods use local features like His-

tograms of Oriented Gradients (HOG) [49], Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) [50] and Local Binary Patterns (LBP) [51]. In LBP, local feature extraction process can be considered as dimension reduction for identifying most significant parts of the objects. If we make a comparison among feature-based object recognition methods, there are circumstances where they are advantageous and disadvantageous compared to each other. For example, SIFT is slow and doesn't give good results at illumination changes. But it is invariant to rotation, scale changes and affine transformations. SURF is fast and has good performance as the same as SIFT, but it is not stable to rotation and illumination changes [52]. There are also work that combine region and feature-based methods. For example, regions of interest are determined through active perception and then the most probable segmented area is named according to features [32]. In general, the feature extraction method is followed by class model construction which is generally accomplished with SVM.

Neural networks, especially CNNs have recently become very popular in object recognition. Training is done via multiple layers with this method [53]. The way of feature extraction and generalization from simple to complex is very similar to human perception of the scene. CNNs yield best results in generalization of object categories and classification tasks. Although CNNs are very stable and show the highest performance among other methods today, it has some disadvantages. This method is very demanding at training phase and memory usage. You need to train CNNs with a big training dataset and there are lots of parameters that needed to be set [54].

Our object recognition method falls into region-based recognition method category. First, we extract segments with our novel approach. Second, we recognize objects with our SVM based recognition algorithm. In our SVM based recognition algorithm we improved detection precision by using SVM recognition probabilities. Our work can be generalized to many other object classes because there is no specific constraint in feature selection. It can be easily modified according to object categories.

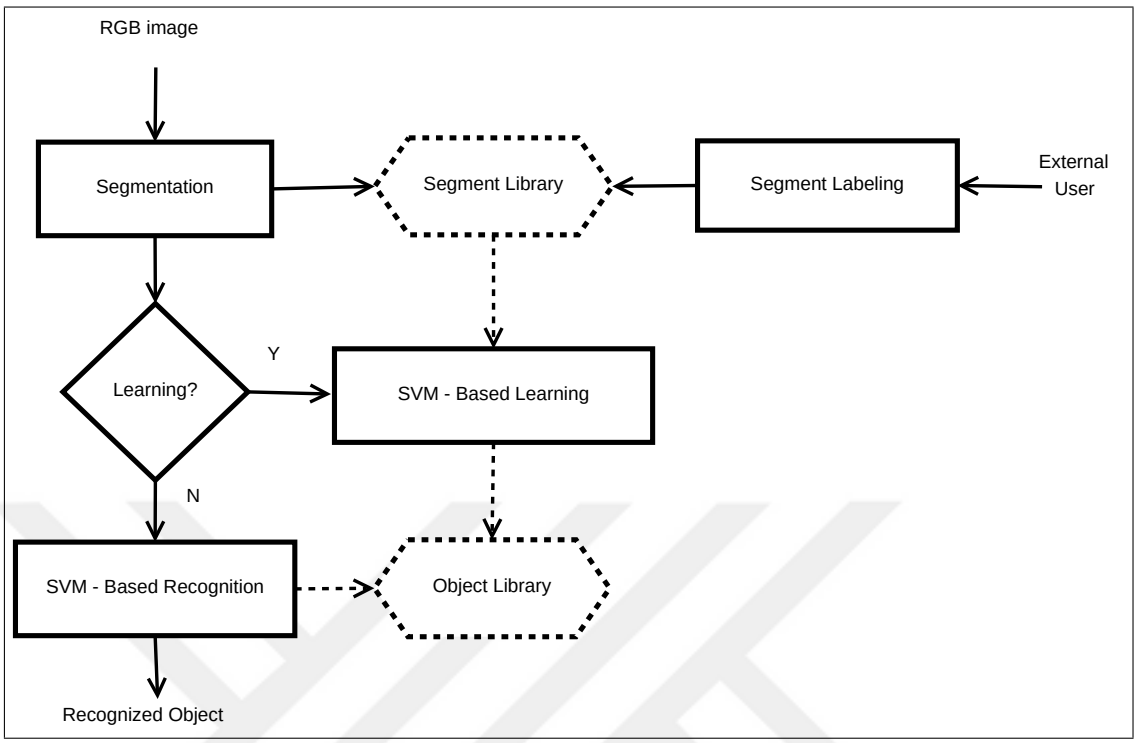


Figure 3.1. Recognition flow

3.3. Proposed Approach

The object recognition flow is as shown in Figure3.1. The approach is based on the segments that have been determined. All segments are registered to the segment library with their unique ID. There are two phases: Learning and recognition. In the learning phase, an external user first associates the learning segments with the object classes. This is followed by the SVM learning of each object class. The segments in the learning set are first encoded as descriptor vectors. Here, the descriptors are constructed using the vertical, horizontal, hyperboloid, polar and concentric visual filters [55]. Each class is learned with the same number of samples. In the recognition phase, the learned SVM classes are used to recognize each segment via assigning one of the learned object classes.

3.4. Segments - Objects Database

The segmentation of images with the method that we proposed in previous chapter is done and some of the segments eliminated according to their size. We defined

the threshold for elimination according to statistical results that are gathered from ISL database. We observed that there is no meaningful object that occupies area less than 0.001953% of an image. Each segment is represented by a descriptor vector $x \in \mathbb{R}^n$ that encodes its various features.

We use an extensive set of features in order to be able to distinguish between a wide range of objects. For example, the shape elongation feature is effective in distinguishing a wall and a turtlebot but it is not such sufficient to pick out the difference between the couch and the toolbox. As another example, hue value is efficient at distinguishing a wall and the couch but for a wall and a turtlebot it is vice versa. Hence, the following features are used:

- Segment center: The first feature is segment center. It indicates the relative position of the segment. For this feature, a complex scene image is theoretically divided into three proportions. Each proportion is represented by a number which can be seen in Table 3.1. This feature is important for our object recognition method because, objects aim to reside usually at the same places. For example, likelihood of finding a pen on a desk is higher than finding it on the floor. Thus, statistical results of the relative position of objects can give valuable information.

Table 3.1. Segment center

Above horizon	0.9
Horizon	0.6
Below horizon	0.3

In Figure 3.2 a complex scene image from ISL database is shown. The relative positions of some of the objects in Figure 3.2 are:

(i) Turtlebot, Ground : Below horizon

(ii) Jaguar, Toolbox, Monitor: Horizon

(iii) Wall : Above horizon

Statistical results for the average relative positions of the objects in this thesis are given at Table 3.2. Those statistical results are obtained on the ISL database



Figure 3.2. A complex scene image given with three proportions images for manually labeled segments.

Table 3.2. Average Segment Locations

1	Jaguar	0.36
2	Turtlebot	0.32
3	Chair	0.58
4	Wall	0.74
5	Monitor	0.72
6	Toolbox	0.33
7	Switch Box	0.9
8	Couch	0.6
9	Ground	0.3
10	Window	0.9
11	Others	N/A

- Shape ratio(elongation): Shape ratio is scale-invariant morphological feature that provides clues about the shapes of the objects. All objects have a minor and a major axis (Figure 3.3). The ratio of lengths of those axes tell us whether the object is round or elongated.

$$Shape\ ratio\ = \frac{minor\ \ axis\ \ length}{major\ \ axis\ \ length}$$

It is a distinguishing feature for the recognition of dissimilar objects. Of course this feature is not a unique feature. For example, the couch and the toolbox objects tend to have very similar shape ratios. On the other hand, a ground and a wall have a square-form in images. As the toolbox, the couch, a jaguar and a turtlebot haves elongated shapes, this feature is important to group such shapes. While there is a small difference between the toolbox, the couch, a jaguar and a turtlebot, there is huge difference between a wall and one of those objects.

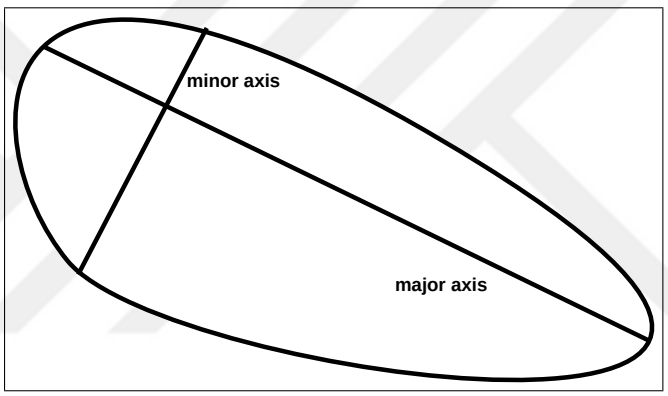


Figure 3.3. Major axis and minor axis

- Segment ratio: Segment ratio feature demonstrates the ratio between the area of the object and the whole image area.

$$Segment\ ratio = \frac{segment\ \ area}{total\ \ image\ \ area}$$

Consider Figure 3.4 as a scene image and the green area as an object in this scene. It can be seen from Figure 3.4 that this green object occupies 6870 pixels. The segment ratio for this object is $6870/76800 = 0.0894$.

Some objects are expected to have a large value as this feature value to show large area occupation. For example, a wall and a ground occupy very large areas, nearly half of the image and they cover larger area than other objects do. In Figure 3.5 examples for the area occupation percentage are given for different objects.



Figure 3.4. Segment ratio example



(a) Couch, 0.0819



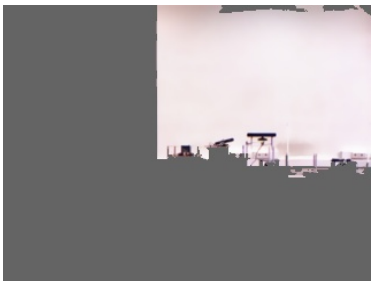
(b) Turtlebot, 0.0046



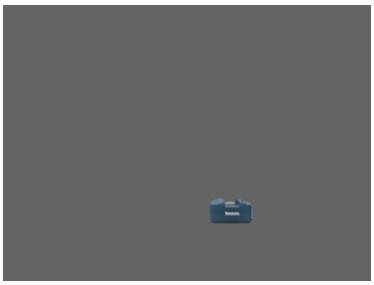
(c) Ground, 0.2851



(d) Monitor, 0.0061



(e) Wall, 0.3227

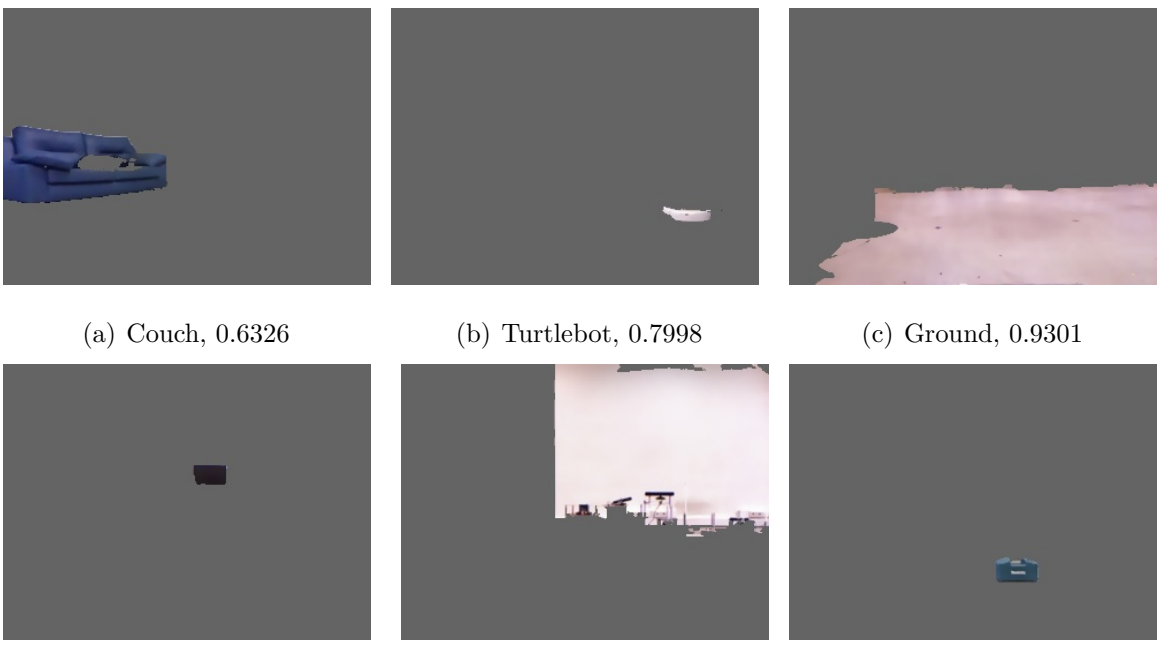


(f) Toolbox, 0.0094

Figure 3.5. Segment ratio for different objects

- Average Hue value: The average Hue value feature demonstrates the average hue color channel values on a segment. In the segmentation algorithm, we used HSV color space, hence we already have the hue value information for each segment. In order to calculate this feature, we need the segment area measurement. We use the total number of pixels in a segment as the segment area. Hue values of each pixel are known and we sum those values. Different applications use different scales for HSV. OpenCV uses H: $0-180$, S: $0-255$, V: $0-255$ ranges. The average hue value is calculated by dividing the sum of the hue values by the segment area. The normalized average hue value for each segment is obtained as follows:
  (i) Segment area : A
  (ii) Hue value : $h_n \in [0-180]$, $n \in [0, 1, ..., A]$
  (iii) Summation of hue values in segment : $\mathcal{H} = \Sigma\ h_n$
  (iv) Average hue value for a segment in OpenCV ranges : $\mathcal{H}/A$
  (v) Normalized average hue value of a segment: $\dfrac{\mathcal{H}/A}{180}$

  In Figure 3.6 objects with normalized hue values are given.



(a) Couch, 0.6326 (b) Turtlebot, 0.7998 (c) Ground, 0.9301

(d) Monitor, 0.8417 (e) Wall, 0.8153 (f) Toolbox, 0.5554

Figure 3.6. Normalized hue values for different objects

- Results of 54 spatial filters (Figure 3.9) on segment, [55]: The results of 54 spatial filters is another feature we use in our algorithm. Filters in Figure 3.9 are applied on each image. The 54 spatial filter results are critical. As histograms of filter results are used, the choice of histogram size was important. We needed an optimized size of histograms in order to carry valuable information without data pollution. A histogram size is decided on after experiments with different histogram sizes. An 8 bin histogram size is used for each filter. As there are 54 different spatial filters, the input vector obtained from those filters is a 432 dimensional vector.

Table 3.3. Effect of histogram size on validation test data

Histogram size	Test data size	Correctly labeled
3	55	51
8	55	53
12	55	53
16	55	53

Example results of filters for different objects are given in Figure 3.10. In Figure 3.10, Filter 10 responses for couch object and chair objects are given. As can be seen from this image, different objects give different responses for the same filter but identical objects give similar responses. Spatial filter responses represented with normalized histogram values. In Table 3.4, example filter responses for chair and jaguar objects are given. In Figure 3.7 and Figure 3.8 the graphical representations of data given in Table 3.4 is shown. In Figure 3.11 different filter responses for chair object are given.

Table 3.4. Example filter responses for different objects represented with normalized histogram values

Filter number	8 bin histogram values for chair	8 bin histogram values for jaguar
0	66,3,1,0,2,3,3,3	160,1,2,6,8,2,0,2
1	30,2,3,0,1,1,1,0	188,5,1,2,1,3,2,3
2	0,0,0,0,0,0,0,0	128,9,6,5,9,5,10,3
3	2,0,2,3,3,6,11,5	83,18,6,21,24,26,15,22
4	1,0,6,5,16,19,16,19	31,9,7,22,31,22,22,30
5	0,0,0,6,43,70,57,61	3,4,11,22,131,220,170,96
6	1656,24,15,34,57,49,38,43	936,1,1,0,0,0,1,2
7	1617,50,51,98,82,67,43,22	910,0,1,0,1,1,1,0
8	1729,147,124,46,44,24,34,26	887,2,3,1,1,4,1,1
9	1850,180,65,49,57,29,38,32	863,1,6,2,3,4,1,2
10	2188,153,105,55,39,20,1,8	837,16,19,5,4,7,4,3
11	2624,52,17,8,10,10,16,9	863,47,14,13,11,5,9,9
12	2882,3,1,3,4,0,1,0	961,2,0,2,3,2,1,2
13	2753,11,10,19,12,8,7,6	943,2,1,1,3,2,1,1
14	2900,0,0,0,0,0,0,0	985,2,0,0,1,0,0,1
15	2827,10,9,11,10,11,12,6	921,13,10,6,8,2,4,4
16	2887,10,3,0,0,0,0,0	968,12,6,4,6,4,0,0
17	2900,0,0,0,0,0,0,0	1004,0,0,0,0,0,0,0
18	1962,18,25,39,24,29,43,51	781,5,3,3,7,4,4,0

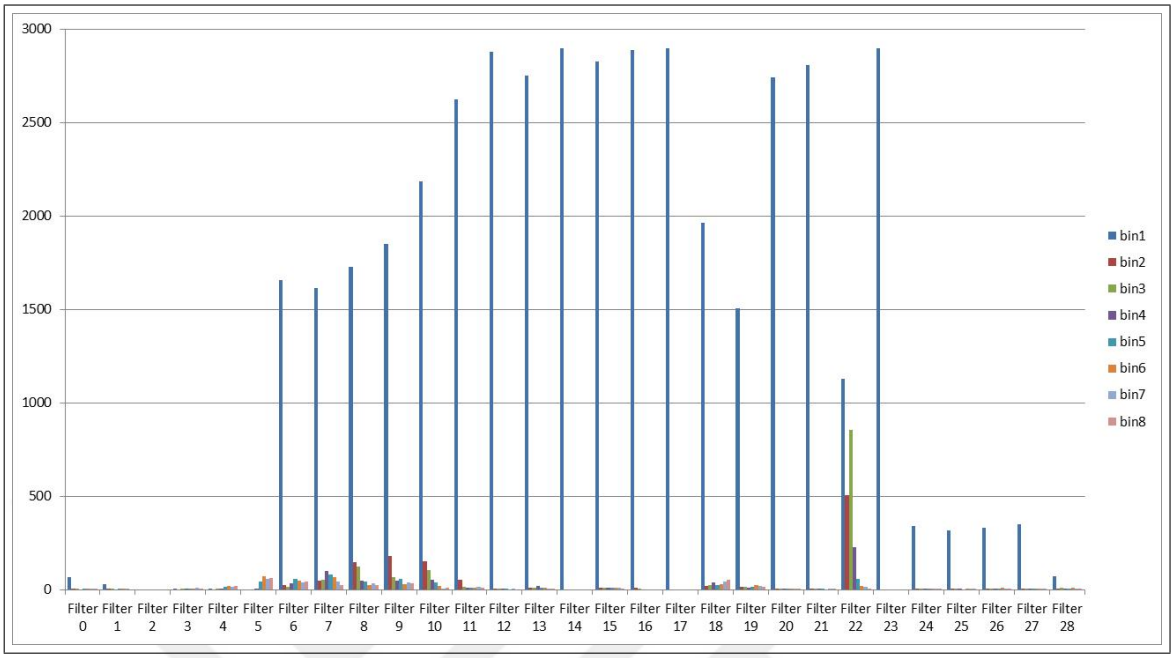


Figure 3.7. 8 bin histogram values for chair

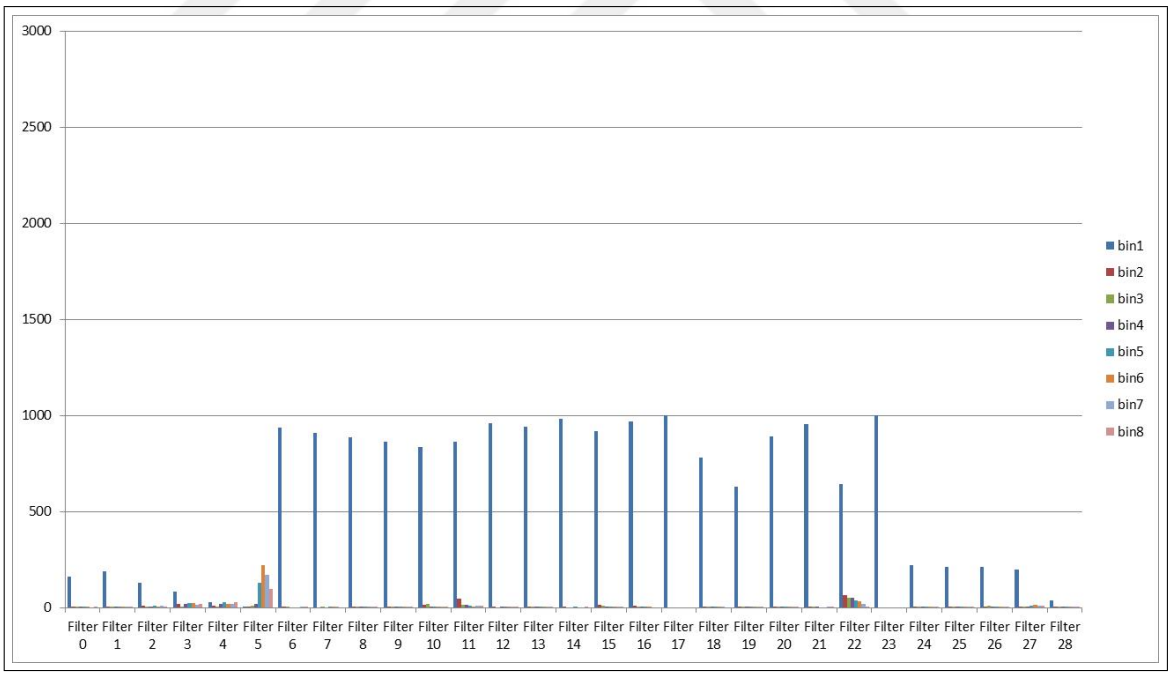


Figure 3.8. 8 bin histogram values for jaguar

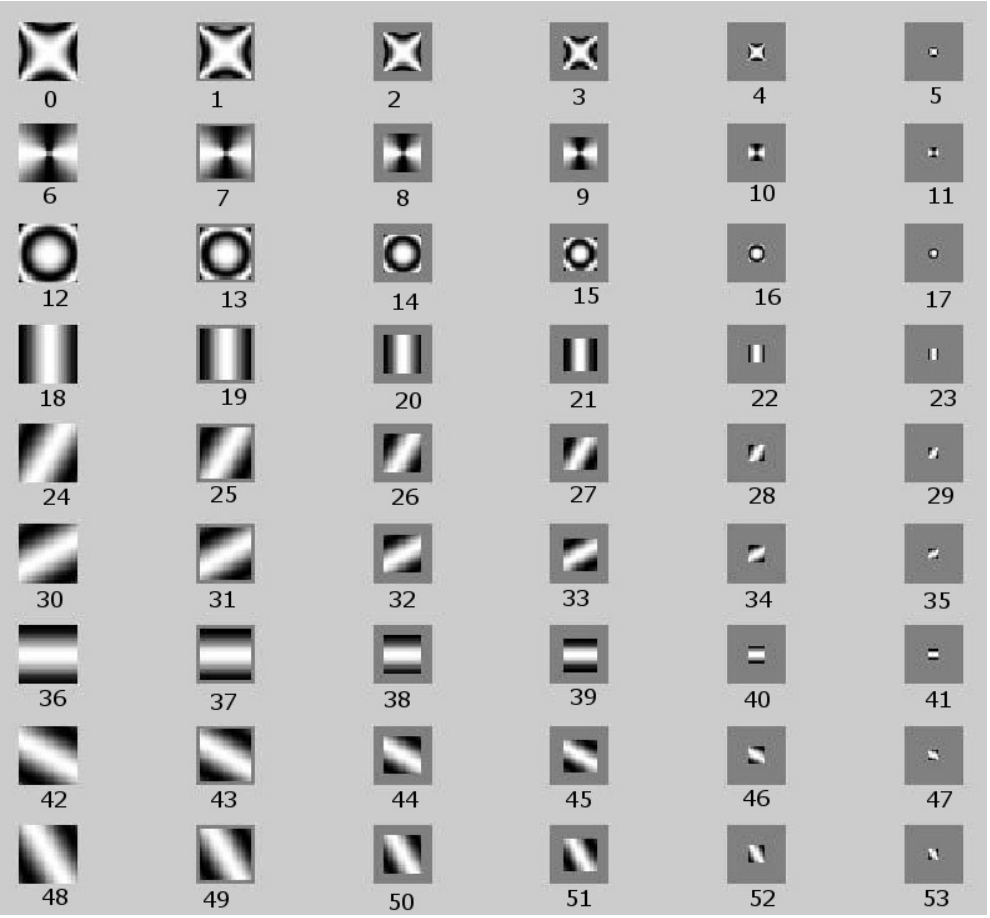


Figure 3.9. 54 Spatial filters [55]

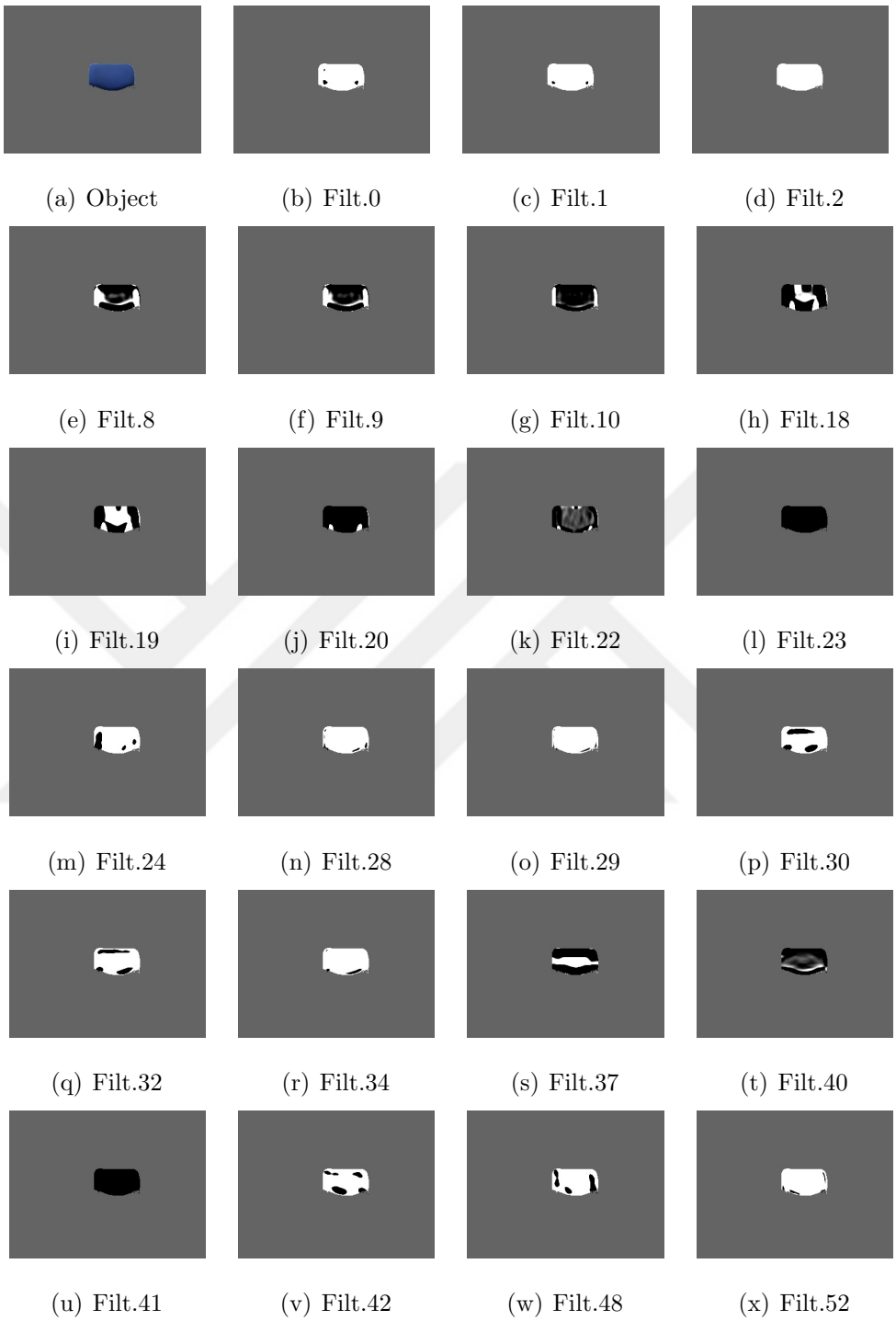


(a) Filt.10, Couch (b) Filt.10, Couch (c) Filt.10, Chair



(d) Filt.10, Chair (e) Filt.1, Jaguar (f) Filt.1, Jaguar

Figure 3.10. Example filter responses for different objects



(a) Object (b) Filt.0 (c) Filt.1 (d) Filt.2

(e) Filt.8 (f) Filt.9 (g) Filt.10 (h) Filt.18

(i) Filt.19 (j) Filt.20 (k) Filt.22 (l) Filt.23

(m) Filt.24 (n) Filt.28 (o) Filt.29 (p) Filt.30

(q) Filt.32 (r) Filt.34 (s) Filt.37 (t) Filt.40

(u) Filt.41 (v) Filt.42 (w) Filt.48 (x) Filt.52

Figure 3.11. Example filter responses for chair

In summary, each segment is represented by a 436-dimensional descriptor with components as given in Table 3.5.

Table 3.5. Features

Feature	Size
Segment center	1
Shape ratio (elongation)	1
Segment ratio	1
Average Hue value	1
Filter responses	432
$x \in \mathbb{R}^n$ feature descriptor	436

3.4.1. SVM Based Learning

SVM is a well-known and extensively used method for object recognition. Basically, SVMs can only solve binary classification problems. In classical SVM learning, a classifier function g_i is built for each class to be learned where w is the weight vector, x_i is the feature descriptor and b is the offset constant.

$$g_i = w^T x_i + b \tag{3.1}$$

In this thesis, we use multi-class implementation of SVM in LibSVM [56]. Multi-class implementation of LibSVM uses one-against-one approach. For learning, SVM classifiers are constructed and used to classify objects. If N is the number of classes, then $N_C = \frac{N(N-1)}{2}$ classifiers are trained to classify objects. Each classifier trains data from two classes. For training data from the ith and the jth classes, given the training vectors, x_t, LibSVM solve the following two-class classification problem:

$$\begin{aligned} \underset{w^{ij},b^{ij},\xi^{ij}}{min} \quad & \frac{1}{2}(w^{ij})^T w^{ij} + \mathcal{C}\sum_t (\xi^{ij})_t \\ \text{subject to} \quad & (w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij} \quad \text{,if } x_t \text{ in the } i^{\text{th}} \text{ class,} \\ & (w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij} \quad \text{,if } x_t \text{ in the } j^{\text{th}} \text{ class,} \\ & \xi_t^{ij} \geq 0. \end{aligned} \tag{3.2}$$

where $\phi(x_t)$ maps x_t into a higher dimensional space and $\mathcal{C}$ is the tradeoff constant, ξ_t^{ij} is the slack variable. w^{ij} is the weight vector and b^{ij} is the offset constant for the given two-class classification problem.

In classification, a voting strategy is used. Each binary classification is considered to be a voting where votes can be cast for all data points x - in the end a point is designated to be in a class with the maximum number of votes. SVM predicts only class label without probability information by default but LibSVM uses an implementation for extending SVM in order to calculate probabilities [57, 58]. Probability calculation is given below. N is the number of classes and the goal is to estimate p_i for any x.

$$p_i = P(y = i|x), \;\; i = 1, ..., N \tag{3.3}$$

After setting of the one-against-one approach for multi-class classification, first pairwise class probabilities estimated:

$$r_{ij} \sim P(y = i|y = i \;\; or \;\; j, x) \tag{3.4}$$

In Equation 3.4, i and j are stand for representing the related classes and an improved implementation of [59] is used [60]. If $\hat{f}$ is the decision value at x, then the following assumption is made:

$$r_{ij} \sim \frac{1}{1 + e^{A\hat{f}+B}} \tag{3.5}$$

In Equation 3.5, A and B are the objective parameters [60] and they are estimated by minimizing the negative log likelihood of the training data.

After collecting all probability estimations, r_{ij}, distribution of probabilistic inferences, p_i, generated. In LibSVM, the following optimization problem is considered.

This approach is used to obtain $p_i, \forall i$:

$$\begin{aligned} \min_{p} \quad & \frac{1}{2} \sum_{i=1}^{k} \sum_{j:j \neq i} (r_{ij} p_i - r_{ij} p_j)^2 \\ \text{where} \quad & p_i \geq 0, \forall i \end{aligned} \tag{3.6}$$

Sum of the probabilities of each class prediction for a test sample is always equal to 1 (Equation 3.7).

$$\sum_{i} p_i = 1 \tag{3.7}$$

The probability inference for a given set of data is calculated with a simple iterative method. More implementation details for LibSVM are given in [57].

3.5. SVM Based Recognition

The decision making algorithm is based on SVM recognition probabilities. For multi-class object recognition problem, SVM calculates all probabilities of each object class for given test data. The most common approach is to use the class with highest probability as the classification result. The greatest number indicates the most probable one and it is returned as predicted label.

A sample case is given in Table 3.6. In this case, the predicted label would be 5. However, this may lead to wrong decisions. That's why, for multi-class recognition problems in computer vision, posterior probabilities are widely used to make decisions in algorithms [61]. A threshold for these algorithms can be set to reject unknown classes. In our decision making algorithm we applied this idea. We make an assumption for low recognition probabilities because low recognition probabilities mostly mean unknown objects in our case. Table 3.7 is given as example for this assumption rule in our decision making algorithm. SVM returns label 9 with highest probability but the highest value is less than our threshold, τ_3. The decision for this segment is $o^* = 11$. Due to experimental results, we also defined other rules in decision making. We defined

Table 3.6. Probabilities

Class Nr.	Probabilities scaled [0-1]
1	0.038090
2	0.006203
3	0.205763
4	0.022492
5	0.429298
6	0.038183
7	0.129432
8	0.025104
9	0.006093
10	0.043601
11	0.055741

Table 3.7. Decision example for low recognition probability, rule # 4

Class Nr.	Probabilities scaled [0-1]
1	0.016274
2	0.160025
3	0.010550
4	0.094950
5	0.009645
6	0.023954
7	0.021574
8	0.060542
9	0.313841
10	0.037261
11	0.251384

one of the rules taking in consideration the probability measurement for 'other object' class. If probability of predicted label is greater than a certain value, τ_2, but is less than a certain value, τ_3, we check the probability of 'other object' class. In this case, the probability of other object class gives us clue about the real label for the test data. With this rule, we reduced false positives significantly. Further explanations are given with rules accordingly. Sample case is given in Table 3.8. In this example, correct label of sample segment is 11 but SVM returns label 10. The decision with our algorithm is $o^* = 11$ for this example. Our algorithm corrects such label decisions. As such, a

Table 3.8. Decision example for rule # 2

Class Nr.	Probabilities scaled [0-1]
1	0.009817
2	0.037706
3	0.020037
4	0.078427
5	0.018557
6	0.015014
7	0.105839
8	0.041748
9	0.130000
10	0.402055
11	0.140800

different approach has been used. First, consider the following definitions.

- o^* : The class with the highest probability.
- $p_{N_{o*}}$: Probability of $o*$ class, $o = \{1, \ldots, 11\}$
- $p_{N_{11}}$: Probability for undefined object class
- τ_1: Probability threshold
- τ_2: Probability threshold
- τ_3: Probability threshold
- $\hat{o}$: Decision

The rules for decision making are as follows:

(i) $p_{N_{o*}} > \tau_3 \rightarrow \hat{o} = o*$

If probability of predicted label is greater than τ_3 we assume that this prediction is correct.

(ii) $\tau_2 < p_{N_{o*}} \leq \tau_3$ & $p_{N_{11}} > \tau_1 \rightarrow \hat{o} =$ Other

If probability of predicted label is greater than a certain value, τ_2, but is less than a certain value, τ_3, we check the probability of 'other object' class. If $p_{N_{11}}$ is greater than τ_1, label of the test sample is equal to 11 which is representation of 'other object' class.

(iii) $\tau_2 < p_{N_{o*}} \leq \tau_3$ & $p_{N_{11}} \leq \tau_1 \rightarrow \hat{o} = o*$

If probability of predicted label is greater than a certain value, τ_2, but is less than a certain value, τ_3, we check the probability of 'other object' class. If $p_{N_{11}}$ is less than τ_1, we assume that predicted label is correct.

(iv) $o* \leq \tau_2 \rightarrow \hat{o} =$ Other

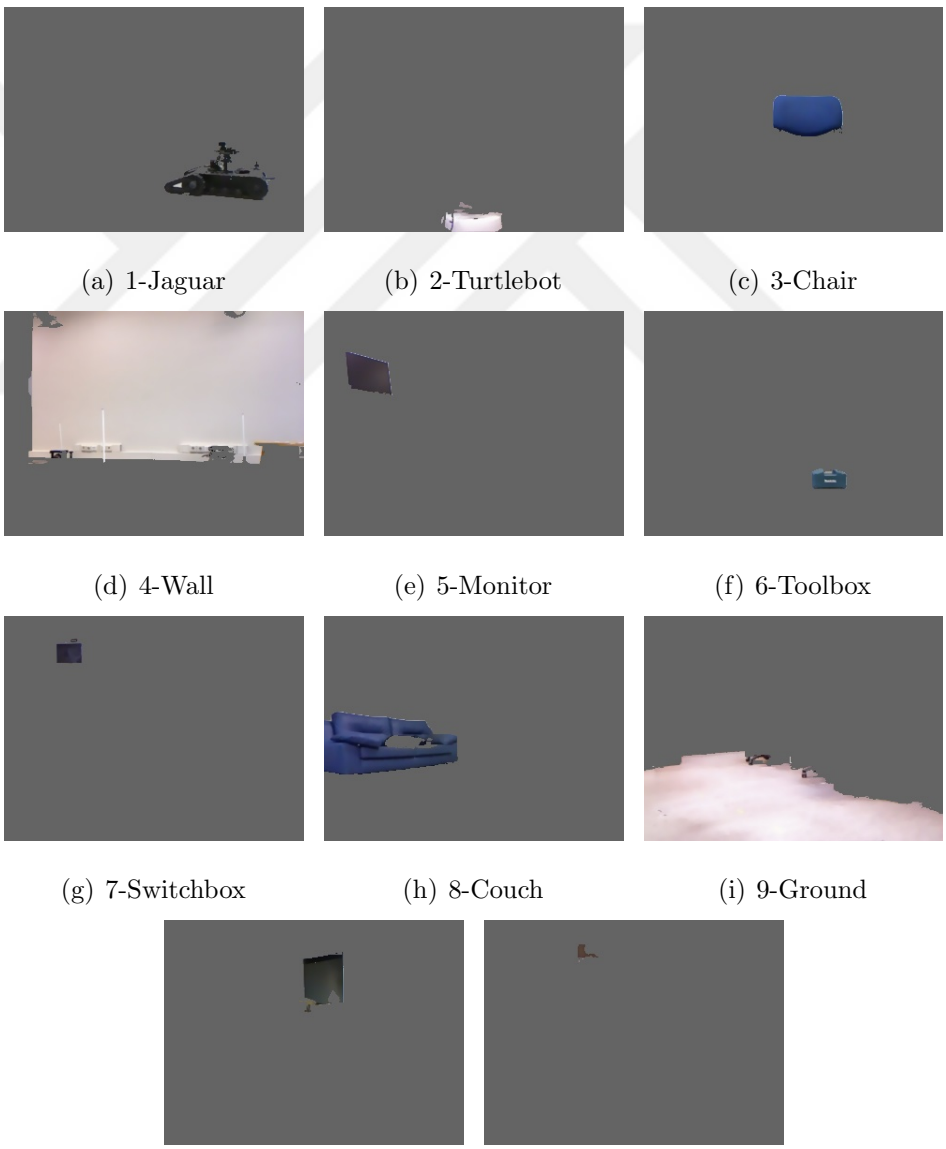
If probability of predicted label is less than τ_2, we assume that this prediction is not correct and the label of the test sample is equal to 11 which is representation of 'other object' class.

For example, in the case as given in Table 3.6, the decision making would be as given. In this case rule # 3 handles, decision is $o^* = 5$. First note that $p_{N_5} = 0.429298$, $p_{N_{11}} = 0.055741$. Also note that $p_{N_5} < \tau_3 \rightarrow \hat{o} = 5$ and $p_{N_{11}} < \tau_1$. Since, $\tau_2 < p_{N_5} \leq \tau_3$ and $p_{N_{11}} \leq \tau_1 \rightarrow \hat{o} = o*$.

3.6. Experimental Results

Experiments are conducted considering indoor scenes from the ISL database. ISL database images are originally 640×480. We apply a down-sampling algorithm in the preprocessing step (Figure 2.1) and we use 320×240 images. We have conducted two sets of experiments. In the first set, segments are provided. The first set is to validate the approach as such, segmentation results are manually selected before recognition. In the second set, segmentation results are fed automatically.

The object library is constructed with 10 object classes and 1 other class for unknown objects, $y \in 1, 2, ..., 11$. In Figure3.12 objects are given. We constructed our SVM model according to 11 object classes with 100 samples per class. Training samples are not used in testing in either experiment set. The training takes around 1 hour on Windows Vista (64-bit) with Intel Core Duo CPU 3.00 GHz and RAM 4.00 GB machine. The most time consuming parts are segmentation and extraction of features. After feature extraction, we keep those values and use them in SVM training. SVM training itself takes around 3 minutes.



(a) 1-Jaguar (b) 2-Turtlebot (c) 3-Chair

(d) 4-Wall (e) 5-Monitor (f) 6-Toolbox

(g) 7-Switchbox (h) 8-Couch (i) 9-Ground

(j) 10-Window (k) 11-Others

Figure 3.12. Segments and the objects they represent

In the recognition experiments, the parameters are set as $\tau_1 = 0.1$, $\tau_2 = 0.35$ and $\tau_3 = 0.45$. We also compare our object recognition method with a state-of-the-art method, Caffe [54]. Caffe is a deep learning framework and is used for computer vision applications. Convolutional Neural Networks (CNNs) are used in this framework and it is widely used in recent research. As we do in our method, Caffe constructs a model for classification and uses this model for object recognition. The training model with Caffe takes approximately 3 hours (GPU). The segments are input to the system and the rest of the model training is done with this framework.

3.6.1. Validation Experiments

In these experiments, 18 test samples for each class are used. Experimental results are given in Table 3.9. The column named 'Correctly labeled' represents the number of samples that are labeled correctly for each object class. The results are given in Table 3.9. As expected, CNN has better overall performance than ours. In particular, its accuracy is $80,3\%$ accuracy while that of our method has $79,79\%$ accuracy.

Table 3.9. Comparative recognition performance

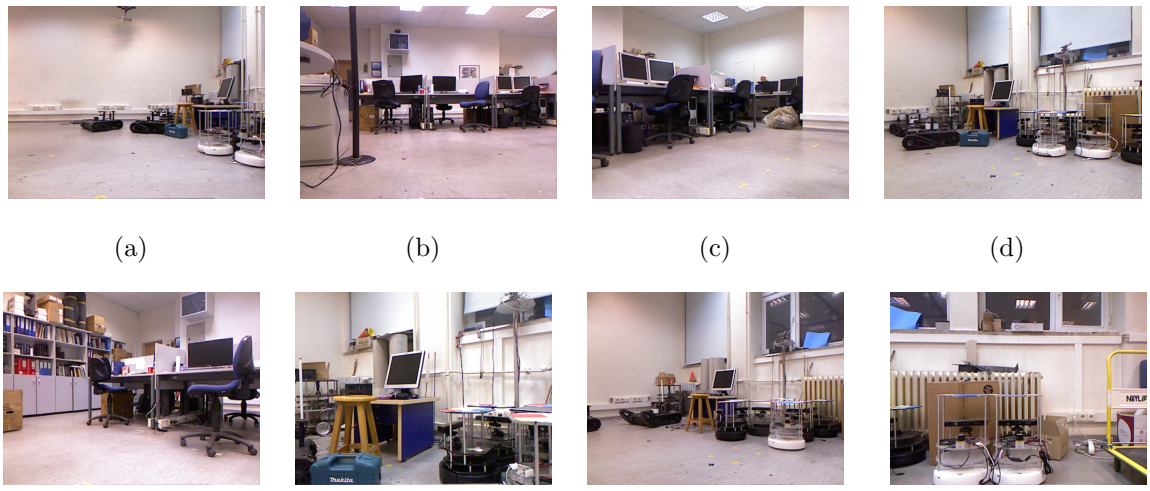
	Correctly labeled		Accuracy	
Objects	Our method	CNN	Our method	CNN
Jaguar	14	15	0.77	0.83
Turtlebot	12	13	0.66	0.72
Chair	14	12	0.77	0.66
Wall	17	16	0.94	0.88
Monitor	16	15	0.88	0.83
Toolbox	15	14	0.83	0.77
Switch Box	15	15	0.83	0.83
Couch	13	14	0.72	0.77
Ground	18	17	1	0.94
Window	11	12	0.61	0.66
Others	13	16	0.72	0.88

3.6.2. Integrated Object Detection & Recognition

In these experiments, object detection and recognition are integrated together.

For auto-search, images are given to the system and by using the steps given in Figure 3.1 object recognition is done. For this part of experiments, 40 images from ISL database are used. Some of the images used in auto-search test are given in Figure 3.13. The segmentation process results in 1210 segments in total from 40 images. Each segment is labeled with the both method mentioned in the experimental results section for comparative results. For further analysis, labeled segments are checked manually and the precision, the recall and the accuracy rates defined accordingly. The results are presented in Table 3.10. In this table, the number of segments can be found that belong to each terminological classification which obtained with our given method. In Table 3.10, results obtained from the CNN model are also given. Statistical results given in Table 3.12 are derived from Table 3.10.

Similar to validation data test results, our method displays good performance, very close to CNN in auto-search test set. The following statistical values were obtained in our method: accuracy of 86,19%, precision of 58,43% and recall of 86,99%. With CNN method, an accuracy of 89,33%, precision of 65,06% and recall of 91,03% are obtained.



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

(e) (f) (g) (h)

Figure 3.13. Example images used for auto-search test.

The specific definitions for TP,FN,FP,TN for this thesis are given at Table 3.11.

Table 3.10. Our method vs CNN

	Our method	CNN
TP	194	203
FN	29	20
FP	138	109
TN	849	878
Total	1210	1210

Table 3.11. Specific Definitions

		Predicted Class	
		Predicted Condition True	Predicted Condition False
Real class	Condition Positive	Real Class : 1-10 Predicted Class : True	Real Class : 1-10 Predicted Class : False
	Condition Negative	Real Class : 11 Predicted Class : 1-10	Real Class : 11 Predicted Class : 11

Table 3.12. Statistical results for experiments. Precision, recall and accuracy definitions are given in Equation 3.8, 3.9 and 3.10.

	Our method	CNN
Precision	58,43 %	65,06 %
Recall	86,99 %	91,03 %
Accuracy	86,19 %	89,33 %

$$Precision = \frac{TP}{TP + FP} \tag{3.8}$$

$$Accuracy = \frac{TP}{Total\ nr.\ of\ segments} \tag{3.9}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.10}$$

4. CONCLUSION

This thesis considers segmentation and SVM-based object recognition. Object recognition is a challenging and important task in the area of computer vision due to its necessity for more complicated tasks like object localization and classification of images. It is important to process complex scenes with optimized algorithms and obtain results as fast as possible.

Although state-of-the-art methods offer more standardized results and high precision, they also have some disadvantages. Especially, if there is lack of samples, state-of-the-art methods tend to fail. As we have compared our method under same training conditions with a CNN algorithm, CNN showed better performance for real-life application. State-of-the-art methods are not always the best solution for small-scale applications but it is not eligible for our method. If we have provided less sample to CNN our method would show better performance. But in order to make comparison with a state-of-art method we just had to train our method with more samples.

It is important to suggest an optimized solution for every application or need. The advantages of our method are, firstly it is efficient in time and secondly, it doesn't require too many samples for training. In addition to that, it is much more flexible than CNN. You can interfere with all steps starting from segmentation to object recognition. The method proposed in this thesis can be used application specific and can fulfill the needs of a specific application.

Instead of using a black box method, our approach to object recognition can be used.

REFERENCES

1. Dass, R. and S. Devi, "Image Segmentation Techniques 1", *International Journal of Electronics and Communication Technology*, 2012.

2. Chen, Q., X. Yang and E. M. Petriu, "Watershed segmentation for binary images with different distance transforms", *Haptic, Audio and Visual Environments and Their Applications, The 3rd IEEE International Workshop*, pp. 111–116, 2004.

3. He, L., Y. Chao and K. Suzuki, "A new two-scan algorithm for labeling connected components in binary images", *Proceedings of the World Congress on Engineering*, Vol. 2, pp. 1141–1146, 2012.

4. Yogamangalam, R. and B. Karthikeyan, "Segmentation techniques comparison in image processing", *International Journal of Engineering and Technology (IJET)*, Vol. 5, No. 1, pp. 307–313, 2013.

5. Dhivyaa, C. and R. Suganya, "A Survey On Image Segmentation Techniques", *International Journal of New Technologies in Science and Engineering*, Vol. 1, No. 3, pp. 2349–0780, 2014.

6. Banimelhem, O. and Y. A. Yahya, "Multi-Thresholding Image Segmentation Using Genetic Algorithm", *World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2011.

7. Sural, S., G. Qian and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval", *IEEE International Conference on Image Processing.*, Vol. 2, pp. II–589, 2002.

8. Lei, Z., L. Fuzong and Z. Bo, "A CBIR method based on color-spatial feature", *TENCON 99. Proceedings of the IEEE Region 10 Conference*, Vol. 1, pp. 166–169, 1999.

9. Smith, J., *Integrated spatial and feature image systems: retrieval, compression and analysis*, Ph.D. Thesis, Columbia University, 1997.

10. Haifeng, S. and L. Lanlan, "Clustering Color Image Segmentation Based on Maximum Entropy", *The 2nd International Conference on Computer Application and System Modeling*, pp. 1466–1468, 2012.

11. Deng, Y., C. Kenney, M. S. Moore and B. Manjunath, "Peer group filtering and perceptual color image quantization", *IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 21–24, 1999.

12. Alsultanny, Y., "Color image segmentation to the RGB and HSI model based on region growing algorithm", *Proceedings of the 4th WSEAS International Conference on Computer Engineering and Applications*, pp. 63–68, 2010.

13. Preetha, M. M. S. J., L. P. Suresh and M. J. Bosco, "Image segmentation using seeded region growing", *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, pp. 576–583, IEEE, 2012.

14. Abbas, K. and M. Rydh, "Satellite Image Classification and Segmentation by Using JSEG Segmentation Algorithm", *International Journal of Image, Graphics and Signal Processing*, Vol. 4, No. 10, p. 48, 2012.

15. Tejaswini C, M. Y., "Region Based Satellite Image Segmentation Using JSEG Algorithm", *International Journal of Computer Science and Mobile Computing*, Vol. 4, No. 5, pp. 1012–1016, 5 2015, an optional note.

16. Jabari, S., Y. Zhang and A. Suliman, "Stereo-based building detection in very high resolution satellite imagery using IHS color system", *IEEE International Geoscience and Remote Sensing Symposium*, pp. 2301–2304, 2014.

17. Tilton, J. C., Y. Tarabalka, P. M. Montesano and E. Gofman, "Best merge region-growing segmentation with integrated nonadjacent region object aggregation",

IEEE Transactions on Geoscience and Remote Sensing, Vol. 50, No. 11, pp. 4454–4467, 2012.

18. Wang, Y., J. Yang and Y. Zhou, "Unsupervised color-texture segmentation", *Image Analysis and Recognition*, pp. 106–113, Springer, 2004.

19. Christoudias, C. M., B. Georgescu, P. Meer and C. M. Georgescu, "Synergism in Low Level Vision", *International Conference on Pattern Recognition*, pp. 150–155, 2002.

20. Wangenheim, A., R. F. Bertoldi, D. D. Abdala, A. Sobieranski, L. Coser, X. Jiang, M. M. Richter, L. Priese and F. Schmitt, "Color image segmentation using an enhanced Gradient Network Method", *Pattern Recognition Letters*, Vol. 30, No. 15, pp. 1404–1412, 2009.

21. Wang, X., H. Li, C.-E. Bichot, S. Masnou and L. Chen, "A graph-cut approach to image segmentation using an affinity graph based on ℓ 0-sparse representation of features", *20th IEEE International Conference on Image Processing.*, pp. 4019–4023, 2013.

22. Arbelaez, P., M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 33, No. 5, pp. 898–916, 2011.

23. Vicente, S., V. Kolmogorov and C. Rother, "Graph cut based image segmentation with connectivity priors", *IEEE Conference on Computer Vision and Pattern Recognition.*, pp. 1–8, 2008.

24. Pantofaru, C. and M. Hebert, "A comparison of image segmentation algorithms", *Robotics Institute*, p. 336, 2005.

25. Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs", *arXiv*

preprint arXiv:1412.7062, 2014.

26. Tai, L. and M. Liu, "Deep-learning in Mobile Robotics-from Perception to Control Systems: A Survey on Why and Why not", *arXiv preprint arXiv:1612.07139*, 2016.

27. Shapiro, L. and G. Stockman, "Computer Vision", *Prentice Hall*, 2002.

28. Pronobis, A. and B. Caputo, "COLD: COsy Localization Database", *The International Journal of Robotics Research (IJRR)*, Vol. 28, No. 5, pp. 588–594, May 2009.

29. Kurian, M., "Various Object Recognition Techniques for Computer Vision", *Journal of Analysis and Computation*, Vol. 7, No. 1, pp. 39–47, 2011.

30. Achatz, S., "State of the Art of Object Recognition Techniques", *Technical University of Munich*, 2016.

31. Lopez-Villa, J., H. Insuasti-Ceballos, S. Molina-Giraldo, A. Alvarez-Meza and G. Castellanos-Dominguez, "A novel tool for ground truth data generation for video-based object classification", *Signal Processing, Images and Computer Vision (STSIVA), 2015 20th Symposium on*, pp. 1–6, IEEE, 2015.

32. Mittal, S., M. S. Karthik, S. Kumar and K. M. Krishna, "Small Object Discovery and Recognition Using Actively Guided Robot.", *ICPR*, pp. 4334–4339, 2014.

33. Nüchter, A. and J. Hertzberg, "Towards semantic maps for mobile robots", *Robotics and Autonomous Systems*, Vol. 56, No. 11, pp. 915–926, 2008.

34. Anand, A., H. S. Koppula, T. Joachims and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds", *The International Journal of Robotics Research*, p. 0278364912461538, 2012.

35. Zender, H., O. Martínez Mozos, P. Jensfelt, G. J. M. Kruijff and W. Burgard, "Conceptual Spatial Representations for Indoor Mobile Robots", *Robot. Auton.*

Syst., Vol. 56, No. 6, pp. 493–502, 2008.

36. Wu, C., I. Lenz and A. Saxena, "Hierarchical semantic labeling for task-relevant rgb-d perception", *Robotics: Science and systems (RSS)*, pp. 1–9, 2014.

37. Galindo, C., A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics", *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2278–2283, IEEE, 2005.

38. Rogers, J. G. and H. I. Christensen, "Robot planning with a semantic map", *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2239–2244, IEEE, 2013.

39. Zhu, L., Y. Chen, Y. Lin, C. Lin and A. Yuille, "Recursive segmentation and recognition templates for image parsing", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 34, No. 2, pp. 359–371, 2012.

40. Newman, P., G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole and I. Reid, "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers", *The Int. J. of Rob. Res.*, Vol. 28, No. 11-12, pp. 1406–1433, 2009.

41. Özkan, K., Ş. Işık, Ö. N. Gerek and M. Doğan, "A new subspace based solution to background modelling and change detection", *International Conference on Advanced Technology and Sciences*, Vol. 1, p. 2, 2016.

42. Sun, M. and S. Savarese, "Model-Based Object Recognition", *Computer Vision: A Reference Guide*, pp. 488–492, Springer US, 2014.

43. Budden, D., S. Fenn, J. Walker and A. Mendes, "A novel approach to ball detection for humanoid robot soccer", *AI 2012: Advances in Artificial Intelligence*, pp. 827–838, Springer, 2012.

44. Khurana, K. and R. Awasthi, "Techniques for object recognition in images and multi-object detection", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol. 2, No. 4, pp. pp–1383, 2013.

45. Hu, W., A. M. Gharuib and A. E.-D. S. Hafez, "Template match object detection for inertial navigation systems", *Positioning*, Vol. 2, No. 02, p. 78, 2011.

46. Ahuja, N. and S. Todorovic, "From Region Based Image Representation to Object Discovery and Recognition", E. R. Hancock, R. C. Wilson, T. Windeatt, I. Ulusoy and F. Escolano (Editors), *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR&SPR 2010, Cesme, Izmir, Turkey, August 18-20, 2010. Proceedings*, pp. 1–19, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

47. Tsai, C.-F., Y.-H. Hu and W.-C. Lin, "The effect of region segmentation on object categorization", *Signal Processing, Communications and Computing (ICSPCC), 2016 IEEE International Conference on*, pp. 1–4, IEEE, 2016.

48. Gould, S., T. Gao and D. Koller, "Region-based segmentation and object detection", *Advances in neural information processing systems*, pp. 655–663, 2009.

49. Ren, X. and D. Ramanan, "Histograms of sparse codes for object detection", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3246–3253, 2013.

50. Panchal, P., S. Panchal and S. Shah, "A comparison of SIFT and SURF", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 1, No. 2, pp. 323–327, 2013.

51. Satpathy, A., X. Jiang and H. L. Eng, "LBP-Based Edge-Texture Features for Object Recognition", *IEEE Transactions on Image Processing*, Vol. 23, No. 5, pp. 1953–1964, 2014.

52. Juan, L. and O. Gwun, "A comparison of sift, pca-sift and surf", *International Journal of Image Processing (IJIP)*, Vol. 3, No. 4, pp. 143–152, 2009.

53. LeCun, Y., Y. Bengio and G. Hinton, "Deep learning", *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

54. Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", *arXiv preprint arXiv:1408.5093*, 2014.

55. Karaoğuz, H., Ö. Erkent and H. I. Bozma, "RGB-D based place representation in topological maps", *Machine Vision and Applications*, Vol. 25, No. 8, pp. 1913–1927, 2014.

56. Chang, C.-C. and C.-J. Lin, "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, pp. 27:1–27:27, 2011, software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`, accessed at December 2017.

57. Wu, T.-F., C.-J. Lin and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling", *Journal of Machine Learning Research*, Vol. 5, No. Aug, pp. 975–1005, 2004.

58. Lin, C.-J., R. C. Weng *et al.*, "Simple probabilistic predictions for support vector regression", *National Taiwan University, Taipei*, 2004.

59. Platt, J. *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", *Advances in large margin classifiers*, Vol. 10, No. 3, pp. 61–74, 1999.

60. Lin, H.-T., C.-J. Lin and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines", *Machine learning*, Vol. 68, No. 3, pp. 267–276, 2007.

61. Jain, L. P., W. J. Scheirer and T. E. Boult, "Multi-class open set recognition using probability of inclusion", *European Conference on Computer Vision*, pp. 393–409, Springer, 2014.

APPENDIX A: COLOR QUANTIZATION

Figure A.1 is given in order to visualize 25 color quanta. These colors are defined manually. All color quantization process done according to visual appearance of colors. Table A.1 established according to color appearance as well. Full coverage of HSV color space is accomplished and 172 color regions extracted. Those regions are given in Table A.1.



Figure A.1. Colors

Table A.1. HSV ranges

HueLow	HueHigh	SaturationLow	SaturationHigh	ValueLow	ValueHigh	Color
176	180	205	255	103	255	1
0	5	205	255	103	255	1
176	180	166	204	78	179	1
0	5	166	204	78	179	1
6	15	153	255	153	255	2
6	15	180	255	103	153	2
16	25	129	255	153	255	3
16	20	129	255	103	152	3
26	30	103	179	205	255	3
31	37	78	128	205	255	3
26	30	180	255	153	255	4
21	25	129	255	103	152	5
26	30	180	255	78	152	5
26	30	51	102	78	179	5
26	30	103	179	51	204	5
31	85	129	255	51	255	5
38	85	78	128	205	255	5
31	85	51	128	51	204	5
86	95	128	255	78	179	5
86	95	78	128	78	204	5
26	30	180	255	27	77	6
26	30	103	179	27	50	6
31	85	51	255	27	50	6
86	95	78	255	27	77	6
86	95	129	255	180	255	7
86	95	51	77	129	179	8
86	95	78	128	205	255	8
96	110	153	255	103	255	8

Table A.1. HSV ranges (cont.)

HueLow	HueHigh	SaturationLow	SaturationHigh	ValueLow	ValueHigh	Color
96	110	103	152	78	128	8
111	130	78	204	153	255	8
111	130	78	128	78	152	8
111	130	103	128	51	77	8
111	117	129	204	78	152	8
118	130	129	179	78	152	8
96	110	51	102	51	102	8
101	110	153	255	39	102	9
96	110	103	152	39	77	9
111	130	205	255	39	255	9
111	118	103	128	27	50	9
111	117	129	204	27	77	9
118	125	129	204	51	77	9
131	137	51	102	129	204	10
131	137	103	255	153	255	10
131	137	103	204	78	152	10
138	145	180	255	153	255	10
138	145	103	179	103	179	10
138	145	51	102	129	230	10
146	155	103	152	129	179	10
146	155	78	102	78	152	10
119	130	103	128	27	50	11
118	130	180	204	78	152	11
126	130	129	204	51	77	11
131	137	205	255	39	152	11
131	137	103	204	39	77	11
138	145	180	255	39	152	11
138	145	103	179	39	102	11
146	155	205	255	39	77	11
146	155	153	204	39	102	11

Table A.1. HSV ranges (cont.)

HueLow	HueHigh	SaturationLow	SaturationHigh	ValueLow	ValueHigh	Color
146	155	103	152	39	128	11
146	155	78	102	39	77	11
156	175	78	128	39	102	11
146	155	205	255	78	152	12
146	155	153	204	103	179	12
156	175	78	128	103	152	12
156	175	129	255	78	102	12
176	180	78	165	103	255	13
0	5	78	165	103	255	13
6	15	103	152	129	255	13
6	15	153	179	103	153	13
176	180	166	204	180	255	14
0	5	166	204	180	255	14
156	175	129	255	51	77	14
176	180	64	77	78	179	15
138	145	103	179	180	255	15
146	155	103	152	180	255	15
146	155	78	102	153	255	15
156	175	78	128	153	255	15
146	155	205	255	153	255	16
146	155	153	204	180	255	17
156	175	129	255	103	255	17
176	180	0	77	27	77	18
0	5	0	77	27	77	18
6	15	0	50	27	77	18
16	25	0	77	27	77	18
16	25	78	102	78	152	18
26	30	0	102	27	77	18
31	85	0	50	27	77	18
86	95	0	50	39	77	18

Table A.1. HSV ranges (cont.)

HueLow	HueHigh	SaturationLow	SaturationHigh	ValueLow	ValueHigh	Color
86	95	50	77	27	50	18
96	100	153	255	39	102	18
96	110	0	50	27	102	18
96	110	51	102	27	50	18
96	110	103	152	27	38	18
111	130	0	50	39	102	18
111	130	51	77	51	128	18
111	130	78	102	27	77	18
131	137	0	50	51	128	18
131	137	51	102	39	128	18
138	145	0	50	39	77	18
138	145	51	102	39	102	18
146	155	0	77	39	102	18
156	175	0	50	39	102	18
156	175	51	77	39	128	18
176	180	0	63	78	179	19
0	5	0	63	78	179	19
0	5	64	77	78	179	19
6	15	0	102	78	128	19
16	25	0	50	78	128	19
16	25	51	77	78	152	19
26	30	0	50	78	102	19
31	85	0	50	78	152	19
86	95	0	50	78	152	19
86	95	50	77	51	128	19
96	110	0	50	103	179	19
111	130	0	50	103	179	19
111	130	51	77	129	230	19
131	137	0	50	129	179	19
138	145	0	50	78	152	19

Table A.1. HSV ranges (cont.)

HueLow	HueHigh	SaturationLow	SaturationHigh	ValueLow	ValueHigh	Color
138	145	51	102	103	128	19
146	155	51	77	103	204	19
146	175	0	50	103	152	19
156	175	51	77	129	204	19
176	180	78	165	27	102	20
0	5	78	165	27	102	20
176	180	205	255	27	102	21
0	5	205	255	27	102	21
176	180	166	204	27	77	21
0	5	166	204	27	77	21
6	15	153	255	27	102	21
6	15	51	102	27	77	21
6	15	103	152	27	128	21
16	25	129	255	27	102	21
156	175	129	255	39	50	21
16	25	78	128	27	38	22
86	95	0	50	27	38	22
96	110	153	255	27	38	22
111	130	205	255	27	38	22
111	130	0	50	27	38	22
111	130	51	77	27	50	22
118	130	129	204	27	50	22
131	175	0	255	27	38	22
16	25	78	128	39	77	22
0	30	26	50	180	255	24
6	30	0	50	129	179	24
96	130	0	50	180	204	24
146	175	51	77	205	255	24
138	175	0	50	153	204	24
176	180	39	77	180	255	24

Table A.1. HSV ranges (cont.)

HueLow	HueHigh	SaturationLow	SaturationHigh	ValueLow	ValueHigh	Color
0	5	39	77	180	255	24
16	25	51	77	153	255	24
16	25	78	128	180	255	24
26	30	0	50	103	128	24
26	30	51	102	180	255	24
31	85	0	50	153	179	24
31	85	51	77	205	255	24
86	95	0	50	153	204	24
86	95	51	77	180	255	24
96	110	51	102	205	255	24
111	130	51	77	231	255	24
131	137	51	102	205	255	24
138	145	51	102	231	255	24
6	15	51	102	129	255	24
16	25	78	102	153	179	24
16	25	103	128	78	179	24
176	180	0	50	180	255	25
0	30	0	25	180	255	25
31	85	0	50	180	255	25
86	130	0	50	205	255	25
131	137	0	50	180	255	25
138	175	0	50	205	255	25