Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Weakly-Supervised Learning of Visual Object Models

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Sami Alper Ince BSc
Matrikelnummer 0928288

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Univ.Prof.Dipl.-Ing.Dr. Christian Breiteneder
Mitwirkung: Mag.Dipl.-Ing.Dr. Matthias Zeppelzauer

Wien, 28.11.2017

Sami Alper Ince

Christian Breiteneder

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Weakly-Supervised Learning of Visual Object Models

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Sami Alper Ince BSc
Matrikelnummer 0928288

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ.Prof.Dipl.-Ing.Dr. Christian Breiteneder
Assistance: Mag.Dipl.-Ing.Dr. Matthias Zeppelzauer

Vienna, 28.11.2017

Sami Alper Ince

Christian Breiteneder

ERKLÄRUNG ZUR VERFASSUNG DER ARBEIT

Sami Alper Ince BSc
Rainergasse 35/1/8, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 28.11.2017

Sami Alper Ince

ACKNOWLEDGEMENTS

First of all, I would like to thank Matthias Zeppelzauer for his outstanding supervision of this master’s thesis. It is a great honor for me to have such an opportunity of getting supervision from Matthias Zeppelzauer. This great opportunity has already resulted many advantages in my on-going professional career and life thanks to the exceptional knowledge and experience regarding Computer Science shared by Matthias Zeppelzauer. Secondly, I thank all the instructors who have a part in my studies at Vienna University of Technology. Thirdly, I thank my parents and sister for their mental support since the beginning of my studies. Finally, I thank my wife Joanna for making me be more creative and ambitious during my career.

KURZFASSUNG

Diese Masterarbeit untersucht die schwach beaufsichtigte visuelle Objektdetektion eines gegebenen Satzes von Bildern. Hauptziel ist es, ein optimales Objektmodell für jedes ausgewählte visuelle Objekt zu erhalten, indem dieses sowohl positiv als auch negativ auf Bildniveau markierte Bilder lernt. Zu diesem Zweck wird ein Analyseprozess vorgeschlagen, der Segmente des Zielvisualobjekts aus positiven Trainingsbildern sammelt. Ein allgemeineres Objektmodell wird durch die Bestimmung der am meisten diskriminierenden detektierten Objektsegmente aufgebaut. Diese endgültige Form des Objektmodells wird von einem binären Klassifikator verwendet, um Segmente des Zielvisualobjekts aus Testbildern zu detektieren.

Der vorgeschlagene Ansatz zur Wiederherstellung eines optimalen Objektmodells umfasst vier Hauptverarbeitungsschritte: Segmentierung, Merkmalsextraktion, Ähnlichkeitsmessung und Lernen. Für jeden dieser Schritte wird die Eignung verschiedener Techniken ausgewertet. Als erstes wird eine Auswertung in Bezug auf die Segmentierung mit einer mittleren Schichtsegmentierung und einem einfacheren und schnelleren Gleitfensteransatz durchgeführt. Zweitens werden für die Beschreibung der im ersten Schritt erhaltenen Segmente unterschiedliche Arten von Merkmalen (Farbhistogramme, dichte SIFT-Deskriptoren, PHOW-Deskriptoren, VLAD-Deskriptoren, CEDD und MPEG-7-Farbdeskriptoren) ausgewertet. Der dritte Schritt beinhaltet bei der Ähnlichkeitsmessung eine Auswertung unterschiedlicher Distanz- und Ähnlichkeitsfunktionen. Schließlich wird im Lernschritt ein nicht parametrisches, diskriminierendes Lernschema verwendet, welches auf Informationsgewinn basiert. Das Ergebnis des Lernens ist ein Ranking, das die Unterscheidungskraft jedes Kandidatensegments ausdrückt. Darüber hinaus wird die MPEG-Videocodierung als alternative Technik für die Merkmalsextraktion und die Ähnlichkeitsmessung untersucht. Zu diesem Zweck wird ein Ansatz, der aus der Texturklassifizierung stammt, auf Farbbildsegmente erweitert.

Die experimentellen Ergebnisse zeigen eine geringere rechnerische Komplexität für alle Kombinationen von untersuchten Merkmalsdeskriptoren und Distanzfunktionen im Vergleich zum MPEG-Video-Codierungs-basierten Ansatz. Weiterhin ist die Genauigkeit von visuellen Objektmodellen, die durch MPEG-Videocodierung erhalten werden, niedriger als diejenigen, welche in einigen der vorgeschlagenen Ansätze verwendet werden, die getrennte Verarbeitungsschritte für die Merkmalsextraktion und die Ähnlichkeitsmessung verwenden. Diese Masterarbeitsergebnisse zeigen, dass die Repräsentation in Form von VLAD-Deskriptoren zusammen mit einer der drei Distanz-Funktionen (Chi-Quadrat-Statistik, Diffusions-Distanz und Euklidischer Abstand) die besten Ergebnisse liefern. Darüber hinaus wird eine Auswertung von Zielobjektdetektoren durch Auswählen eines der wiederhergestellten Objektmodelle für jede Auswertung durchgeführt. Der Zielobjektdetektor ist ein SVM-Klassifikator, ein Naïve Bayes-Klassifikator und ein alternativer Ansatz, der Informationsgewinn verwendet, um Entscheidungsschwellen zu lernen.

ABSTRACT

This master's thesis investigates the weakly-supervised visual object detection from a given set of images. The main goal is set as obtaining an optimal object model for any selected visual object by learning from positive- and negative-labeled images. To this end, an analysis process is proposed that gathers segments of the target visual object from positive training images. A more common object model is built by determining the most discriminative detected object segments. This ultimate form of the object model is employed by a binary classifier in order to detect segments of a target visual object from test images.

The proposed approach for the recovery of an optimal object model comprises of four major processing steps: segmentation, feature extraction, similarity measurement, and learning. For each of these steps, the suitability of different techniques is evaluated. Firstly, an evaluation with respect to segmentation is made with mean shift segmentation and a simpler and faster sliding window approach. Secondly, different types of features (color histograms, dense SIFT descriptors, PHOW descriptors, VLAD descriptors, CEDD and MPEG-7 color descriptors) are evaluated for the description of the segments obtained in the first step. Thirdly, in similarity measurement an evaluation involves different distance and similarity functions. Lastly, in the learning step a non-parametric discriminative learning scheme based on information gain is employed. The result of learning is a ranking that expresses the distinctiveness of each candidate segment. In addition, MPEG video encoding is investigated as an alternative technique for both feature extraction and similarity measurement. For this purpose, an approach originating from texture classification is extended to color image segments.

The experimental results demonstrate lower computational complexity for all combinations of investigated feature descriptors and distance functions compared to the MPEG video encoding-based approach. Furthermore, the accuracy of visual object models obtained by MPEG video encoding is lower than those presented in some of the proposed approaches employing separate processing steps for feature extraction and similarity measurement. These master thesis results suggest using the feature descriptors obtained from VLAD descriptors and one of three distance functions: chi square statistics, diffusion distance and Euclidean distance. Moreover, an evaluation of target object detectors is performed by selecting one of the recovered object models for each evaluation. The target object detector is an SVM classifier, a Naïve Bayes classifier, and an alternative approach employing information gain to learn decision thresholds.

LIST OF FIGURES

Figure 1.1: Demonstration of the idea of weakly-supervised visual object detection from images............ 3
Figure 2.1: The visualization of some examples of different “generic” descriptors by assigning them to the respective pixel locations in the case of obtaining the highest probability for the respective “generic” descriptors [21]. 8
Figure 2.2: The visualization of the pixel locations which produce the highest probabilistic scores for the given test image [21]. 10
Figure 2.3: A spectrogram mapping an insect sound for a specific time frame [23]. 12
Figure 2.4: The presentation of the samples created by different techniques in order to bring CK-1 distance function in the property of rotation invariance [28]. 17
Figure 2.5: The illustration of maximizing the information gain for some samples of the lined up positive and negative fragments of the spectrograms on a number line based on their CK-1 distances to a sample of candidate fragment [23]. 19
Figure 2.6: The number line representations of CK-1 distances between the sample fragments and two different selected fragments having exactly the same information gain [23]. 19
Figure 2.7: The visualization of the increase in Mean AP as a consequence of the dimensionality reduction while PCA is applied into the local SIFT Descriptors [44]. 22
Figure 2.8: The visualization of the Segmentation as Selective Search proposed by Uijlings et al. [45]. 22
Figure 3.1: The workflow collecting a set of the parts belonging to the target visual object from a set of positive and negative image-level labeled color images. 26
Figure 3.2: A sample image which is given to the processing step of segmentation as an input. 28
Figure 3.3: The patches collected by the sliding windows. While each vertical step size is equal to the height of each patch, the size of each horizontal step is equal to the width of each patch. 28
Figure 3.4: The patches collected by the sliding windows. While the size of each vertical step has a two times smaller height of each patch, each horizontal step size has a two times width of each patch. 29
Figure 3.5: An instance from a set of positive color images. Since the elephant is selected as the target visual object, the image is positive. 30
Figure 3.6: Some output samples of the segmentation unit with the use of both the mean shift segmentation method and the alignment of rectangular boundaries are illustrated. 30
Figure 3.7: Two sample color images whose color histograms are recognized as similar based on the standards of similarity function defined by Pass et al. [55]. 32
Figure 3.8: The visual representation of CIELAB color space, which is modelled with opponent coloring in 3-dimensions [58]. 33
Figure 3.9: Four different segments obtained in the color space of CIE L*a*b by the application of mean shift segmentation. 33
Figure 3.10: The exhibition of AB color histograms obtained from four different segments shown in Figure 3.9. 34

Figure 3.11: Four different segments obtained in the color space of CIE L*u*v by the application of mean shift segmentation. 35
Figure 3.12: The exhibition of UV color histograms obtained from four different segments shown in Figure 3.11. 36
Figure 3.13: The AB and UV color histograms extracted from two different samples of the segments are depicted. 37
Figure 3.14: An instance of a cube representing the manifold of colors indicated by RGB color model [61]. 39
Figure 3.15: A sample image segment obtained by mean shift segmentation and extracted with rectangular boundary by setting background region with black color in order to be used for the extraction of MPEG-7 color structure descriptor. 41
Figure 3.16: Histograms constructed from beginning to the end of extraction process of MPEG-7 color structure descriptor from the image segment shown in Figure 3.15 in addition to a performed nonlinear quantization function. 41
Figure 3.17: Histograms constructed from beginning to the end of extraction process of MPEG-7 scalable color descriptor from the image segments shown in Figure 3.18 in addition to a performed nonlinear scaling function. 42
Figure 3.18: A sample image segment obtained by mean shift segmentation and extracted with rectangular boundary by setting background region with black color in order to be used for the extraction of MPEG-7 scalable color descriptor. 43
Figure 3.19: The illustration of the updated design of Figure 3.1 with the additional usage of 10 positive images selected from the initial positive image set in order to construct a visual vocabulary in the processing step of feature extraction. 44
Figure 3.20: The depiction of dense SIFT computation over one of the selected positive training images with different sampling step sizes (a single pixel (a) or 2 pixels (b) or 5 pixels (c)). 46
Figure 3.21: The enlarged representation of the gathering dense SIFT descriptors from a segment sample. 47
Figure 3.22: Demonstration of precision calculations for the recovered object models in order to select optimal ones. 51
Figure 3.23: The visualization of the threshold spaces created by different fingerprint samples shown in Figure 3.24 in order to detect a test sample as positive or negative. 53
Figure 3.24: 4 different fingerprint samples of an optimal object model. 53
Figure 4.1: The presentation of two positive sample images from the elephant dataset and their ground truth data. 55
Figure 4.2: Two samples from the positive images of the tiger dataset with their ground truth data. 56
Figure 4.3: Demonstration of the dependencies between the processing steps and their degrees of freedom to setup an experiment with respect to an object model recovery. 58
Figure 4.4: Demonstration of the dependencies between the processing steps and their degrees of freedom to setup an experiment with respect to the detection of target visual objects. 64
Figure 5.1: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each single pixel from the mean shift segments. 72

Figure 5.2: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 74
Figure 5.3: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 5 pixels from the mean shift segments. 75
Figure 5.4: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 5 pixels from the sliding windows. 76
Figure 5.5: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the sliding windows. 77
Figure 5.6: The recovery processes of 4 different object models based on the first sample training image are visualized. 79
Figure 5.7: The recovery processes of 4 different object models based on the second sample training image are visualized. 81
Figure 5.8: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the violet curve presented in Figure 5.2. 82
Figure 5.9: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.8. 82
Figure 5.10: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the yellow curve presented in Figure 5.2. 83
Figure 5.11: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.10. 84
Figure 5.12: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the violet curve presented in Figure 5.3. 84
Figure 5.13: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.12. 85
Figure 5.14: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model

whose precision score places on the node defining 50 fingerprints of the green curve presented in Figure 5.3. 85
Figure 5.15: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.14. 86
Figure 5.16: The representation of the precision scores received from the object models recovered by using GB color histograms and performing mean shift segmentation in CIELAB color space. 87
Figure 5.17: The representation of the precision scores received from the object models recovered by using GB color histograms and performing mean shift segmentation in CIELUV color space. 87
Figure 5.18: The representation of the precision scores received from the object models recovered by using UV color histograms. 88
Figure 5.19: The recovery process of an object model based on two different sample training images by using UV color histograms and diffusion distance is visualized. 89
Figure 5.20: The representation of the precision scores received from the object models recovered by utilizing 1d RGB color histograms whose construction is based on the concatenation of each 2d histogram of RG, GB and BR channels into a single dimensional vector. 90
Figure 5.21: The representation of the precision scores received from the object models recovered by using AB color histograms. 91
Figure 5.22: The representation of the precision scores received from the object models built by using CK-1 distance measure. 92
Figure 5.23: The representation of the precision scores received from the object models built by using either MPEG-7 color structure descriptors or MPEG-7 scalable color descriptors. . 93
Figure 5.24: The precision scores of the object models built by using CEDDs are represented. 94
Figure 5.25: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 95
Figure 5.26: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 97
Figure 5.27: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using chi square statistics and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each pixel from the mean shift segments. 98
Figure 5.28: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and

250 best fingerprints) recovered by using chi square statistics and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each pixel from the mean shift segments. 99
Figure 5.29: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using l_2 distance and the GB color histograms extracted from the mean shift segments obtained in CIELUV color space. 101
Figure 5.30: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using l_2 distance and the GB color histograms extracted from the mean shift segments obtained in CIELUV color space. 102
Figure 5.31: The representation of both ROC and PR curves received from the linear SVM classifier initially trained with the object model (keeping the 25 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments and the ones received from re-trained of the same classifier with the hard examples. 104
Figure 5.32: The representation of both ROC and PR curves received from the linear SVM classifiers initially trained with the object models from the set (keeping the 50, 100 and 150 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 105
Figure 5.33: The representation of both ROC and PR curves received from the linear SVM classifiers initially trained with the object models from the set (keeping the 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 106
Figure 5.34: The representation of both ROC and PR curves received from the linear SVM classifiers re-trained with the hard examples after their initial training with the object models from the set (keeping the 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 107
Figure 5.35: The representation of both ROC and PR curves received from the linear SVM classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using l_2 distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 108
Figure 5.36: The representation of both ROC and PR curves received from the linear SVM classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using l_2 distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 109
Figure 5.37: The representation of both ROC and PR curves received from the classifiers constructed based on the thresholds of the fingerprints belonging to the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 111

Figure 5.38: The representation of both ROC and PR curves received from the classifiers constructed based on the thresholds of the fingerprints belonging to the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 112
Figure 5.39: The representation of the precision scores collected based on the tiger dataset from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 114
Figure 5.40: The representation of both ROC and PR curves collected based on the tiger dataset from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 115
Figure 5.41: The representation of both ROC and PR curves collected based on the tiger dataset from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments. 116
Figure 5.42: The recovery process of the object model (storing the 25 best fingerprints and having 100% precision score shown in Figure 5.39) based on the first sample training image of the tiger dataset is visualized. 117
Figure 5.43: The recovery process of the object model (storing the 25 best fingerprints and having 100% precision score shown in Figure 5.39) based on the second sample training image of the tiger dataset is visualized. 118
Figure 5.44: The recovery process of the object model (storing the 25 best fingerprints and having 100% precision score shown in Figure 5.39) based on the third sample training image of the tiger dataset is visualized. 118

LIST OF TABLES

Table 2.1: The implementation of CK-1 distance function in Matlab which exploits the inter frame compression step of MPEG-1 format for the symmetrically obtained reference and predicted movie frames [28]. 16
Table 2.2: The performance comparison between Multiple Instance Learning (MIL) and Multi-fold Multiple Instance Learning based on the criteria such as “CorLoc” and “Detection mAP” by making use of foreground (F), background (B) and contrastive (C) features [40]. 24
Table 4.1: The representation of all the experimental setups with respect to the object model recovery containing VLAD extraction. 59
Table 4.2: The representation of all the experimental setups with respect to the object model recovery containing the extraction of one of color histograms. 60
Table 4.3: The representation of all the experimental setups with respect to the object model recovery containing the extraction of either dense SIFT or PHOW descriptors. 61
Table 4.4: The representation of the experimental setups for the employment of CK-1 distance function. 62
Table 4.5: The representation of all the experimental setups with respect to the object model recovery containing the extraction of one of MPEG-7 color descriptors. 63
Table 4.6: The representation of all the experimental setups with respect to the object model recovery containing the extraction of CEDD. 63
Table 4.7: The configuration details of each experimental setup utilizing only Naïve Bayes classifier in its processing step of classification are demonstrated. 65
Table 4.8: The configuration details of each experimental setup utilizing only linear SVM classifier in its processing step of classification are demonstrated. 66
Table 4.9: The configuration details of each experimental setup containing only the classifier constructed by using the thresholds of selected fingerprints in its processing step of classification are demonstrated. 67
Table 5.1: The highest mean precisions obtained from VLAD-based model recovery 86
Table 5.2: The highest mean precisions obtained from Color Histograms-based model recovery 92
Table 5.3: The highest mean accuracies obtained from Naïve Bayes-based Object Detection 103
Table 5.4: The highest mean accuracies obtained from SVM-based Object Detection 110
Table 5.5: The highest mean accuracies obtained from Thresholds-based Object Detection 113
Table 5.6: The mean precisions of the set of models built based on the same experimental setup and the tiger dataset of Figure 5.39 by using the same distance function for each set are presented. 114
Table 5.7: The accuracies received from the Naïve Bayes classifiers whose ROC and PR curves are demonstrated in Figure 5.40 and Figure 5.41 are presented with their mean accuracy. .. 117

Table of Contents

ERKLÄRUNG ZUR VERFASSUNG DER ARBEIT V

ACKNOWLEDGEMENTS VII

KURZFASSUNG IX

ABSTRACT XI

LIST OF FIGURES XIII

LIST OF TABLES XX

1 INTRODUCTION 1

1.1 MOTIVATION 2
1.2 PROBLEM STATEMENT 2
1.3 APPLICATIONS 3
1.4 OBJECTIVES 4
1.5 OUTLINES 5

2 RELATED WORK 6

2.1 PROBABILISTIC VISUAL OBJECT MODELING PROPOSED BY SCHMID [21] 6
2.1.1 DESCRIPTION OF METHOD 6
2.1.2 RELATION TO THIS THESIS 11
2.1.3 IDEAS TAKEN OR EXTENDED FOR THIS THESIS 11
2.2 FINGERPRINT TECHNOLOGY PROPOSED BY HAO ET AL. [23] 11
2.2.1 DESCRIPTION OF METHOD 11
2.2.2 RELATION TO THIS THESIS 20
2.2.3 IDEAS TAKEN OR EXTENDED FOR THIS THESIS 20
2.3 MULTI-FOLD WEAKLY-SUPERVISED LEARNING PROPOSED BY CINBIS ET AL. [40] 20
2.3.1 DESCRIPTION OF METHOD 21
2.3.2 RELATION TO THIS THESIS 25
2.3.3 IDEAS TAKEN OR EXTENDED FOR THIS THESIS 25

3 WEAKLY-SUPERVISED LEARNING OF VISUAL OBJECT MODELS 26

3.1 SEGMENTATION 27
3.1.1 SLIDING WINDOWS 27

3.1.2 MEAN SHIFT SEGMENTATION 29
3.2 FEATURE EXTRACTION 31
3.2.1 COLOR HISTOGRAMS 31
3.2.1.1 AB Color Histograms 32
3.2.1.2 UV Color Histograms 34
3.2.1.3 Comparison Between AB and UV Color Histograms 36
3.2.1.4 RGB Color Histograms 38
3.2.1.5 GB Color Histograms 39
3.2.2 MPEG-7 COLOR DESCRIPTORS 40
3.2.2.1 MPEG-7 Color Structure Descriptor 40
3.2.2.2 MPEG-7 Scalable Color Descriptor 42
3.2.3 COLOR AND EDGE DIRECTIVITY DESCRIPTOR (CEDD) 43
3.2.4 BAG OF VISUAL WORDS (BOVW) 43
3.2.4.1 Dense SIFT and PHOW Descriptors 45
3.2.4.2 Vector of Locally Aggregated Descriptors Encoding (VLAD) 48
3.3 MODEL BUILDING 48
3.3.1 SIMILARITY MEASUREMENT 49
3.3.2 APPLICATION OF CK-1 DISTANCE 49
3.3.3 APPLICATION OF INFORMATION GAIN 50
3.4 DETECTION OF TARGET VISUAL OBJECTS 51

4 EVALUATION 54

4.1 DATASETS 54
4.1.1 DEVELOPMENT DATASET 54
4.1.2 VERIFICATION OF GENERALIZABILITY 55
4.2 PARAMETER SELECTION 56
4.3 EXPERIMENTAL SETUP 57
4.3.1 EXPERIMENTAL SETUP OF VISUAL OBJECT MODEL RECOVERY 58
4.3.1.1 VLAD-based Model Recovery 59
4.3.1.2 Color Histograms-based Model Recovery 60
4.3.1.3 Dense SIFT and PHOW Descriptors-based Model Recovery 60
4.3.1.4 CK-1-based Model Recovery 61
4.3.1.5 MPEG-7 Color Descriptors-based Model Recovery 62
4.3.1.6 CEDD-based Model Recovery 63
4.3.2 EXPERIMENTAL SETUP OF TARGET VISUAL OBJECT DETECTION 64
4.3.2.1 Naïve Bayes-based Object Detection 65
4.3.2.2 SVM-based Object Detection 66
4.3.2.3 Thresholds-based Object Detection 67
4.4 EVALUATION MEASURE 67
4.5 RUN TIME AND STORAGE EVALUATION 68

5 RESULTS 71

5.1 RESULTS OF VISUAL OBJECT MODEL RECOVERY 71

5.1.1 VLAD-based Model Recovery 71
5.1.2 Color Histograms-based Model Recovery 86
5.1.3 CK-1-based Model Recovery 92
5.1.4 MPEG-7 Color Descriptors-based Model Recovery 93
5.1.5 CEDD-based Model Recovery 94
5.2 Results of Target Visual Object Detection 94
5.2.1 Naïve Bayes-based Object Detection 95
5.2.2 SVM-based Object Detection 103
5.2.3 Thresholds-based Object Detection 110
5.3 Results Obtained from Tiger Dataset 113
5.4 Conclusion of The Experiments 119

6 SUMMARY 121

6.1 Future Work 123

APPENDIX A 124

A.1 Formal Proof of A Problem Reduction 124
A.2 Results of RGB Color Histograms-based Model Recovery 125
A.3 Results of Dense Sift and PHOW Descriptors-based Model Recovery 126
A.4 Supplementary Results of Thresholds-based Object Detection 131

REFERENCES 134

CHAPTER 1

1 INTRODUCTION

The recognition of visual objects by automated software systems is used in different industries; e.g. automotive, medical, sport, military, security, governmental and remote sensing [1] [2] [3] [4] [5] [6] [7]. Depending on the labeling structure and ratio of input data utilized during the learning process of the automated systems, different machine learning techniques such as supervised, weakly-supervised, semi-supervised and unsupervised [8] are employed in order to provide object recognition capabilities to the automated systems. Supervised learning is defined as the training of a classifier by using a labeled training dataset in which the category information is supplied for each data by a label; whereas the unsupervised learning focuses on clustering an unlabeled training dataset intrinsically [9]. By taking these both learning strategies into consideration, the definitions of semi-supervised and weakly-supervised learning lie between supervised and unsupervised learning [8] [10]. In semi-supervised learning, category labels are provided only for a small amount of data placing in a training dataset. The rest of the data does not contain any label; therefore, it is ambiguous [10]. If a training dataset is gathered from visual space such as a set of image segments and a binary classification task is specified to group the segments as positive or negative, then only a small number of labeled segments from this complete set is sufficient for semi-supervised learning. On the other hand, the only required information for weakly-supervised learning is that in which images exists at least one positive segment. However, which image segment is positive or negative is unknown. When the labels are attached to only images, they are called image-level labels. Apart from that, if the labels are attached to segments, they are called pixel-level labels.

Gathering pixel-level labeled images for supervised or semi-supervised learning requires a per-pixel annotation process, which is performed either manually or automatically. For example, doctors manually label signs of diseases in medical images to utilize them as a set of pixel-level labeled medical images for supervised or semi-supervised learning [2]. Likewise, oracles from all other industries utilizing pixel-level labeled images need to manually perform a per-pixel annotation process if there is no automated solution for this. Since the image-level labeling does not contain a per-pixel annotation process, it requires less labeling effort than pixel-level labeling. Apart from that, millions of online images are accessible free of cost with their image-level labels in the image hosting websites such as Flickr [11]. Therefore, assigning the oracles to a pixel-level labeling task is a labor-intensive and time-consuming process [2].

Apart from the employing of a machine learning technique, the visual object recognition also requires two more processing steps in its entire pipeline such as segmentation and feature extraction [12]. Segmentation is performed on a set of input images as an initial processing step of visual object recognition in order to isolate visual objects from the background. Furthermore, feature extraction follows the segmentation in the pipeline and derives the feature values from the image segments by eliminating their redundant data and preserving their significant and discriminative information. Therefore, the feature values which are derived by the feature extraction facilitate the interpretation of segments. The processing step of machine learning follows the feature extraction in the pipeline of visual object recognition for this interpretation.

1.1 MOTIVATION

Since the supervised learning methods are operated with a set of completely labeled samples, the statistical models constructed by the supervised learning methods are stronger than the ones constructed with only a set of completely unlabeled samples by the unsupervised learning methods [9]. Nevertheless, unsupervised learning methods can be used to either improve the models initially constructed by supervised learning methods with a small number of labeled samples, or extract features from unlabeled samples [9]. In fact, the former case gets involved in the scope of semi-supervised learning since the input data is formed with the mixture of both labeled and unlabeled samples [10]. Furthermore, it is a superior idea to construct a model with a weakly-supervised learning method instead of a supervised or semi-supervised learning method in order to avoid pixel-level labeling effort. From this point of view, the weakly-supervised visual object detection from images is investigated in this thesis.

The overall aim of this thesis is to recover an optimal object model by learning from positive and negative images. For this purpose, the most discriminative parts of the target visual object are mined from positive and negative images. From the detected object parts, a more general object model is built. A classifier is constructed that uses the object model to detect the target visual object in test images.

1.2 PROBLEM STATEMENT

The topic of this thesis is weakly-supervised visual object detection from images. For this purpose, a framework which implements different optional components for each core image processing step of the entire pipeline is implemented to evaluate the results with respect to execution timing, accuracy, required hardware sources, and stability. The implemented technique of the thesis is applicable to any selected species of animals, plants, or inanimate objects as a target visual object. In order to produce experimental results in this thesis, the species of elephant is selected as the target visual object. Color images containing image-level labels are given to the developed algorithm not only as a training dataset, but also as a test dataset. The label of each image is assigned as positive if the image includes the selected target visual object once or multiple times; otherwise it is assigned as negative. One half of the training images includes positive images while the other half includes negative images. During training, object models are recovered based on image features and image segmentation and the precision scores of object models are computed by validating against the pixel-level ground truth of the positive training images.

During testing, one of the implemented binary classifiers can be selected to detect image segments containing target visual objects from a given test dataset by using a previously built object model. The test dataset contains only positive images together with pixel-level ground truth. The image segments classified by a selected binary classifier are used to compute accuracy, ROC curve and precision-recall curve by checking with the ground truth of test set.

Consequently, the performance of each binary classifier is evaluated based on both curves and results. Figure 1.1 demonstrates the explanation of a whole set of tasks described in this section.

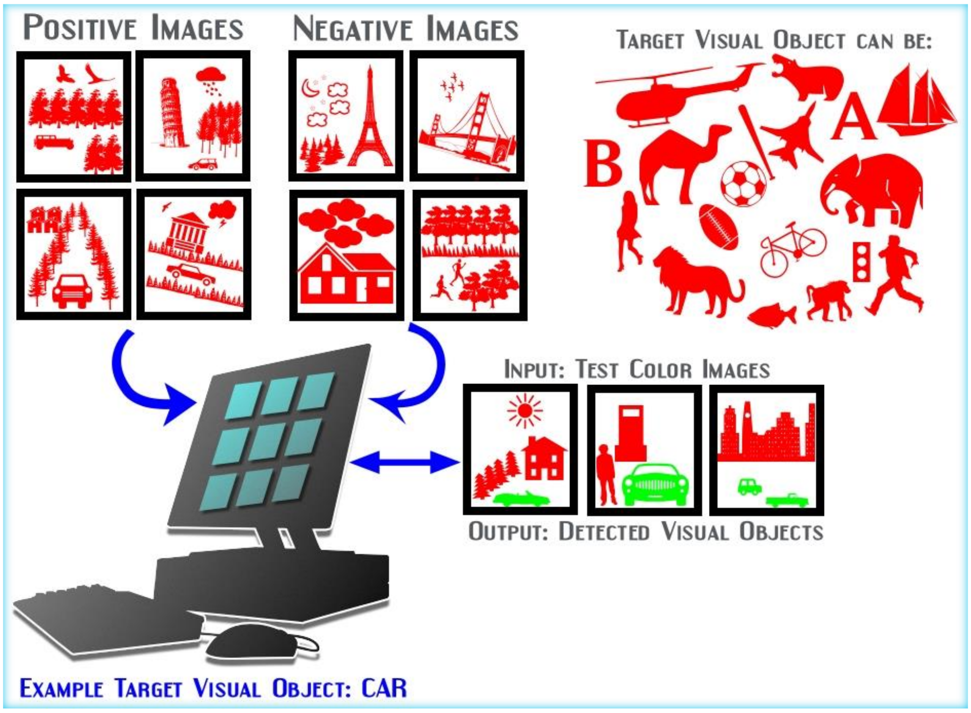

Positive Images
Negative Images
Target Visual Object can be:
A
B
Input: Test Color Images
Output: Detected Visual Objects
Example Target Visual Object: CAR

Figure 1.1: Demonstration of the idea of weakly-supervised visual object detection from images.

1.3 APPLICATIONS

Since fully automated visual object recognition applications achieve given related tasks faster, more cheaply and in a more standardized manner than the ones manually supplied by the oracles [13], the weakly-supervised visual object detection framework introduced in this thesis can enable this technological opportunity to be utilized in the different demanding areas of the industry by distributing its related applications. For instance, an application recognizing visual objects can be implemented by using our approach in order to be established as an integrated vision system of a robotic system, which is used as a production facility with respect to the motion control and regulation aids [14]. More specifically, thanks to such a visual object recognition application, a robot can learn visual objects shown by a human operator. Hence, this improves the Human-Robot Interaction (HRI). Furthermore, Fanello et al. [15] present a work regarding weakly-supervised visual object recognition in the same robotics scenarios; therefore, the approach of this thesis can also be used to implement an application for this work.

Automated image recognition techniques are not only employed as computer vision systems integrated with robotic systems, but also used in different industries without robots. For instance, an application tracking and identifying players from broadcast sport videos can be implemented by using the visual object detection approach of this thesis. Moreover, Zhang et al. [3] present another type of weakly-supervised learning technique in order to achieve the same goals by using the videos of basketball plays. Although the videos used in the scope of the experiments presented in [3] belongs to basketball plays, the approach of is proposed for whole sport industry due to the inferring capability of weakly-supervised learning method for a given target visual object. Moreover, it is also possible to utilize our weakly-supervised approach for the delivery of the applications detecting target visual objects in the remote sensing industry. For instance, the weakly-supervised learning approach for airplane detection in remote sensing images introduced by Han et al. [7] can be replaced with our approach by preserving the same goals of the framework.

Apart from the applications associated with the commercial targets, a non-commercial application supplying universally professional wildlife services can be implemented based on the theoretical approach proposed in this thesis in order to protect animals and increase the universal quality level of animal ethics. For instance, elephants are selected as target visual objects for sets of collected images in order to track the research results of this thesis, although it is applicable to any target visual object, which denotes a group of either the inanimate entities or the living beings as described in Section 1.2. Due to illegal trading of ivory, the existence of the intensive poaching of African elephants is established [16]. African elephants have more advantages than Asian elephants with respect to both mating and collecting food due to their larger tusks [17]. Since ivory poachers are inclined to hunt the elephants with larger tusks and the Asian elephants are seen as the valuable sources of livestock by the inhabitants, the African elephants owning the largest tusks become the main target for the ivory poachers [16]. As a result of this, the new generation of African elephants are born with the smaller tusks due to the immediate change in their genes, in other words, due to the mutation [17]. Since the introduced approach in this thesis is applicable to any selected target visual object, it can be globally used as a supportive and technological wildlife management tool by the law enforcement agencies to protect the animals having similar problems.

The common behavior of all of these aforementioned approaches is not only being fully automated, but also having an inferring capability of recognizing a given target visual object without any dependency of object specification and pixel-level labeling. This inferring capability is obtained due to the natural consequence of weakly-supervised techniques. Such learning methods are useful when no pixel-level ground truth labels exist and applicable to a broad range of visual objects.

1.4 OBJECTIVES

This thesis has two main objectives. The first objective of this thesis is to extract compact and expressive feature descriptors for visual content. The compactness of feature descriptors has two particular advantages. The first one is a reduced storage cost of feature descriptors due to their reduced size. The second advantage of compactness regarding feature descriptors is their shortened computation and retrieval time frames. Furthermore, the expressiveness of feature descriptors is an advantage for the determination of differentiation or similarity among themselves.

There are companies, which develop or support business intelligence applications for one of the many different areas, for example, finance, sales, or multimedia [18]. In this case, these companies utilize data warehouses or data marts in addition to databases containing only raw data related to the areas of their industries. The data warehouses are formed by merging and selecting the repositories of heterogeneous raw data to gather correlated data in a central data station and assist the progress of the data mining [19]. The multimedia data warehouses comprise the integrated, elaborated and correlated data obtained from the multiple heterogeneous databases containing videos, audio recordings, images and textual information [19]. The data marts are the subsets of the data warehouses and contain the specific scopes of the integrated repositories of the data [20].

Based on the assumption that a commercial multimedia application holds all the feature descriptors extracted from a set of images in a multimedia data warehouse and this set of images in other separated databases, the compactness of the feature descriptors enables this multimedia data warehouse to store a decreased amount of required physical data. If the data placing in the data warehouse of a production multimedia application running for a company is purged, then it also causes a data purging for all the development and quality applications used for the business purposes of the same company in the multimedia industry. Therefore, the compactness of feature descriptors reduces the costs of companies to develop, maintain or support multimedia applications. When the required data to be stored in a data warehouse becomes smaller, then the sizes of hard disk drives can be setup with the smaller options for the servers of the data warehouse. Apart from that, when the size of the processed data is reduced, the required capacities of hardware such as CPUs and swap spaces become smaller for the related servers to be settled. Moreover, the required time for the retrieval of feature descriptors from a data warehouse apparently decreases when their sizes are reduced.

The second objective of this thesis is to investigate large sets of different visual feature descriptors, different similarity functions, clustering methods to provide a guidance framework for the future research in the construction of weakly-supervised visual object recovery from color images. Since only image-level labeled images are utilized for weakly-supervised learning approaches, the differentiations between positive and negative visual feature descriptors have to be characterized by similarity functions without the information of their pixel-level labeling. Moreover, these differentiations characterized by similarity functions have to contain sufficient information to cluster corresponding visual feature descriptors correctly. Therefore, the combinations of different methods used in the entire pipeline of weakly-supervised visual object recovery are evaluated based on the accuracies of clustering visual feature descriptors extracted from color images.

1.5 OUTLINES

The rest of the thesis is structured as follows: Chapter 2 elaborately reviews three papers introducing the approaches of weakly-supervised visual object recovery from either greyscale or color images. Some ideas presented in these three articles are used or extended for the investigation of this thesis. In Chapter 3, the detailed theoretical and practical explanations of the complete set of techniques investigated for the methodology of the thesis are given. Chapter 4 presents the setup of the experiments and their contributions to the objectives of the thesis. In Chapter 5, the obtained both qualitative and quantitative results according to the experimental setups described in the previous chapter are illustrated and, additionally, comparisons of these results are discussed. Finally, the summary containing the pathways of the research completed in the thesis and the found answers and remedies for the questions and issues raised during the research are provided in Chapter 6. An overview of the future work concludes Chapter 6.

CHAPTER 2

2 RELATED WORK

This chapter presents three papers, which are related to the topic of weakly-supervised learning framework for the recognition of visual objects. All of these three papers mainly contribute the research of this thesis since some of their ideas are included in the research scope of this thesis.

This thesis introduces a set of object models having higher precision scores shown in Chapter 5 than the ones obtained by all the related state-of-the-art methods presented in this chapter. Moreover, the recovery processes of these object models introduced in thesis have faster execution timings and smaller hard disk requirements than the ones utilized in the three articles described in the following subsections of this chapter.

2.1 PROBABILISTIC VISUAL OBJECT MODELING PROPOSED BY SCHMID [21]

The following subsections of Section 2.1 elaborately describe the method of probabilistic visual object modeling presented by Schmid [21] and its relation to this thesis, as well as its ideas utilized in this thesis.

2.1.1 DESCRIPTION OF METHOD

An approach to weakly-supervised learning of visual object models is introduced by Schmid [21]. The target of this method is to generate a distinctive visual object model for a target visual object such as a “textured” animal or a human face. Therefore, the generated visual object model can be utilized for the applications of content-based image retrieval from the large databases of images as well as a target visual object localization for a given set of test images.

Greyscale images are used in the approach proposed by Schmid [21] as both test and training images. Firstly, the grey value descriptors $\mathbf{d}_l$ are produced for each pixel location (x, y) of given five positive and ten negative images by applying the filter shown below:

$$F(x,y,\tau,\sigma) = F_0(\tau,\sigma) + \cos\left(\frac{\sqrt{x^2+y^2}\ \pi\tau}{\sigma}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad 2.1\ [21]$$

This “Gabor-like” filter shown in equation 2.1 also takes the scale and frequency values as parameters σ and τ respectively. The value of frequency τ defines the number of cycles in the harmonic result which is produced by this function. In order to have a DC-balanced waveform, $F_0(\tau,\sigma)$ is added into the function. This addition decreases the effect of illumination changes while the grey value descriptors are calculated. Schmid [21] investigates the results of

13 different filters by changing the values of parameters τ and σ. While the range of values for τ is between 1 and 4, the range of values for σ is between 2 and 10.

Once the generation of the grey value descriptors is completed for each pixel location of all the training images, this set of grey value descriptors is normalized by utilizing their mean and variance. Normalization provides an advantage to decrease the effect of scaling in this visual object modeling technique. Afterwards, the k-means clustering algorithm is applied to the normalized grey value descriptors by using Euclidean distance for the extraction of "generic" descriptors. The k-means algorithm focuses on finding the clusters and their centers for the given feature vectors as samples by iteratively updating the centers which are randomly initialized [22]. The iteration continues until the minimum sum of squares is calculated by using the distance between the samples and their closest centers [22]. The optimum value of k is found as 50 by comparing the distinctiveness of the extracted "generic" descriptors. Once k centers of the normalized grey value descriptors are found by applying k-means clustering, a covariance matrix $\sum$ for each center μ is computed by using the matrix of the only grey value descriptors which are in the cluster of the respective center. Each grey value descriptor, which is assigned to respective closest center, lies on a column vector of this matrix in order to be used for the calculation of the covariance matrix. Therefore, each "generic" descriptor refers to one cluster and is defined with the center and the covariance matrix of that cluster such as $C_i = (\mu_i, \sum_i)$.

A pixel-level labeled image is created by assigning the index of a "generic" descriptor to each pixel location after all "generic" descriptors are extracted from a given training image. The decision of the assignment of each pixel location is made by choosing the generic descriptor, which has the highest probability for the grey value descriptor of that pixel location. For this purpose, the probabilities $P(C_i|\mathbf{d}_l\ )$ of the "generic" descriptors placing each pixel location are calculated for a given grey value descriptor $\mathbf{d}_l$ of each pixel location of a given training image shown as below.

$$P(C_i|d_l\ ) = \frac{P(d_l|C_i)P(C_i)}{P(d_l)} = \frac{P(d_l|C_i)P(C_i)}{\sum_{j=1}^{k} P(d_l|C_j)P(C_j)} \quad 2.2\ [21]$$

Two approximations are used as assumptions in order to compute a probability of "generic" descriptor C_i for a given grey value descriptor $\mathbf{d}_l$. One of these approximations is that the probability of each "generic" descriptor $P(C_i)$ is equal. Another one is used to approximate a distribution of Gaussian to compute the probability of a grey value descriptor for a given "generic descriptors."

Figure 2.1 shows the visualization of some examples of "generic" descriptors, which are manually chosen. To this end, the pixel locations assigned by a selected "generic" descriptor are colored with black while the rest of pixel locations are colored with white in Figure 2.1. A given positive image of which the target visual object is selected as the species of cheetah is shown in Figure 2.1 [a]. The images shown in Figure 2.1 [b] and Figure 2.1 [c] are the visualizations of the examples of two different "generic" descriptors by utilizing the pixel-level labeled image shown in Figure 2.1 [a]. It is easily seen from Figure 2.1 [b] and Figure 2.1 [c] that the locations of pixels, which are grouped by those two different "generic" descriptors, characterize the target visual object. On the other hand, Figure 2.1 [d] shows an example of the visualization of another "generic" descriptor and the pixels which give the highest probability for this "generic" descriptor, and characterize the background.

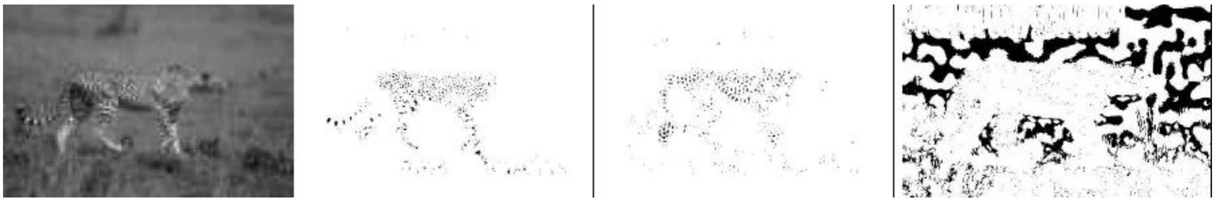

(a) (b) (c) (d)

Figure 2.1: The visualization of some examples of different “generic” descriptors by assigning them to the respective pixel locations in the case of obtaining the highest probability for the respective “generic” descriptors [21].

(a) An original positive sample whose target visual object is determined as the species of cheetah (b)(c) The black colored pixel locations based on two different assigned “generic” descriptors show mostly the pixel locations of target visual object (d) The black colored pixel locations based on a “generic” descriptor show mostly the pixel locations of the background.

The second layer processing is applied into the pixel-level labeled images holding the indices of the “generic” descriptors by utilizing the probability computation shown in the formula 2.2 to generate more distinctive descriptors than “generic” descriptors and rotationally invariant in contrast to the “generic” descriptors. The descriptors, which are obtained by applying the second layer processing into the pixel-level labeled images, are called as “neighbourhood-frequency” descriptors by the author of [21]. In order to generate “neighbourhood-frequency” descriptors, a circular window with radius 10 traverses the pixel-level labeled images by sliding the center window from pixel to pixel. Therefore, a frequency vector $\mathbf{v}_l$ is computed as shown in the formula 2.3 for each pixel location $\boldsymbol{p}$ where the center of circular window stays.

$$\mathrm{v}_l = \begin{pmatrix} P(C_1|w_l) \\ \vdots \\ P(C_k|w_l) \end{pmatrix} = \begin{pmatrix} \frac{|C^*(p)=C_1|\ p\in w_l|}{|w_l|} \\ \vdots \\ \frac{|C^*(p)=C_k|\ p\in w_l|}{|w_l|} \end{pmatrix} \quad 2.3\ [21]$$

In formula 2.3, $\mathbf{w}_l$ is the definition of the centered window, while $|\mathbf{w}_l|$ refers to the size of the window. Once a frequency vector is computed for each pixel location, k-means clustering algorithm is applied to this list of vectors like the extraction of “generic” descriptors. However, this time, k is chosen as 10 instead of 50. Euclidean distance is used for comparing the “neighbourhood-frequency” descriptors in k-means clustering algorithm. Consequently, each cluster refers to one “neighbourhood-frequency” descriptor. After the obtaining of the “neighbourhood-frequency” descriptors, another kind of pixel-level labeled images is created by assigning the indices of “neighbourhood-frequency” descriptors to the respective pixel locations which provide the highest probability for “neighbourhood-frequency” descriptors for given grey value descriptors and frequency vectors of those pixel locations. The computation of “neighbourhood-frequency” descriptor $\mathbf{V}_{ij}$ for a given grey value descriptor and frequency vector of each pixel is shown in formula 2.4.

$$\mathrm{P}(\mathrm{V}_{ij}|\mathrm{v}_l \wedge \mathrm{d}_l) = \frac{\mathrm{P}(\mathrm{v}_l \wedge \mathrm{d}_l|\mathrm{V}_{ij})\mathrm{P}(\mathrm{V}_{ij})}{\mathrm{P}(\mathrm{v}_l \wedge \mathrm{d}_l)} = \frac{\mathrm{P}(\mathrm{v}_l|\mathrm{d}_l \wedge \mathrm{V}_{ij})\mathrm{P}(\mathrm{d}_l|\mathrm{V}_{ij})\mathrm{P}(\mathrm{V}_{ij})}{\sum_s \sum_t \mathrm{P}(\mathrm{v}_l|\mathrm{d}_l \wedge \mathrm{V}_{st})\mathrm{P}(\mathrm{d}_l|\mathrm{V}_{st})\mathrm{P}(\mathrm{V}_{st})} \quad 2.4\ [21]$$

It should be noted that there are two assumptions in the computation shown in formula 2.4 similar to the assignment of the most probable “generic” descriptors to the pixel-level labeled images. One of those assumptions is that the probability of each “neighbourhood-frequency” descriptor P($\mathbf{V}_{ij}$) is equal. The second assumption is to compute the probability of frequency vector for a given grey value descriptor and “neighbourhood-frequency” by approximating with a Gaussian ($\mu_i, \sum_i$).

Finally, the visual object model is built by choosing the positive and distinctive “neighbourhood-frequency” descriptor from positive and negative training images. For this purpose, the significance measurement of each “neighbourhood-frequency” descriptor from all training images is computed. There is an assumption that the probability of each training image is independent and equal. One should bear in mind that inter-pixel dependence is ignored and each pixel has the same probability. Therefore, the probability of a “neighbourhood-frequency” descriptor from a given training image can be computed as shown in formula 2.5. Moreover, the computation of the significance measurement is presented in formula 2.6. The definition of the symbol “I” is an image shown in the formula 2.5, while the symbol of “M” expresses the model in the formula 2.6. The number of pixels in a given image is assumed as “n” and specified in the formula 2.5.

$$\mathrm{P}(\mathrm{V}_{ij}|\mathrm{I}) = \mathrm{P}(\mathrm{V}_{ij}|\{\mathrm{p}_1, \mathrm{p}_2 \dots \mathrm{p}_n\}) = \sum_{l=1}^{n} \mathrm{P}(\mathrm{V}_{ij}|\mathrm{p}_l) = \sum_{l=1}^{n} \begin{cases} \mathrm{P}(\mathrm{V}_{ij}|\mathrm{p}_l) & \text{if } \mathrm{V}^*(\mathrm{p}_l) = \mathrm{V}_{ij} \\ 0 & \text{otherwise} \end{cases} \quad 2.5\ [21]$$

$$\mathrm{Sig}(\mathrm{V}_{ij}|\mathrm{M}) = \frac{\mathrm{P}(\mathrm{V}_{ij}|\{\mathrm{I}_{pos}\})}{\mathrm{P}(\mathrm{V}_{ij}|\{\mathrm{I}_{pos}\}) + \mathrm{P}(\mathrm{V}_{ij}|\{\mathrm{I}_{neg}\})} \quad 2.6\ [21]$$

As it is shown in the formula 2.6, the value of the significance measurement increases while the probability of a “neighbourhood-frequency” descriptor from a given positive image gets closer to 1. Likewise, while the probability of a “neighbourhood-frequency” descriptor from a given negative image gets closer to 0, the value of the significance measurement increases. The range value of the significance measurement is between 0 and 1. Therefore, the “neighbourhood-frequency” descriptors, which cause the computation of significance measurement close to 1, can be categorized as positive and distinctive descriptors for the target visual object.

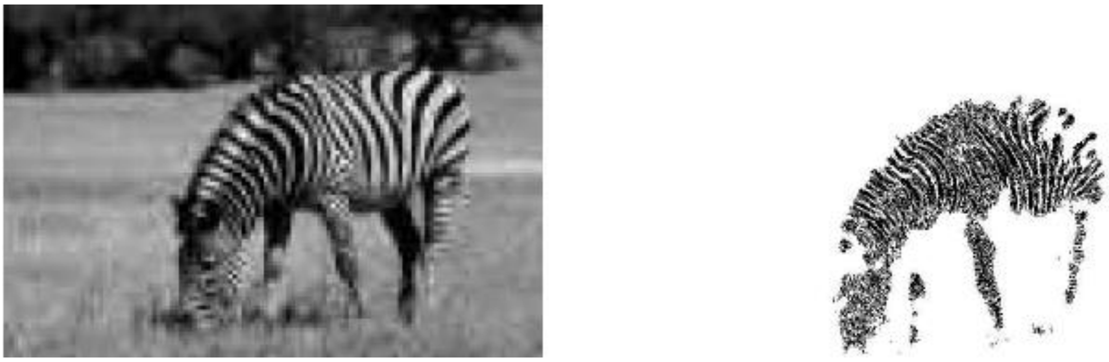
Once a model of a target visual object is constructed by the algorithm explained above, retrieving candidate positive images from a set of test images or localizing the target visual object in the test images can be done by exploiting this model. A target visual object model consists of a number of “generic” descriptors, a number of “neighbourhood-frequency” descriptors and the significance measurements of each of those “neighbourhood-frequency” descriptors. Schmid [21] proposes an image retrieving strategy by using a target visual object model. This strategy is based on computing a probabilistic score for each pixel of the test images by utilizing the descriptors and the significance measurements of the built model. For this purpose, firstly, a grey value descriptor is constructed for each pixel of a given test image. A pixel-level labeled image is constructed by assigning the index of “generic” descriptor of the visual object model which has the highest probability for the given grey value descriptor computed from each respective pixel location. Afterwards, a frequency vector for each pixel is generated by using this pixel-level labeled image. Finally, a significance measurement can be computed by using those frequency vectors and “neighbourhood-frequency” descriptors of the visual object model. The probability of the model for each pixel, which is called a probabilistic score of each pixel $P(M|\mathbf{p}_l)$, is set to 0 if the significance measurement which is computed in the test case is smaller than the specified threshold 0.5. Therefore, the probabilistic score of a pixel is computed as shown in the formula 2.7.

$$\mathrm{P}(\mathrm{M}|\mathrm{p_l}) = \mathrm{P}(\mathrm{C}^*(\mathrm{p_l})|\mathrm{d_l})\mathrm{P}(\mathrm{V}^*(\mathrm{p_l})|\mathrm{v_l} \wedge \mathrm{d_l})\mathrm{Sig}(\mathrm{V}^*(\mathrm{p_l})|\mathrm{M}) \quad 2.7\ [21]$$

In order to have a decision procedure with respect to retrieving candidate positive test images, the probabilistic score of a pixel shown in the formula 2.7 is extended as the probability of a model for a given test image shown in the formula 2.8.

$$\mathrm{P}(\mathrm{M}|\mathrm{I}) = \mathrm{P}(\mathrm{M}|\{\mathrm{p_1}, \mathrm{p_2} \dots \mathrm{p_n}\}) = \textstyle\sum_{\mathrm{l}=1}^{\mathrm{n}} \mathrm{P}(\mathrm{M}|\mathrm{p_l}) \quad 2.8\ [21]$$

The number of pixels in a given test image is fixed and defined as “n”. If the value of the probability of the model for a given test image is greater than the specified threshold, then it is retrieved as a positive test image. Moreover, the probabilistic score of a pixel is used for localizing the target visual object in a given test image by selecting the pixel locations, which give the high probabilistic scores. The Figure 2.2 [a] shows one of the test images, while Figure 2.2 [b] shows the image depicting the pixel locations giving high probabilistic scores in black color and the rest of the pixel locations in white color.



(a) (b)

Figure 2.2: The visualization of the pixel locations which produce the highest probabilistic scores for the given test image [21].
(a) A given test image (b) The visualization of selected pixel locations of given test image based on high probabilistic scores.

2.1.2 RELATION TO THIS THESIS

The overall aim of the method introduced by Schmid [21] is an optimal object model recovery by learning from positive and negative image-level labeled images same as the one of this thesis. There are only two distinct differences between the targets of the method proposed in [21] and the one introduced in this thesis. The first difference is that the method proposed in [21] constructs an object model based on the selected pixels of the image-level labeled images while the approach of this thesis recovers an object model by gathering the selected segments of the image-level labeled images. In other words, the object models which are recovered by using the method of [21] consist of the pixel locations of the image-level labeled images instead of the segment locations of the image-level labeled images. On the other hand, the segment locations of the image-level labeled images construct the object models which are recovered by the method of this thesis. The second difference between the methods proposed in [21] and in this thesis, is that the training and test dataset consists of greyscale images in [21] while color images construct the training and test dataset utilized in this thesis.

2.1.3 IDEAS TAKEN OR EXTENDED FOR THIS THESIS

Instead of utilizing the mathematical functions presented in the paper [21] of Schmid for this thesis, the overall aim of this thesis is derived from the target of [21] by extending to color images and segment-based object model recovery. Moreover, the general idea of retrieving candidate positive test images by using thresholds is applied for the detection of candidate positive test image segments. More specifically a threshold regarding the number of positive pixel locations is set to have a decision procedure for the retrieving candidate positive test images in [21]. Similarly, a decision procedure is used for the detection of candidate positive image segments from test images in this thesis by using a threshold regarding the number of similar positive segments placing in an object model.

2.2 FINGERPRINT TECHNOLOGY PROPOSED BY HAO ET AL. [23]

The details of the method of fingerprint technology presented by Hao et al. [23] is described in the following Subsection 2.2.1. Furthermore, Subsection 2.2.2 shows the relation of [23] to this thesis while Subsection 2.2.3 briefly expresses a list of ideas taken from [23] to be used in this thesis.

2.2.1 DESCRIPTION OF METHOD

Hao et al. [23] present an innovative recognition/classification tool for bioacoustics to monitor insects by operating the spectrograms of their sounds. The spectrogram is defined as an acoustic descriptor, which visualizes the magnitude of Short-time Fourier Transform (STFT) [24]. Since the typical sound signals produced by organisms comprise varying characteristics of the batch of formants for each particular time frame over time, there is a requirement for a technique, which examines the signals separately extracted from the sequential time frames instead of examining the entire signals at once. For this purpose, STFT is designed with the expansion of the employment of the domains for discrete-time Fourier transform (DTFT) and discrete Fourier transform (DFT) [24]. The extended domain is the combination of time and frequency domains. While both DTFT and DFT are applied to the entire input signal, STFT is applied to equally divided sections of the signal over time to generate sinusoidal outputs in the frequency domain [24]. DTF maps a finite discrete set of complex numbers taken from the time domain to a finite set of coefficients, which are complex numbers in the frequency domain [24]. On the other hand, DTFT converts the infinite discrete set of complex or real numbers taken from the time domain to the continuous and periodic function in the frequency domain [24].

Since both DTFT and DFT take discrete numbers as input, a continuous signal has to be sampled with the determined minimum sampling frequency by Nyquist rate [22]. Nyquist-Shannon sampling theorem ensures that the sampled discrete signal is sufficiently representative to recover the original continuous signal [22]. From the practical point of view, the sampling process is executed by multiplying the continuous signals with the drain of pulses based on the Nyquist rate. The spectrograms are the intelligent way of representing in aid of two-dimensional space of the time varying spectral data produced by the application of STFT into the sound signals which require the four-dimensional space of representation due to the application of DTFT or DFT into small timeframes over time. When the computation results of the DFT or DTFT concerning small timeframes are restricted with magnitude, then the required numbers of dimensions are diminished to three from four dimensions for the representation space. To interpret the results even easier, the spectrograms are depicted as greyscale images, which code the magnitude values of the computation results of DFT or DTFT as the level of hues. For instance, the dark hues represent the small values while the bright hues are used to describe the high values of the magnitude. Figure 2.3 presents a sample spectrogram, which maps an insect sound demonstrated in [23].

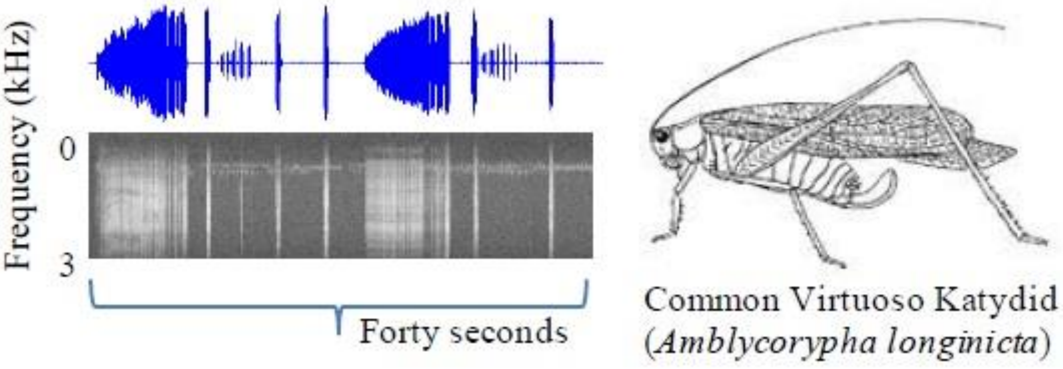


Figure 2.3: A spectrogram mapping an insect sound for a specific time frame [23].

Hao et al. [23] mention about another research of acoustic recognition also using the spectrograms, which is introduced by Mellinger and Clark [25]. The purpose of this research is to build an automatic acoustic recognition tool for the bowhead whales, but in aid of supervised learning method since the training stage requires a significant amount of only labeled samples. All those training samples are the selected divisions of the spectrograms extracted from the conclusive records of the songs of the bowhead whales. The conclusive records take place once or a sequential number of times at the end of each separate song of the bowhead whales [25]. Apart from that, the time frame of each separate song of the bowhead whales is approximately

60-70 s while the interval between two separate songs is about 5-15 s [25]. The conclusive records from the songs of the bowhead whales are chosen as selected divisions of the spectrograms because those divisions contain more distinctive frequencies, in other words: higher frequencies due to the produced louder songs by the bowhead whales.

During the training stage of this proposed automatic recognition tool, a correlation kernel is built by using the averaged consequent image of the selected divisions of the spectrograms based on the conclusive records from songs of the bowhead whales. The parameters of the spectrograms related to the time and frequency are chosen in [25] such that the contours of the frequencies are distinctly observable in those spectrograms. After the generation of the correlation kernel, the proposed automatic recognition tool computes the maximum cross-correlation coefficient between each test division of the spectrograms and the correlation kernel. This computation provides a function of time, which gives the detection scores as results. Mellinger and Clark [25] investigate different threshold values for those detection scores in their work to detect test divisions which are the conclusive actual records of the songs of the bowhead whales. The correlation coefficient is a statistical tool to find the quantity, which presents the nature of the relationship between two random variables [26]. Since a relationship between two random variables is defined as positive or negative depending on the sign of the quantity of the covariance [27], the covariance does not provide any information concerning the strength of the relationship. Therefore, to get more information about the strength of the relationship, the quantity of the covariance is divided by the value of the multiplication of the standard deviations of both random variables [27].

Hao et al. [23] emphasize the large number of parameters, which are required to be adjusted to build the automatic recognition tool proposed by Mellinger and Clark [25] as one of the drawbacks. The limitations on the invariance of both local and global transformations of the time are also highlighted by Hao et al. [23] as another drawback of the approach of Mellinger and Clark [25]. Hao et al. [23] specify that these limitations occur due to the nature of the correlation coefficient, which is fundamentally linear. Therefore, to prevent the construction of the recognition/classification tool from overfitting [27], Hao et al. [23] apply CK-1 distance measurement introduced by Campana et al. [28] for the comparison of the subsets of the spectrograms between each other. These spectrograms are captured from the sounds of the insects. To compute the measure of CK-1 distance between two samples, it is not necessary to extract any feature from those samples. Instead, CK-1 distance is computed by directly exploiting those samples as inputs. Moreover, the computation of CK-1 distance is not required to be aligned by setting any parameter; therefore, CK-1 distance supplies a parameter-free measurement between two samples.

Campana et al. [28] mention about the approximation of the Kolmogorov complexity with the generalization of different compression algorithms to point out the fundamental structure of the CK-1 distance. The definition of the Kolmogorov complexity [29] is the length of the shortest string s which is the formal implementation of an algorithm created by a primitive programming language without any input based on Church-Turing Thesis [30] in order to print out a specified string x. In this case, the Kolmogorov complexity is presented as $K(x)$. In other words, Kolmogorov complexity as a function provides a quantity as an output based on the encoding quality of a string, which is requested to be printed out and is given to the function as an input. When the input string gets more random, the length of the program becomes larger.

Moreover, the interpretation of $\mathrm{K}(\mathrm{xy})$ is the function of the Kolmogorov complexity taking the string xy as an input which is the concatenation of the strings x and y. As it is specified in the above definition of Kolmogorov complexity, there is no input for the program which prints out the string x. This notion is extended by adding a string y as an input for the program printing out the string x and introduced as a conditional Kolmogorov complexity $\mathrm{K}(\mathrm{x}|\mathrm{y})$ which is also mentioned in [28]. Conversely, $\mathrm{K}(\mathrm{y}|\mathrm{x})$ defines the Kolmogorov complexity of the program to print out the string y while the string x is given as an input. By using Kolmogorov complexity $\mathrm{K}(\mathrm{xy})$ with the conditional Kolmogorov complexities $\mathrm{K}(\mathrm{x}|\mathrm{y})$ and $\mathrm{K}(\mathrm{y}|\mathrm{x})$, the distance d_k between two strings x and y is defined as the following formula 2.9:

$$\mathrm{d}_{\mathrm{k}}(\mathrm{x}, \mathrm{y}) = \frac{\mathrm{K}(\mathrm{x}|\mathrm{y}) + \mathrm{K}(\mathrm{y}|\mathrm{x})}{\mathrm{K}(\mathrm{xy})} \quad 2.9\ [28]$$

Since the computation of Kolmogorov and conditional Kolmogorov complexities does not require the setting of any parameter to provide the quantity for a given program, the above-presented distance d_k is a parameter-free distance measurement function. Campana et al. [28] present a correspondence between the Kolmogorov complexity and universal string compression by transforming the problem of Kolmogorov complexity into the problem of universal string compression. More formally, the new problem of universal string compression is reduced to an already known problem of Kolmogorov complexity; therefore, the problem of universal string compression is not harder than the problem of Kolmogorov complexity. The correspondence between those two problems is initially and informally introduced, but the formal definition and the proof of this reduction is not described in the work of Campana et al. [28]. Accordingly, the formal definition and proof of this reduction is presented in Appendix A.1 of this thesis.

According to this problem reduction, the distance formula of the universal string compression for given strings x and y can be presented as formula 2.10 by making use of the settings of the problem reduction.

$$\mathrm{d}_{\mathrm{c}}(\mathrm{x}, \mathrm{y}) = \frac{\mathrm{C}(\mathrm{x}|\mathrm{y}) + \mathrm{C}(\mathrm{y}|\mathrm{x})}{\mathrm{C}(\mathrm{xy})} \quad 2.10\ [28]$$

The function of the universal string compression for the given string x is presented as $\mathrm{C}(\mathrm{x})$ which returns the size of the compressed string after the execution of the compressor on the string x. In the formula 2.10, the function of $\mathrm{C}(\mathrm{x}|\mathrm{y})$ provides the compressed size of the string x whose compressor is trained on the string y. In the same way, $\mathrm{C}(\mathrm{y}|\mathrm{x})$ returns the size of the string which is obtained after the execution of the compression on the string y by using the compressor which is previously trained on the string x. Moreover, $\mathrm{C}(\mathrm{xy})$ is the function which takes a string xy obtained by concatenating strings x and y as an input and distributes the size of the compressed string after the execution of the universal string compression on this concatenated string xy.

The proof of this reduction also provides such information that the complexity of the distance formula of the universal string compression is not harder than the complexity of the distance formula of Kolmogorov complexity. However, the body of the distance formula of the universal string compression is not sufficient to be used for the domains of real numbers.

Because, the lossless compression techniques do not act based on the repeated structure in the domains of real numbers; conversely, the lossless compression techniques replace the repeated structures in the discrete data with the symbols which have shorter lengths than those in repeated structure [31]. Therefore, the formula 2.10 is not unswervingly applicable to real-valued images or videos as a distance formula.

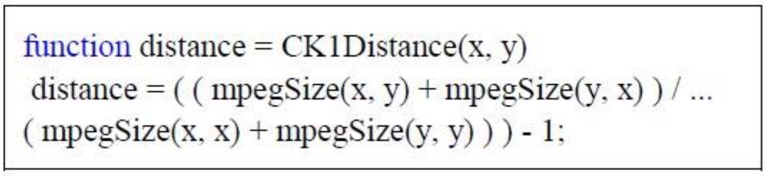
Since the goal of the research introduced in [28] is to build an effective and parameter-free distance measurement for the texture similarities of the given samples of the real-valued greyscale images, the distance formula of the universal string compression has to be modified for the usage of the domains of real numbers. Furthermore, the functions of the universal string compression in the distance formula have to be replaced with a lossy compression function which does not require any parameter alignment for its execution and succeeds based on the similarities of the given samples. For this purpose, Campana et al. [28] extend the idea of the distance function of universal string compression by exploiting of the video encoding format of MPEG-1 and call this new distance function as CK-1 distance. MPEG-1 is a standardized lossy compression format for the multimedia data published by Motion Picture Experts Group (MPEG) in the early 1990s in order to store a video or audio in the digital media with the bit rates which are up to about 1.5 Mbit/s [32].

Campana et al. [28] mention two basic encoding steps of the video compression techniques, which are intra and inter frame compressions. The intra frame compression succeeds by encoding the similarities in the single frame, while the similarities between frames are encoded by the inter frame compression [32]. Campana et al. [28] express that if a video is created with only two image frames and then compressed into a video encoding format, then the size of the encoded video varies based on the similarities between those two image frames due to the inter frame compression step of the video compression technique. Accordingly, Campana et al. [28] gather the set of Matlab movie frames by converting the dataset of the research which is color image frames. In order to provide color invariance for the similarity measurements of the textures from the dataset, the color image frames are firstly transformed into grayscale image frames and then those grayscale image frames are transformed into Matlab movie frames. Each of those two Matlab movie frames from the transformed dataset, which are required to be checked concerning their similarities between each other, are firstly used to form four different Matlab videos after the transformed dataset is gathered as a set of Matlab movie frames. One should assume that the first considered Matlab movie frame is called x while the second one is called y. One of those four Matlab movies is created with the Matlab movie frames x and y in a consecutive manner and the respective temporal order, while another one is created with the same Matlab movie frames but in the reverse temporal order. Apart from them, the same Matlab movie frame x is sequentially repeated in the temporal axis in order to form the third Matlab movie while the fourth Matlab movie is formed in the same manner but with the Matlab movie frame y instead of x. After all those four Matlab movies are created, they are separately compressed into MPEG-1 format. Consequently, the algorithm of CK-1 distance function uses the sizes of these compressed videos as essential arguments for the computation of distance measurements between the samples since MPEG-1 video compression runs a temporal reduction on the bits in addition to the spatial reduction by keeping only the differences from the reference frame [33].

The first image frame is assigned as a reference image frame and the second image frame is designated as a predicted image frame regarding the temporal order during the encoding process of a video via MPEG-1 compression. Hence, four videos created with the different temporal order of image frames are required to construct a symmetric distance function which is exploiting the ratios of the sizes of those four compressed videos. In other words, the size of a compressed Matlab movie into MPEG-1 format provides an asymmetric measurement for the similarity between the textures of the Matlab movie frames which form this Matlab movie; therefore, four Matlab movies are required to build a symmetric distance function taking two Matlab movie frames as inputs. In the light of this approach, Campana et al. [28] introduce the body of the symmetric MPEG distance function with the following equation 2.11.

$$d_{mpeg}(x,y) = \frac{C(x|y)+C(y|x)}{C(x|x)+C(y|x)} - 1 \quad 2.11\ [28]$$

The function of C from the above-shown equation 2.11 is replaced in [28] with the function of mpegSize which returns the size of the MPEG-1 format of the compressed Matlab movie with the same approach explained in the previous paragraph for two given samples from the dataset and called CK-1 distance. Apart from that, CK-1 distance is coded in Matlab which can also be understood from the transformed dataset mentioned during the explanation of the approach in the previous paragraph. Moreover, Campana et al. [28] exclude the step of the storage into the digital media from the compression format of MPEG-1 in order to boost the execution of the function of mpegSize. The Matlab code of the function of CK-1 distance implemented in [28] is shown in Table 2.1.



```
function distance = CK1Distance(x, y)
 distance = ( ( mpegSize(x, y) + mpegSize(y, x) ) / ...
( mpegSize(x, x) + mpegSize(y, y) ) ) - 1;
```

Table 2.1: The implementation of CK-1 distance function in Matlab which exploits the inter frame compression step of MPEG-1 format for the symmetrically obtained reference and predicted movie frames [28].

Apart from the property of symmetry, CK-1 distance function returns zero when the same images are given as inputs; otherwise, always returns a positive value. Therefore, another property of CK-1 distance function is being nonnegative. The distance function of CK-1 is also gained with the property of rotation invariance by the additional implementation in the code of the function. This additional implementation enables the function to measure the value of the minimum CK-1 distance between one image with fixed standing and another image, which is rotated at ten different angles. When one of the input images is rotated at an angle, which is different than 90°, 180° or 270°, then the rotated image is not placed in its initial spatial coordinates. For this purpose, Campana et al. [28] utilize three different procedures such as cropping, padding and scaling to fix the issue of not fitting pixels of the rotated image with their initial positions.

Image cropping facilitates the improvement of focusing on the rectangular division of the interest [34]. Since there is a possibility that some or all the relocated pixels do not stand on the pixel coordinates after the scaling of a given image, one of the interpolation methods is required to be used to compute the values of those relocated pixels [34]. Image scaling, which is one of the affine transformation functions, is operated to resize an input image by preserving the points, lines and especially parallel lines [35]. Image padding is a method to extend the boundaries of a given image with additional pixels due to the filtering or representation purposes [36]. These additional pixels are set to a standard value, which is chosen as a minimum, maximum or intermediate from the range of 8-bit grayscale pixel values between 0 and 255 [36]. If the additional pixels are set with zeros, then this type of padding is called as zero-padding [37]. Mirror image is the reflected representation of the original image based on its edge [22].

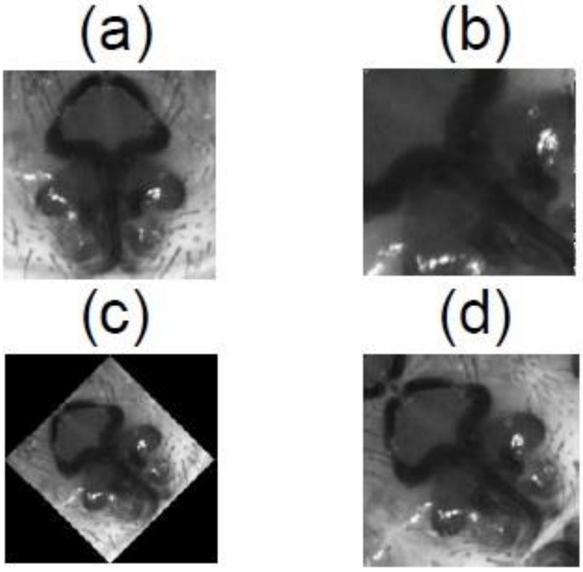


Figure 2.4: The presentation of the samples created by different techniques in order to bring CK-1 distance function in the property of rotation invariance [28].
(a) Initial sample image (b) The result image after the cropping the rotated image from its centroid and then resizing based on the shape of initial sample image (c) The result image after the resizing of the rotated image and the application of the zero-padding (d) Mirrored image from the rotated image based on a specified mirror.

In the work of [28], the mirror images are obtained by flipping over a given image based on a reference mirror. Apart from that, Campana et al. [28] employ the cropping method by cropping the images from the dataset around their centroids. These procedures, which are applied by Campana et al. [28] to fit the coordinates of the rotated image with its initial coordinates, are shown in Figure 2.4. The initial sample image from the dataset is shown in Figure 2.4 [a]. After the rotated image is cropped around its centroid, the cropped image is resized based on the shape of the initial sample image shown in Figure 2.4 [b]. As it is presented in Figure 2.4 [c], the rotated image is scaled with such a maximum ratio that the resized image stands between the boundaries of the location of the initial sample image. Moreover, the pixels, which are in the region between the boundaries of the location of the initial sample image and the outer of the boundaries of the rotated image shown in Figure 2.4 [c], are assigned to the value of zero by the method of zero-padding. Figure 2.4 [d] shows a sample of the procedure of the image mirroring for the given image from the dataset.

Finding a subset from the training subsets of the spectrograms as the most discriminative visual object model of which the dimensions are the smallest for the target visual object underlies all the functionalities of the bioacoustic recognition/classification tool introduced by Hao et al. [23]. This subset, which is found as the most discriminative visual object model, is called “sound fingerprint” for the respective target visual object by Hao et al. [23]. One should note that the subsets of the spectrograms are not labeled as positive or negative for a target visual object based on the existence or absence of the target visual object for the respective subsets of the spectrograms. Instead, only spectrograms in the dataset are labeled as positive or negative. Each spectrogram in the dataset is labeled as positive if it possesses at least one subset containing the target visual object; otherwise, it is labeled as negative.

As it is noticeable from the labeled dataset, the issue of finding a sound fingerprint for a target visual object is a typical example of the problems of the weakly-supervised learning. For this purpose, Hao et al. [23] build a decision tree from the dataset of subsets/fragments of spectrograms. Since a decision tree contains the tasks of data compression and prediction, a constructed decision tree from a chosen dataset can be utilized to predict either unlabeled or labeled samples from the same chosen dataset as well as the samples from another dataset [38]. During the construction of a decision tree from the training dataset, Hao et al. [23] propose to use an idea of data mining called information gain as a decision rule. In order to diminish the computational complexity of the construction of a decision tree from the training dataset, CK-1 similarity measurements between the fragments of spectrograms are exploited by Hao et al. [23] instead of any extracted features from the fragments of spectrograms. Furthermore, CK-1 similarity measurements diminish the computational complexity of the extraction of the required classification information from the dataset by making use of the constructed decision tree. Therefore, the constructed decision tree from the training data used for the research of Hao et al. [23] comprises only two leaves due to the representation of CK-1 similarity measurements between the fragments of spectrograms in a single dimension. CK-1 similarity measurements between each candidate of sound fingerprint and the rest of the training dataset are calculated and represented on a number line. Afterwards, the threshold value on the number line is chosen in order to split the training dataset into two subsets for the purpose of linear classification based on the decision rule which is maximizing the result value of bits obtained by information gain.

The definition of entropy has to be provided beforehand to describe information gain. Entropy provides a value of bits as a measurement based on the amount of ambiguity existing in the content of information obtained from the dataset due to the possibility of multiple classification options [38]. The information gain is defined as the value of bits which are measured as a result by calculating the difference between the entropies of the initial dataset and the subsets acquired after the division of the initial dataset [38]. In the light of the above description related to the whole process of building a decision tree from the dataset, Figure 2.5 shows the ordered sample fragments of the positive and negative spectrograms on the number line. They are ordered based on the computation of the CK-1 distance between the sample candidate fragment and other sample fragments extracted from the positive and negative training dataset.

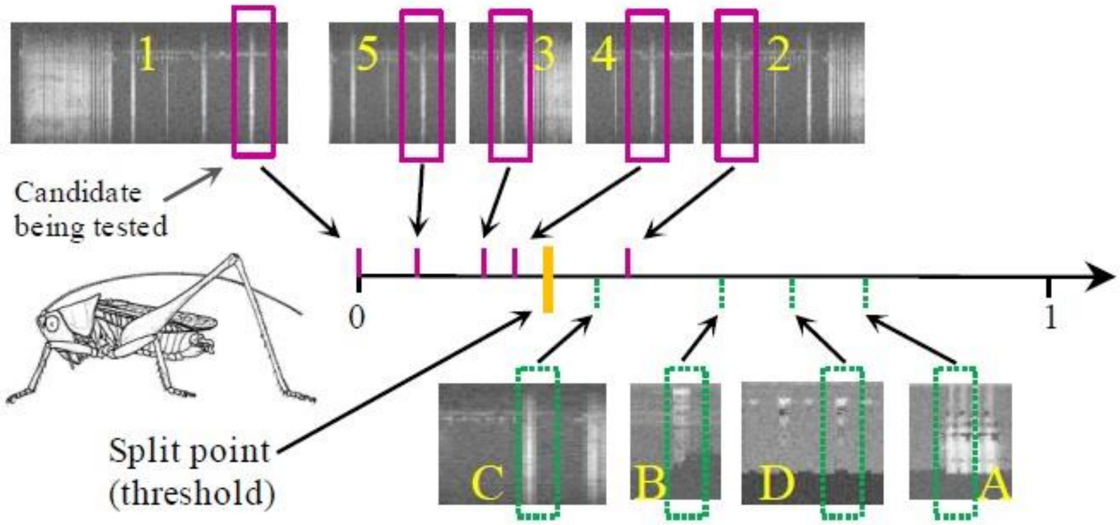


Figure 2.5: The illustration of maximizing the information gain for some samples of the lined up positive and negative fragments of the spectrograms on a number line based on their CK-1 distances to a sample of candidate fragment [23].
While the violet boxes point out the samples of positive fragments of the spectrograms, the negative ones are indicated by the green boxes. The yellow large vertical stripe depicts the location of the point splitting the given dataset into two subsets in such a manner that the computed value of information gain according to this splitting point is the maximum value that can be obtained from all possible splitting points by using this given dataset.

After the maximum values of the computed information gain for the whole set of candidate fragments are obtained, the candidate fragment holding the maximum value of the information gain is set as the best fingerprint for the given dataset. There is also another additional selection criterion among the candidate fragments although their maximum values of the computed information gain are the same. In this case, the candidate fragment of the spectrogram possessing the longer distance between the medians of the positive and negative subsets placed on a number line according to their CK-1 distances to this candidate fragment is selected, instead of another fragment having the same maximum value of the information gain. For instance, it is identifiable in Figure 2.6 that the distance between the means of groups of green and violet lines placing in the lower number line is longer than the one in the upper number line. Therefore, the candidate fragment computed in the lower number line is selected as a better fingerprint rather than the one shown in the upper number line although the values of their information gains are equivalent.

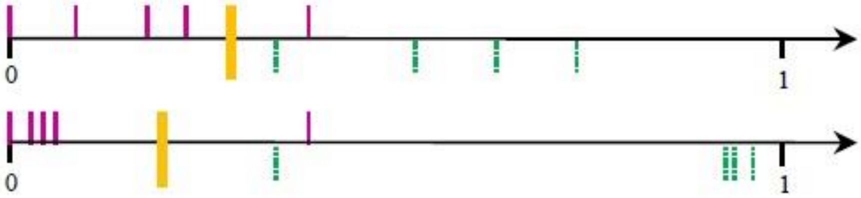


Figure 2.6: The number line representations of CK-1 distances between the sample fragments and two different selected fragments having exactly the same information gain [23].

2.2.2 RELATION TO THIS THESIS

The learning strategy of the approach proposed by Hao et al. [23] is a weakly-supervised learning same as the ones of the previously presented paper of Schmid [21] and the approach of this thesis. The overall aim of [23] is to find a single sound fingerprint belonging to a target visual object by using image-level labeled images while an optimal object model holding minimum 25 number of the best fingerprints is recovered in this thesis by utilizing the same approach of [23]. Segment-based fingerprint selection is performed by the both approaches introduced in this thesis and in [23].

Hao et al. [23] give 77% and 93% insect classification accuracies for 10 and 20 different insect genera, respectively. A wide range of frequency sounds is produced by the various species of insects based on many different constraints (e.g. body size and temperature) [39]. Apart from that, there are also various sounds that can be produced by only a single species [23]. When those complexities of the insect sound and the experimental results presented in [23] are taken into consideration, the approach proposed by Hao et al. [23] is set as an initial fundamental idea of this thesis and extended to color images.

2.2.3 IDEAS TAKEN OR EXTENDED FOR THIS THESIS

Since the initial approach of this thesis is originated from the approach proposed by Hao et al. [23], a complete set of methods described in Subsection 2.2.1 and utilized for object model recovery in [23] is applied in this thesis as well. One should note that the dependencies of these methods in the entire pipeline of object model recovery are same for the approach of this thesis and the approach of [23]. The classic machine learning idea of information gain, which is also used in [23], is performed for each experimental run of object model recovery in this thesis. Moreover, a set of methods which are different than the ones utilized in [23] is also investigated for all the processing steps placing in the entire pipeline apart from the one containing the application of information gain.

While Hao et al. [23] make use of spectrograms, which are the spectral representation of acoustic signals; real-world color images are used as a pool of either test or training datasets in this thesis. CK-1 distance which is utilized in [23] as a tool for building an object model is applied to image segments in this thesis. Furthermore, there are a set of experimental setups which do not contain the application of CK-1 distance to build an object model in this thesis. When CK-1 distance method is not applied in an experimental setup of this thesis, it is replaced with a feature extraction method and a similarity metric.

2.3 MULTI-FOLD WEAKLY-SUPERVISED LEARNING PROPOSED BY CINBIS ET AL. [40]

The following Subsection 2.3.1 explains the details of the method of multi-fold weakly-supervised learning presented by Cinbis et al. [40]. Moreover, the relation of [40] to this thesis is described in Subsection 2.3.2. Finally, Subsection 2.3.3 presents a list of ideas which are taken from [40] are included within the scope of the research of this thesis.

2.3.1 DESCRIPTION OF METHOD

Cinbis et al. [40] propose a weakly-supervised object localization approach, which makes use of multiple instance learning. This approach is applied to color images taken from PASCAL VOC 2007 and 2010 datasets [41]. In multiple instance learning, each object, which is sent to the learner, is represented as a bag of vectors [42]. One should note that the number of vectors could vary and each vector is also named as an instance. Each object is labeled either as positive or negative in multiple instance learning [42]. The decision to set an object as a positive object is taken in the case at least one positive vector exists in its representation [42]. If all vectors of the object representation are negative vectors, then the respective object is labeled as a negative object [42]. Multiple instance learning provides a weakly-supervised learning framework for image-level labeled images by viewing them as bags of vectors [43].

For the case of the approach introduced by Cinbis et al. [40], each object is defined as a window which is re-localized in previously divided disjoint folds of a training image or as a training window. Moreover, the number of disjoint folds is determined at the beginning and the windows, which are used for training, are extracted from other folds of training images than the windows used for re-localization related to each fold. The global representation of each window is created by assembling the extracted local Shift descriptors from each respective window. To generate this global representation, the local Shift descriptors are mined into 64 dimensions by applying Principle Component Analysis (PCA). The goal of the PCA method is to diminish the number of dimensions showing such data that the dimensions of the projected space can present the variation of the data with the most distinguishable manner [9]. Sanches et al. [44] introduce Image Classification with the Fisher vector (FV). The experimental results shown in this journal demonstrate the improvement in the accuracies by applying the PCA method on the local descriptors with the reduction to 48 dimensions or higher dimensions as depicted in Figure 2.7. In fact, the difference in the accuracies between the reduction to 64 and 128 dimensions is computed approximately less than 0.3% based on those experimental results. The PCA dimensionality is set as 64 for both works presented in [44] and [40]. The experimental results of the method introduced in [44] are generated using the color images of PASCAL VOC 2007 [41], which consists of 20 classes. The Average Precision (AP) is computed for each class of this dataset with different size of PCA dimensionalities. Afterwards, the mean value of Average Precision (Mean AP) is computed by taking the average of the values of AP related to 20 classes for each PCA dimensionality. Those Mean AP values are shown in Figure 2.7.

In the approach introduced in [40], after the local Shift descriptors of the windows are projected to lower dimensions such as 64 by using PCA, Gaussian Mixture Models (GMM) is utilized to build “probabilistic visual vocabularies” keeping 4, 64 or 1000 components from these computed principal components [44]. GMM is a parametric probability density function to model a given data, which has a specified number of subpopulations by summing weighted Gaussians [22].

Afterwards, a Fisher Vector (FV) with the size of 16 is constructed by measuring the gradient of its log-likelihood concerning the already built GMM. Fisher Kernel [44] shows the directions in gradient space to align the parameters of the model of the distribution; therefore, adapting the model through the parameters can represent a sample. A global representation of each respective window is obtained by applying l_2 and power normalization to the constructed FV.

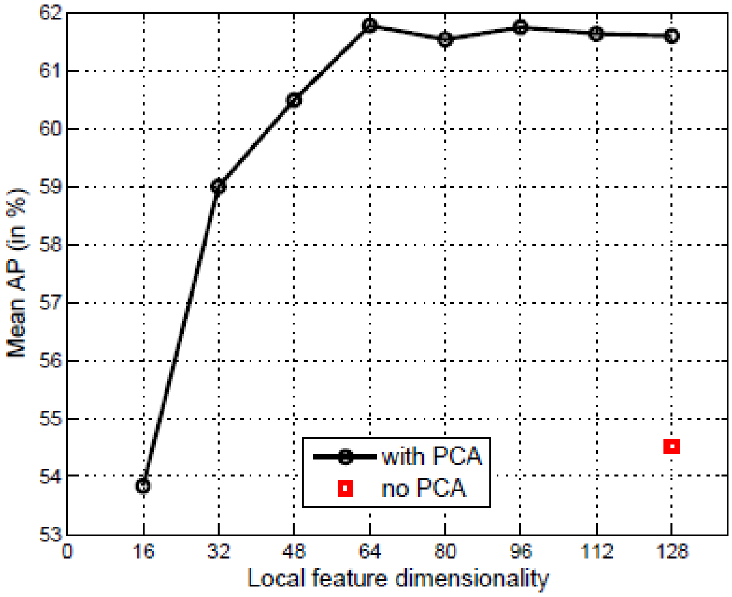


Figure 2.7: The visualization of the increase in Mean AP as a consequence of the dimensionality reduction while PCA is applied into the local SIFT Descriptors [44].

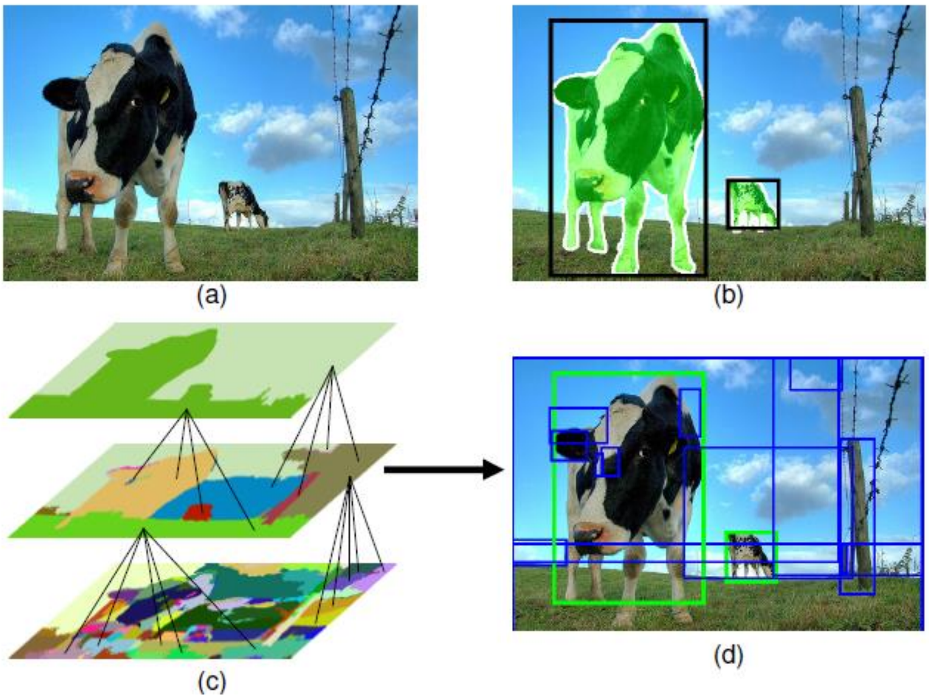


Figure 2.8: The visualization of the Segmentation as Selective Search proposed by Uijlings et al. [45].

(a) A given test/training image. (b) The ground truth of the given test/training image (c) The hierarchically merging strategy of segments, which are generated by over-segmentation during the initial level of the hierarchy (d) The bounding boxes covering some example selected segments

Each initial window is localized from the center of the entire training or test image with at least 4 percent smaller size than the size of the image. Afterwards, those initial windows are divided randomly into disjoint folds with the previously determined number. The classifier is trained with those divided windows from positive training images. The number of training or test windows is diminished by utilizing the approach of Selective Search introduced by Uijlings et al. [45]. The target of this pioneering approach is to facilitate the tasks of the feature extraction and classification units in an ordinary object recognition scenario. For this purpose, the pre-processing step of the object recognition, which is segmentation, is adjusted in such a way that some background candidate segments are eliminated in this upgraded segmentation unit without waiting for the classification unit. Consequently, this improvement intrinsically enables other processing steps placing in the entire pipeline to work more elaborately due to the reduction of the dimensions of respective input data.

The approach introduced by Uijlings et al. [45] initially applies over-segmentation to a given test or training image. When segments are gathered with larger regions, they contain more information. Therefore, the segments collected with larger regions are more convenient to find similarities between each other in different variations, e.g. illumination conditions and noise. However, the probability of the existence of multiple target objects is getting higher when the region of a segment becomes larger. Accordingly, over-segmentation stabilizes the choice of the size for each segment by taking into account those pros and cons [46]. After the segments are generated with over-segmentation for the given training or test images, the similarities between them are computed in ways and by means discussed in [46]. According to the similarity computation between segments explained in [46], both size and texture similarities are employed by summation. The result ranges of the computation of both similarities vary between 0 and 1; therefore, the effects of both similarity computations are the same on the global similarity measurements between the segments. The size similarity between two segments is assigned by computing the proportion of the region formed by joining those segments based on the entire image. Moreover, in order to compute the similarity based on textures, SIFT descriptors [47] are extracted in each segment. On the ground of the approach [48], the extraction of SIFT descriptors is applied into each color channel separately and concatenated as a batch of RGB-SIFT descriptors. Afterwards, the distances between the batches of those RGB-SIFT descriptors are computed by making use of the distance function of histogram intersection [49].

The hierarchically merging process of the method introduced by Uijlings et al. [45] starts with the initially generated segments from the given test or training images by applying over-segmentation. After that, the similarity measurements between adjacent segments are computed, and the segments having the smallest similarity value are merged. The similarity measurements between the newly merged segment and its neighbors are generated. This process continues until the entire image is represented as a single segment shown in the (c) of Figure 2.8. The target visual objects can be in different scales from given images [45]. The hierarchical strategy of this method enables the locations of target visual objects in different scales. The combination of multiple color spaces is also investigated in this method. Apart from mulling over the only segments as selections by the method, the bounding boxes covering those segments are also investigated by Uijlings et al. [45] shown in (b) and (d) of Figure 2.8.

As it is mentioned previously, the positive training images are divided into disjoint folds randomly based on the approach introduced by Cinbis et al. [40]. During the iteration of each fold, the first detector is trained by using the windows extracted from other folds. Afterwards, the re-localization of the windows is applied for the same fold with this already trained detector. After the iteration of each fold reaches the number of folds defined as 10 in the beginning, the second detector is trained by employing the extracted windows from all disjoint folds of positive training images. The second detector is used to improve the classification accuracy of the first detector by catching the false positives and training the first detector with them once again. The false positive examples are called "hard examples" by Dalal et al. [50]. During the investigation of the method of HOG (Histograms of Oriented Gradients) by Dalal et al. [50] to detect humans from giving test images, the final detector is generated by training the already trained detector with hard examples once again. The final detector in the approach introduced by Cinbis et al. [40] is also formed by the same procedure with one difference. In the research of Dalal et al. [50], the negative training examples are the patches extracted from the negative test images, and it is known that a target visual object exists in those patches. Therefore, based on the results of the classifier for those given negative examples, the false positive ones can be determined. However, for the approach of Cinbis et al. [40], the negative examples are not known due to the natural consequence of the weakly-supervised process. As a matter of fact, the second detector determines the false negative examples.

The contextual information from the training or test images is utilized by the method introduced in [40] as an additional feature to improve the performance of the localization of the target visual objects. For this purpose, the foreground, background and contrastive descriptors are investigated. The foreground descriptor is the Fisher Vector, which is developed with the extracted SIFT descriptors from the visual object window. The SIFT descriptors, which are extracted from the complement visual object window (the image area between the full training/test image and visual object window) are used to form the Fisher Vector for background descriptor. The contrastive descriptor is established as the difference of Fisher Vector extracted from background and Fisher Vector extracted from foreground as a 1×1 vector. In the light of the results generated during the investigation of different techniques and features, the highest accuracy of the target visual object localization and the highest mean value of the Average Precision for 20 classes of target visual objects are calculated when the contribution of the foreground and contrastive descriptors is determined as shown below in Table 2.2.

	MIL			**Multi-fold MIL, K=10**		
Descriptors	F	F+B	F+C	F	F+B	F+C
CorLoc	29.1	29.8	29.7	36.5	38.5	**38.8**
Detection mAP	14.0	15.6	15.5	20.0	21.0	**22.4**

Table 2.2: The performance comparison between Multiple Instance Learning (MIL) and Multi-fold Multiple Instance Learning based on the criteria such as "CorLoc" and "Detection mAP" by making use of foreground (F), background (B) and contrastive (C) features [40].

"CorLoc" defines the accuracy percentage related to the correct target visual object localization for given training images. "Detection mAP" shows the percentage of the correctly detected target visual objects for given training images.

2.3.2 RELATION TO THIS THESIS

The approach proposed by Cinbis et al. [40] also utilizes a weakly-supervised learning framework for the recognition of visual objects like the approaches of the previously presented related papers in this chapter and the approach proposed in this thesis. Moreover, inputs of the both approaches of [40] and this thesis are training and test datasets which consist of only color images. Therefore, the targets of the methods of Cinbis et al. [40] and this thesis completely correspond to each other.

2.3.3 IDEAS TAKEN OR EXTENDED FOR THIS THESIS

The research of this thesis uses the idea of applying PCA to local feature descriptors, which is presented in [40]. While Cinbis et al. [40] apply PCA to Fisher vectors obtained from the image segments, the approach of this thesis applies PCA to either a type of color histograms or Vector of Locally Aggregated Descriptors Encodings. The following Chapter 3 describes these feature descriptors and their usage in this thesis. Apart from that, the idea of performing l_2 and power normalization on the principal components obtained from local feature description by the application of PCA is taken from [40] for the investigation of this thesis.

CHAPTER 3

3 WEAKLY-SUPERVISED LEARNING OF VISUAL OBJECT MODELS

In this chapter, the elaborated description of the weakly-supervised learning framework proposed by this thesis is introduced. The target of this weakly-supervised learning framework is to recover an optimal object model from a training dataset containing only image-level labeled color images. After an optimal object model is built by the weakly-supervised learning framework of this thesis, it is used to detect the respective target visual object from a test dataset containing positively image-level labeled color images.

Since collecting the most discriminative parts of the target visual objects is in interest, the datasets are sets of local properties of the images instead of global properties. In order to accomplish this fundamental objective, two different methods such as sliding windows and mean shift segmentation [51] are investigated. As demonstrated in Chapter 2, Hoa et al. [23] extract the fragments from the spectrograms by using the sliding window method and, therefore, they have the same size for each of their dimensions. The image segments as well have intrinsically the same sizes for both dimensions while the sliding window method is applied to them. On the other hand, the sizes of the image segments which are obtained by using mean shift segmentation do not have to be equal [51].

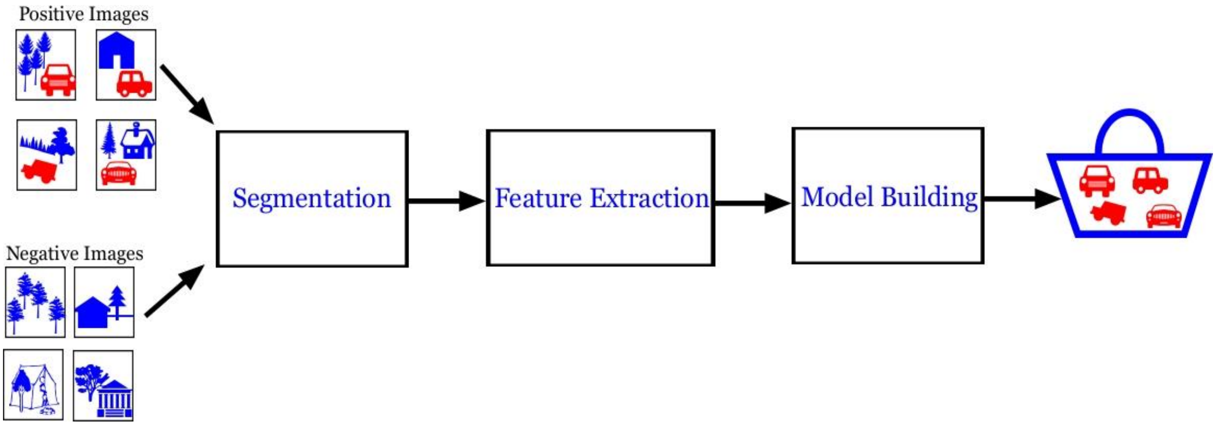


Figure 3.1: The workflow collecting a set of the parts belonging to the target visual object from a set of positive and negative image-level labeled color images.

Figure 3.1 shows the workflow, which takes a set of color images as inputs and returns a set of image segments. One should bear in mind that the input images are two different sets of images as positive and negative ones. The term "positive image" defines an image in which there exists at least one target visual object. In addition, the term "negative image" describes an image in which there exists no target visual object.

As presented in Figure 3.1, the processing step of segmentation takes two sets of images. Afterwards, the segmented images are sent to the processing step of feature extraction in order to extract discriminative feature descriptors for each segment. Ultimately, the processing step of model building creates a model, which has a specific number of image segments along with their ratings. The higher the rating is, the more probable it is that the segment will become a target visual object. Any species of objects being either alive or not can be chosen as a target visual object. The selected target visual object in Figure 3.1 is a vehicle.

Based on the visual attributes of a selected target visual object, single or multiple visual feature descriptors are determined for the feature extraction [12]. The locally selected visual feature descriptors have to be invariant as much as possible to the changes such as scaling, orientation, illumination, affine deformation, etc. while those extracted from non-corresponding segments must stay distinguishable [22] [12]. Single or multiple types of visual attributes such as color, texture, shape and spatial relationships [12] are extracted from each image segment in order to construct the respective visual feature descriptor [52].

While global visual representations map the content of the entire image, the regions obtained from a given image are mapped by local visual feature representations [12]. Since the target visual objects usually do not cover the entire image, either the image is uniformly partitioned, or the analogous regions of the image are clustered by a segmentation method [12].

These segmented images constitute the training dataset and are applied to the feature extraction, which is the essential processing step on the path to success of gathering the parts of the target visual object with the highest precision. In fact, selection of image segments as the most discriminative parts of a target visual object is a successor processing step of the feature extraction.

In the following subsections, the roles of these processing steps shown in Figure 3.1 are presented comprehensively. Furthermore, all the methods utilized in each processing step of the workflow shown in Figure 3.1 are described.

3.1 SEGMENTATION

The goal of the processing step of segmentation presented in this thesis is to partition a given color image into a set of regions. The only known information about the images used as inputs of this processing step of segmentation is their image-level labels. Consequently, the images are divided into two groups as positive and negative ones. There is no information about the location of target visual objects. In this processing step, one segmentation method is selected from two options investigated in this thesis. These two segmentation methods are sliding windows and mean shift segmentation [51]. In Subsection 3.1.1 and Subsection 3.1.2, these segmentation methods are described based on their implementation and performance. Moreover, the following subsections also specify the advantages and disadvantages of utilizing them while comparing them to each other.

3.1.1 SLIDING WINDOWS

The sliding windows are M pixels in width and N pixels in height. M and N are parameters, which are set by each experimental setup of this thesis. The method of sliding windows starts traversing from the pixel location (0, 0) of an input image. The vertical sampling step of the sliding window has the same size with its own height. Firstly, the sliding window moves in the vertical direction and when it reaches the boundary of the image, it backtracks to 0 pixel of the row, however, this time it moves a single step horizontally. The horizontal sampling step of the sliding window is also same size with its width. This traversing process is repeated until the sliding window reaches the lower right corner of the image. Thus, there is no overlap between all the patches obtained by this method. The patches presented in Figure 3.3 are some outputs of the segmentation while the sample image shown in Figure 3.2 is given to this processing step as an input.

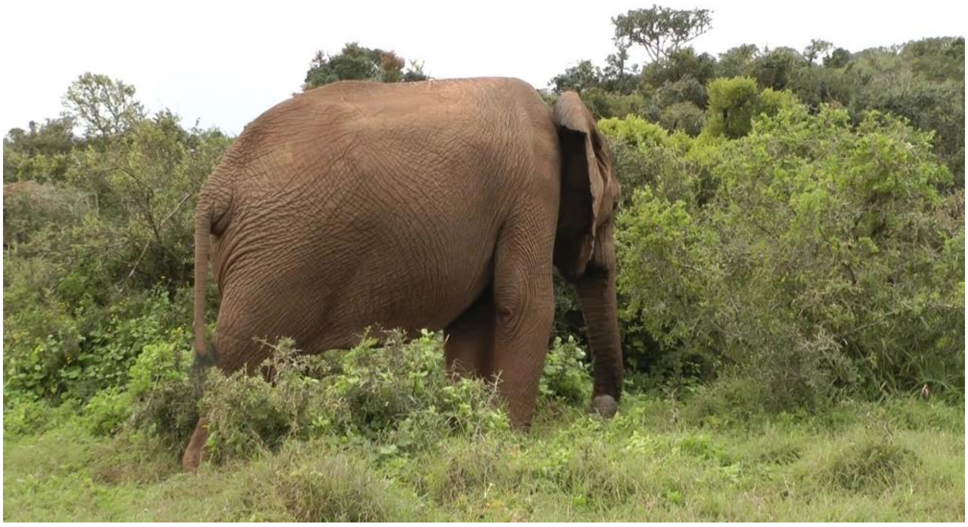


Figure 3.2: A sample image which is given to the processing step of segmentation as an input.

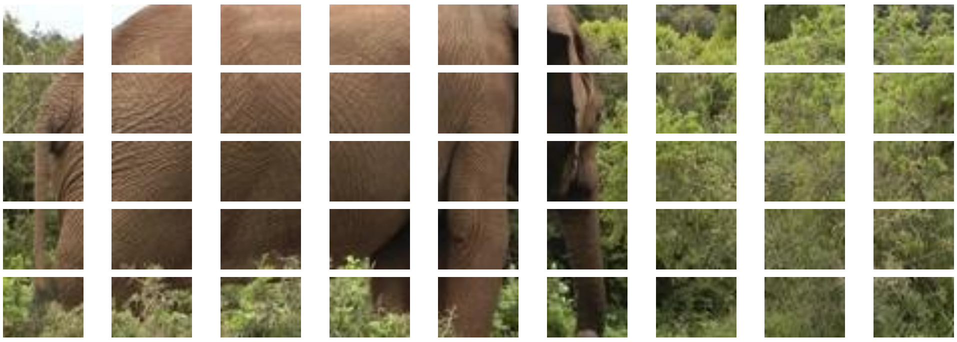


Figure 3.3: The patches collected by the sliding windows. While each vertical step size is equal to the height of each patch, the size of each horizontal step is equal to the width of each patch.

The smaller vertical and horizontal sampling steps are also investigated in this thesis. When the vertical and horizontal sampling steps are chosen as smaller than the height and width of the sliding window, respectively, some of the obtained patches partially overlap each other shown in Figure 3.4. In order to gather patches presented in Figure 3.4, the vertical sampling step is set two times smaller than the width of the window. Furthermore, the selected horizontal sampling step is also two times smaller than the height of the window. Accordingly, the number of generated patches is four times bigger in Figure 3.4 than while choosing width and height presented in Figure 3.3.

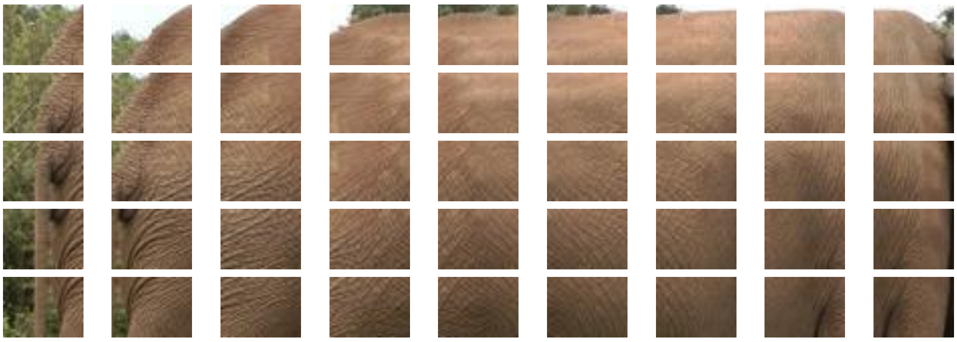


Figure 3.4: The patches collected by the sliding windows. While the size of each vertical step has a two times smaller height of each patch, each horizontal step size has a two times width of each patch.

There are two advantages of using sliding windows instead of mean shift in order to segment the images. The first one is that the implementation of sliding windows segmentation is easier than the one of mean shift segmentation. The second advantage of using it is having far shorter execution time than the one required for the mean shift segmentation. However, the disadvantage of the sliding windows is that a generated segment can contain not only the parts of target visual object, but also the background. Consequently, such segments act as a bottleneck for the following processing steps, which are feature extraction and model building.

3.1.2 MEAN SHIFT SEGMENTATION

The method of mean shift segmentation clusters the regions of sample color images relying on the algorithm of mean shift introduced by Fukunaga and Hostetler [53]. Its idea is to build a non-parametric clustering model for a sample of the data distribution gathered from color images [51]. Therefore, constructing a mean shift segmentation model from a sample is not imposed by a hidden premise. Mean shift segmentation partitions a sample color image by seeking the highest density values for the related local regions of the data distribution obtained as a result of the representation of the sample color image into L*u*v space [22]. These highest density values are called either modes or maxima [22]. Traversing the mean shift windows through the data distribution succeeds seeking the modes. The size of each mean shift window is parameterized. Firstly, the mean shift windows are located uniformly in the data distribution. Afterwards, each mean shift window is centered at the mean location of the data situated within the window. The procedure, which computes the mean location for each respective window and then relocates the window according to this mean location, is executed recursively until convergence. Finally, the windows ending up at the same peak location are merged and the data traversed by the same merged windows is specified as the same class.

The aim of utilizing the mean shift segmentation in this thesis is to partition sets of trainings and test images into regions containing strongly correlated pixels based on their color information. Furthermore, mean shift segmentation employed in this thesis is applicable to the color spaces of L*a*b and L*u*v individually although it is theoretically defined only in L*u*v. As it is explained in the Section 3.2, some investigated methods of feature extractions require segments with the format of rectangular boundaries as inputs. After the segments are gathered by mean shift segmentation, each of them is aligned as a rectangular image segment by filling the background determined based on the boundary of the respective segment with black pixels in case of the requirement by one of the investigated feature extraction techniques. For the purpose of this arrangement related to the background of each segment, 0s are assigned to all three channels of each background pixel location.

One should notice that the execution time for the alignment of rectangular segments with black background pixels is included in the whole execution time of mean shift segmentation. Therefore, the alignment of rectangular segments causes an additional processing burden for the processing step of segmentation with the use of the method of mean shift segmentation.

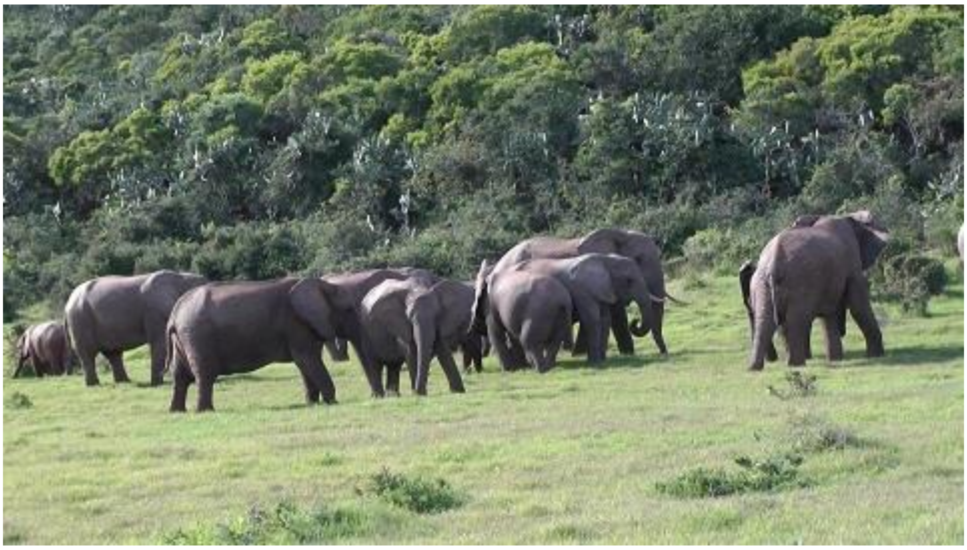


Figure 3.5: An instance from a set of positive color images. Since the elephant is selected as the target visual object, the image is positive.

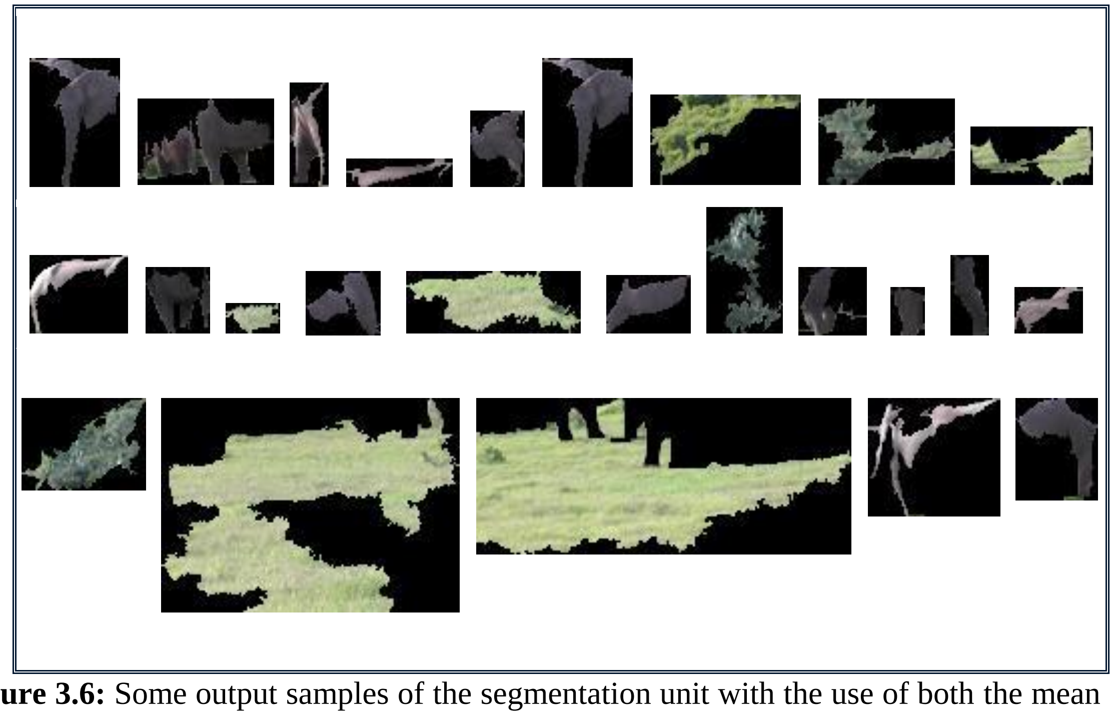


Figure 3.6: Some output samples of the segmentation unit with the use of both the mean shift segmentation method and the alignment of rectangular boundaries are illustrated.

Figure 3.5 shows a sample from positive training color images. Some examples of the segments obtained from the image shown in Figure 3.5 by applying mean shift segmentation are exhibited in Figure 3.6. Another potential drawback of the method of mean shift segmentation related to the required processing time is that all output segments do not have the same dimensions demonstrated in Figure 3.6. Accordingly, if the following section of either feature extraction or model building requires exact width and height sizes for the respective segments, then additional reshaping procedure has to be executed for these segments. Since the dimensions of the obtained segments by the method of sliding windows are exactly the same, such an additional processing load does not exist.

3.2 FEATURE EXTRACTION

Feature extraction is an operation of computing the values of certain features from a given sample [9]. Moreover, extracting feature values from a given sample always provides a data reduction [9]. While the process of feature extraction compresses a given sample, its discriminative data facilitating the recognition of a potentially situated target object has to be preserved and emphasized as much as possible [9].

The goal of feature extraction in this thesis is to obtain a compact and expressive feature descriptor from each image segment given as an input by the segmentation. The feature descriptors extracted from all the image segments are clustered or compared each other by a technique during the operation of model building. Therefore, the size of compressed data has a direct impact on the execution time of model building. Moreover, the accuracy percentage of the results generated by model building is unswervingly affected based on the quality of their distinguishability.

The different techniques analyzed in this thesis as the options of the feature extraction are presented in the following subsections. Apart from that, the strengths and weaknesses of these techniques based on each other are mentioned and compared.

3.2.1 COLOR HISTOGRAMS

Color histograms can be used as index vectors mapping the respective color images [54]. For this purpose, color histograms contain pre-defined number of bins for each color channel in order to compress color information gathered from a sample color image [55]. Each bin of the color histogram maps a specific range of quantities for the related color channel [55]. Therefore, the value of each bin points out the number of pixel locations mapping its specified range of quantities [55].

The drawback of the color histograms is the missing spatial information of compressed data [55]. In other words, after the images are mapped with color histograms, the spatial information of the pixels situated on these images is not preserved. Due to this drawback, Pass et al. [55] investigate some auxiliary computations to maintain the spatial coherence partially during the generation of color histograms. Figure 3.7 presents two color image instances, which have similar histograms based on the similarity measurement characterized by Pass et al. [55] due to red pixels depicting the flowers situated in the left image and the golfer’s t-shirt situated on the right image. Pass et al. [55] express that the values of bins specifying those red pixels are very close to each other in the color histograms extracted from these two-color images shown in Figure 3.7 although their contents are different. However, color histograms also have the advantages of being robust to scaling, rotation and translation. Furthermore, another advantage of this method is that it is widely used.

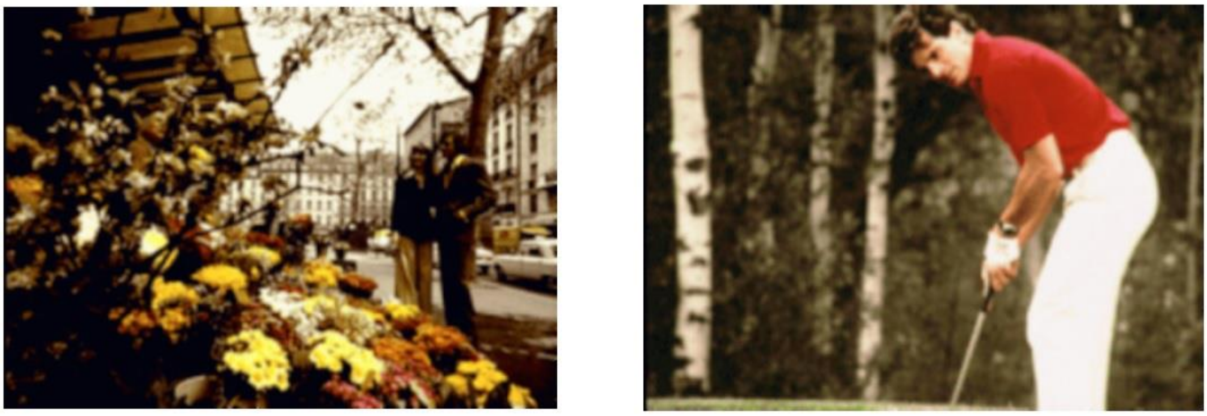


Figure 3.7: Two sample color images whose color histograms are recognized as similar based on the standards of similarity function defined by Pass et al. [55].

3.2.1.1 AB COLOR HISTOGRAMS

AB color histograms are constructed based on the color space of CIE L*a*b (CIELAB). CIELAB color space is derived from the color space of CIE 1931 XYZ in order to provide perceptual differences of color by utilizing the color distance function introduced by MacAdam [56]. In other words, CIELAB is a perceptually uniform color space because each identified three-dimensional color value in this color space has a unique visual perception in humans.

The opponent color theory defines aspects of encoding color process from retina to brain through the neural signals [57]. According to this encoding color process, some opponent colors occur in the visual interpretation of humans such as white against black, red against green and blue against yellow [57]. The representation of CIELAB color space is carried out in three-dimensions by these three components of opponent colors shown in Figure 3.8 because the opponent color theory is the foundation of CIELAB color space [57]. The dimension of perceived brightness (lightness) is named as $\mathbf{L}^*$. The brightness increases from bottom to top in $\mathbf{L}^*$ axis. Other two dimensions are designated as **a*** and **b*** presented in Figure 3.8. The positive values of the dimension **a*** state red while green is indicated by the negative values of the dimension **a***. The dimension **b*** is exploited to specify yellow with its positive values and blue with its negative values.

In this thesis, in order to gain illumination invariant features (extracted as color histograms) defined in CIELAB color space, the dimension of **L** is ignored. Therefore, a two-dimensional AB color histogram is extracted from each segment obtained from the mean shift segmentation described in Subsection 3.1.2. In the implementation of two-dimensional AB color histogram, 101 bins are deployed and the range for the bin centers is [-200, +200]. As the gathered segments from mean shift segmentation are in the color space of RGB, they are firstly converted into CIELAB color space. The extraction of AB color histogram does not require rectangular boundaries for the segments. The color information for each pixel location belonging to segments is executed for the construction of AB color histograms. Therefore, the background color information has no effect in this process.

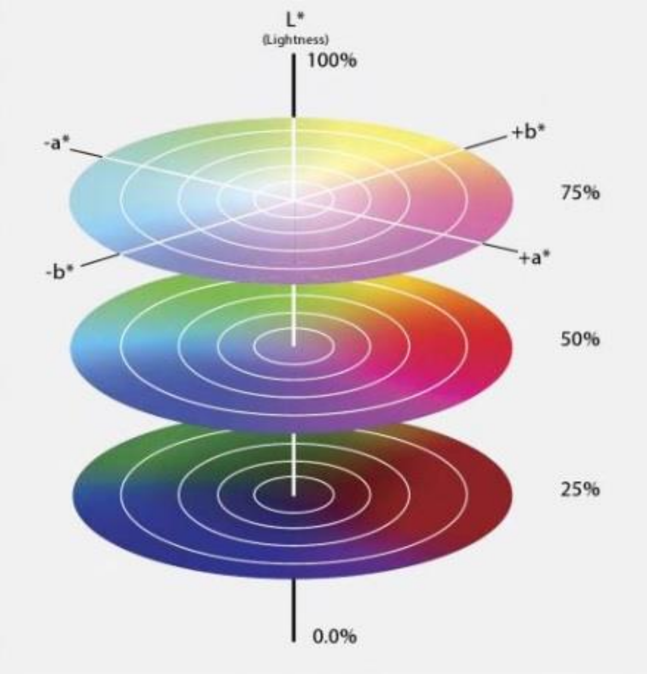
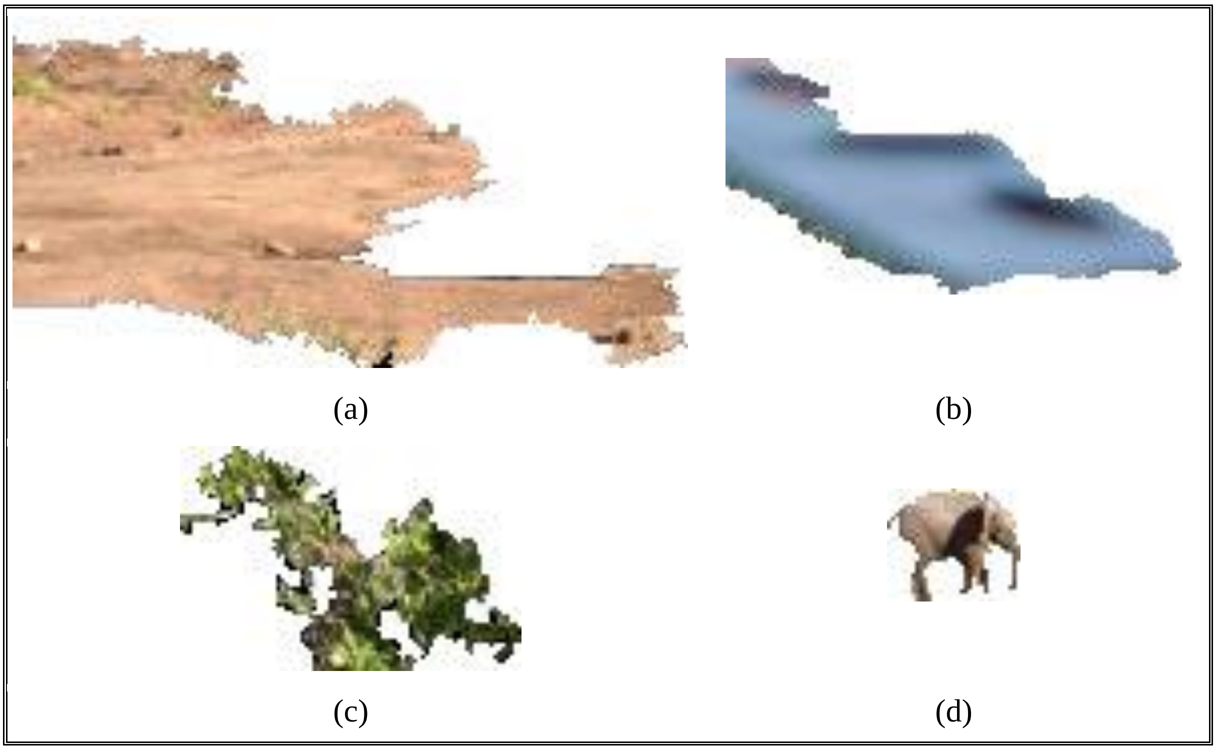


Figure 3.8: The visual representation of CIELAB color space, which is modelled with opponent coloring in 3-dimensions [58].



(a)

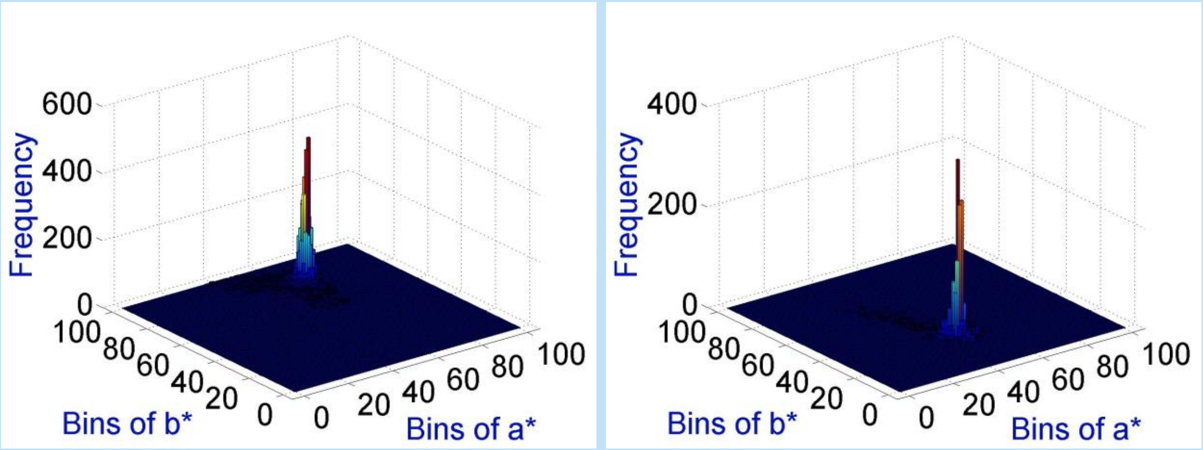
(b)

(c)

(d)

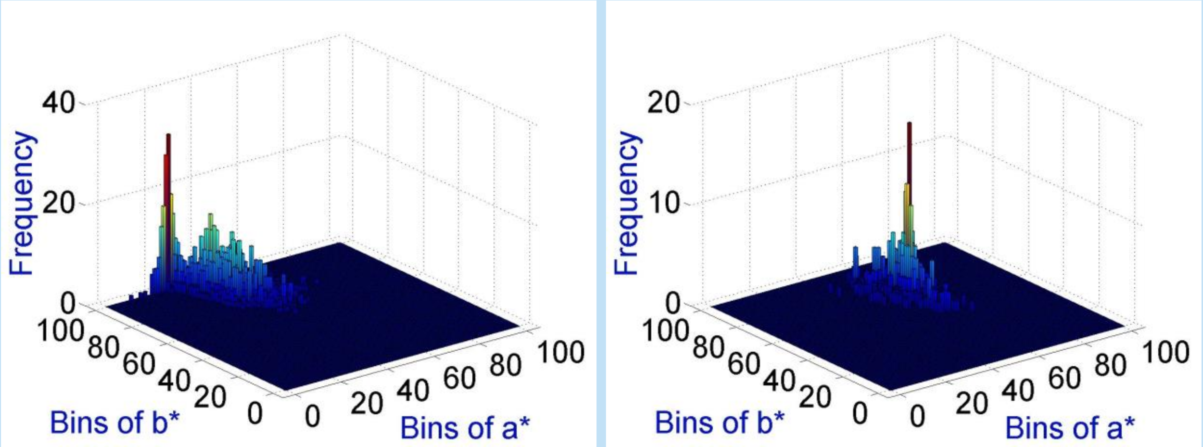
Figure 3.9: Four different segments obtained in the color space of CIE L*a*b by the application of mean shift segmentation.

The AB color histograms extracted from four different segment samples shown in Figure 3.9 are visualized in Figure 3.10. The frequency values of bins in AB color histograms shown in Figure 3.10 imply the range of color values concerning the dimensions of a* and b* existing in a set of discrete pixel locations for each segment. Hence, the bins keeping the highest frequency values of AB color histograms specify the dominant colors of the respective segments.



(a) AB color histogram obtained from the segment shown in (a) of Figure 3.9.

(b) AB color histogram obtained from the segment shown in (b) of Figure 3.9.



(c) AB color histogram obtained from the segment shown in (c) of Figure 3.9.

(d) AB color histogram obtained from the segment shown in (d) of Figure 3.9.

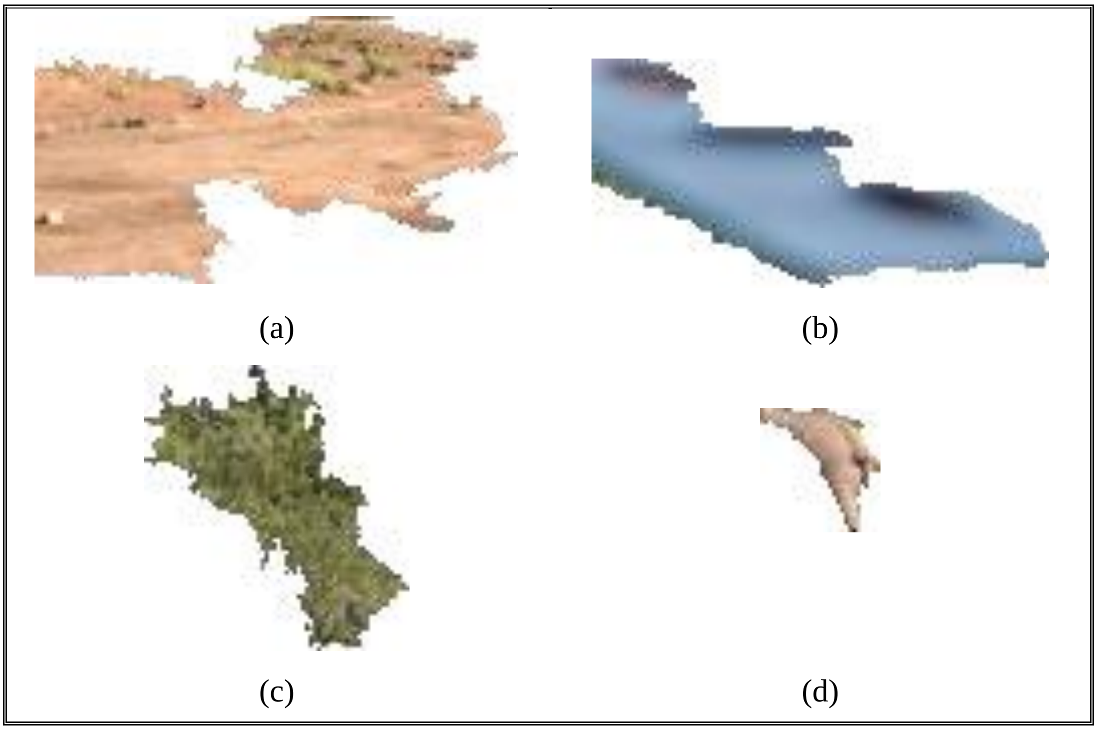
Figure 3.10: The exhibition of AB color histograms obtained from four different segments shown in Figure 3.9.

3.2.1.2 UV COLOR HISTOGRAMS

CIE 1976 L*u*v* (CIELUV) forms the base of the extraction of UV color histograms from the segments obtained by mean shift segmentation. CIELUV is also proposed as a perceptually uniform color space like the color space of CIELAB by International Commission on Illumination (CIE) [59]. CIELUV utilizes the upgraded version of CIE 1964 u, v diagram [59]. The fundamental design of both CIELUV and CIELAB is similar [59]. The intentions of introducing such similar color spaces are pragmatic and historical. Both color spaces of CIELUV and CIELAB rely on human perception instead of on that of any device; therefore, they are both device-independent color spaces [37].

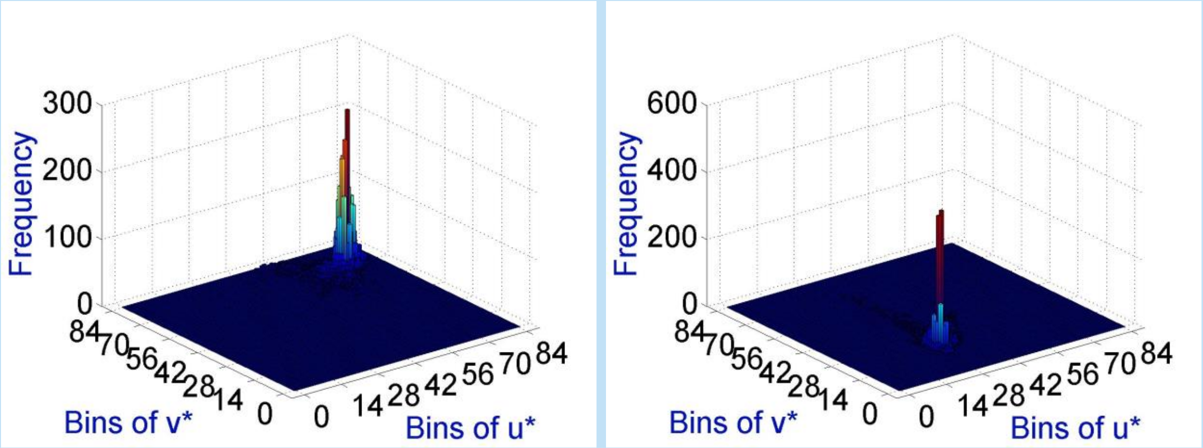
The procedure for the extraction of UV color histograms starts with the conversion of gathered segments from RGB color space into CIELUV color space. The range [-150, +150] is adapted for both dimensions **u*** and **v*** and 85 bin centers are located in the equal intervals between this range. Based on the values of **u*** and **v*** for each pixel situated in the regions of the segments, the value of the proper bin center is incremented by matching the proper bin center for the related color information. This process of finding the values of each bin center is exactly the same with the one of AB color histograms.

Figure 3.11 and Figure 3.12 present four different samples of segments and their corresponding UV color histograms, respectively.


(a) (b)

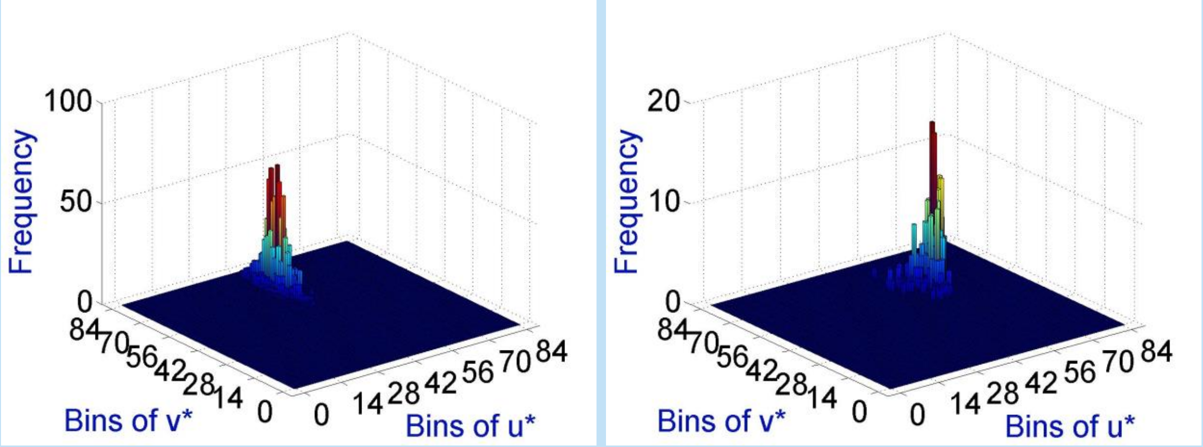
(c) (d)

Figure 3.11: Four different segments obtained in the color space of CIE L*u*v by the application of mean shift segmentation.



(a) UV color histogram obtained from the segment shown in (a) of Figure 3.11.

(b) UV color histogram obtained from the segment shown in (b) of Figure 3.11.



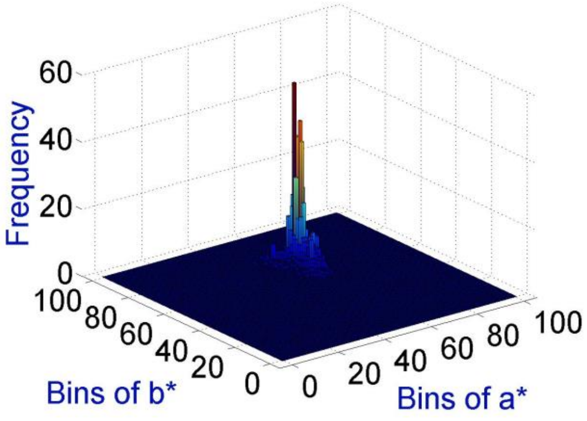
(c) UV color histogram obtained from the segment shown in (c) of Figure 3.11.

(d) UV color histogram obtained from the segment shown in (d) of Figure 3.11.

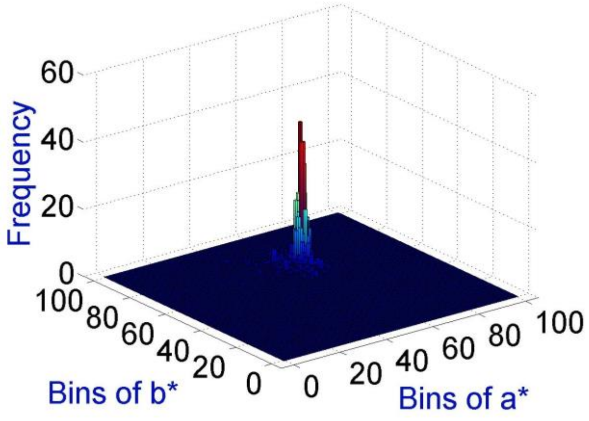
Figure 3.12: The exhibition of UV color histograms obtained from four different segments shown in Figure 3.11.

3.2.1.3 COMPARISON BETWEEN AB AND UV COLOR HISTOGRAMS

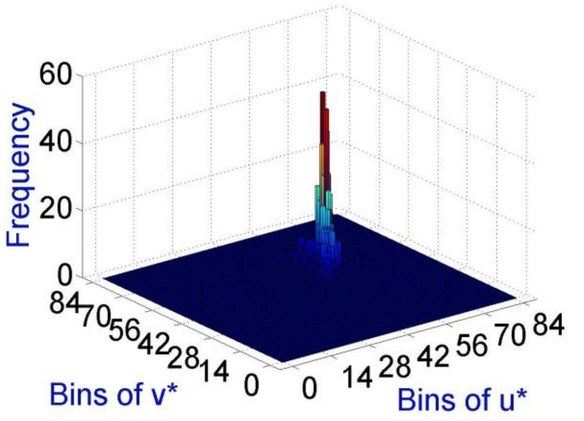
As it is specified in Section 3.1, the implementation of mean shift segmentation supplies the operation of the segmentation in two different color spaces and outputs the segments indicated in the color space of RGB. The UV color histograms presented in Figure 3.12 are collected by integrating the feature extraction of UV color histograms with the mean shift segmentation operated in CIELUV color space. On the other hand, the mean shift segments are obtained by utilizing the color space of CIELAB before the extraction of AB color histograms, which are demonstrated in Figure 3.10. However, Figure 3.13 shows AB and UV color histograms both extracted from mean shift segments gathered in the color space of CIELAB. In Figure 3.13, the upper histograms are AB color histograms and the lower histograms are UV color histograms. Additionally, the leftmost histograms are extracted from the same segment while another segment is used to extract the rightmost histograms.



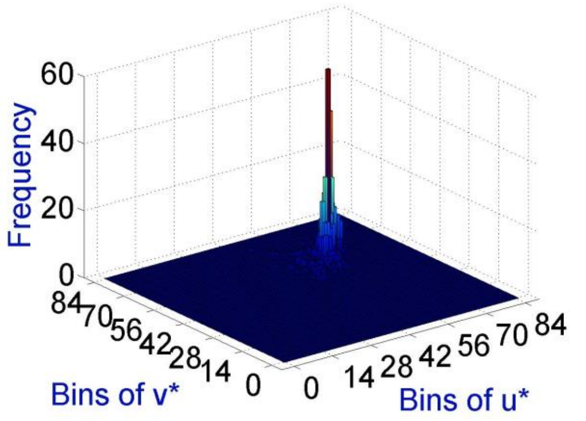
(a) AB color histogram obtained from the segment shown in (e).



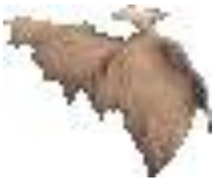
(b) AB color histogram obtained from the segment shown in (f).



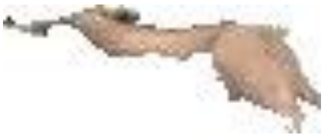
(c) UV color histogram obtained from the segment shown in (e).



(d) UV color histogram obtained from the segment shown in (f).



(e)



(f)

Figure 3.13: The AB and UV color histograms extracted from two different samples of the segments are depicted.

The segments shown in (e) and (f) of Figure 3.13 belongs to the visual object which is the species of elephants. The primary colors of these both segments are similar to each other. Hence, their extracted both UV and AB color histograms contain similar frequencies for their respective dimensions.

Both AB and UV color histograms presented in Figure 3.10, Figure 3.12 and Figure 3.13 are not normalized between 0 and 1. However, there is also an optional component in the processing step of feature extraction, namely that each extracted either AB or UV color histogram are normalized. Therefore, the indices obtained from either AB or UV color histograms referencing the respective segments are scale invariant features in this case of the optional computation flows. In addition to this case, these obtained indices are executed either with their multi-dimensional structure or firstly converted in the form of column vectors in the processing step of model building as another optional case. Furthermore, the final version of each column vector, which represents the corresponding image segment, is constructed in order to be used by the processing step of model building by applying PCA to the respective AB or UV color histogram. The difference of the reduction of dimensions between AB and UV color histograms is that the dimensions of the column vectors stating UV color histograms are reduced to 8 number of principal components while the selected number of principal components is 6 for AB color histograms. Since higher accuracies are obtained by performing l_1 normalization for both AB and UV color histograms instead of non-normalized ones during the experimental runs, l_1 normalization process is activated as default option for the construction of both types of color histograms. The demonstrated results in Chapter 5 regarding both AB and UV color histograms contain a performed l_1 normalization process, which divides each column vector gathered from either AB or UV color histogram with the summation of its values. As the extraction of UV color histograms is applied to the mean shift segments, the outer parts of the segments regarding the respective images do not have any effect for the computation flows of the proposed approaches as the same as the extraction of AB color histograms.

3.2.1.4 RGB COLOR HISTOGRAMS

The different algorithms are investigated in this thesis to construct color histograms, which are based on the color model of RGB, and are called as RGB color histograms. RGB color model denotes each color as the mixture of lights owning proper spectral ranges of primary colors, which are red, green and blue [60]. All the sets of colors indicated by RGB color model can be represented as a cube located in the origin of a Cartesian coordinate system [60]. Such a sample cube is shown in Figure 3.14 [61]. The corner dimensions of the cube indicate the pure colors of black and white and as well as the primary and the secondary colors [60]. The corner holding three-dimensional indicator of pure black color locates in the origin [60]. This black corner is adjacent with the corners holding the primary colors and pure white in a structure shown in Figure 3.14 [61]. Hence, the furthest adjacent of the corner located in the origin is the pure color of white. When the range of dimensions is normalized between 0 and 1 for the representation of RGB color cube, then it is called “unit cube” [60].

All the investigated algorithms, in order to extract RGB color histograms, take the mean shift segments as inputs. Therefore, the outward parts of the segments regarding the input images are excluded for the construction of respective RGB color histograms like the construction of both AB and UV color histograms. The operation of mean shift segmentation is carried out in CIELAB color space for these segments. The first algorithm constructs a two-dimensional histogram for each color channel of the input segment. The range is aligned as [0,255] for all of these two-dimensional histograms and allotted equally for 64 bins. Apart from that, each of these histograms is normalized. Once the construction of normalized two-dimensional histograms from all red, green and blue color channels of the segment instance is accomplished, they are concatenated. Hence, the indices referencing the respective segments are obtained from the compound of these histograms and each index has a form of a column vector with length of 192 (64*3).

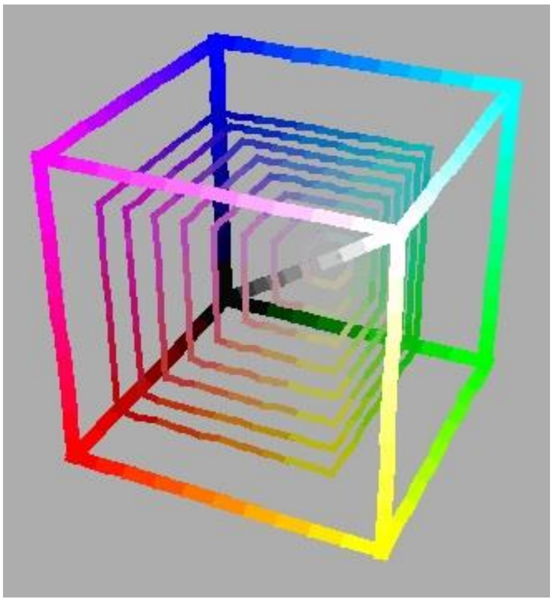


Figure 3.14: An instance of a cube representing the manifold of colors indicated by RGB color model [61]**.**

The second algorithm builds a three-dimensional RGB color histogram as a reference index for each respective input segment. The range for each dimension is adjusted between 1 and 256. In addition, 16 bins group each dimension equally. Each extracted three-dimensional RGB color histogram is normalized in the next step of the algorithm. Therefore, it supplies the property of scale invariance for the indices acquired from three-dimensional RGB color histograms like the ones constructed by the above-mentioned first algorithm constructing a type of RGB color histograms and the ones formed by exploiting both UV and AB color histograms as well.

The ultimate algorithm in the content of this thesis regarding RGB color histograms is based on the combination of two-dimensional histograms constructed by utilizing specified two-color channels of the input segments. The first two-dimensional histogram is built by using the channels of red and green while the channels of green and blue are used for the construction of the second two-dimensional histogram. Moreover, the final two-dimensional histogram embodies the color channels of blue and red. All the dimensions of these histograms are regulated in the range [0,255] and each dimension is classified with 32 bin centers. After the generation of these three histograms for each input segment, they are normalized, converted into single dimensional column vectors and chained sequentially. Finally, the dimensions of each of these joint column vectors are reduced to 12 principal components by performing PCA and these 12 principal components form an index for each respective segment.

3.2.1.5 GB COLOR HISTOGRAMS

GB color histograms are constructed based on only two-color components of RGB color model, which are green and blue components. GB color histograms are created only for the segments, which are gathered by mean shift segmentation processed either in CIELAB color space or

CIELUV color space. Another difference between GB color histogram and the rest of the aforementioned color histograms is the range and bin sizes of the histograms. The specified range of GB color histograms is [-100,100] for both dimensions and there are 51 numbers of bin centers located with equal intervals at each dimension. Same as the construction of the mapping indices for the respective image segments by making use of either UV or AB color histograms, l_1 normalization is performed on the column vectors stating the GB color histograms since the process of l_1 normalization increases the score of precision for each model built by one of these types of color histograms. However, any kind of additional techniques regarding dimensionality reduction of these normalized column vectors is not performed contrary to the column vectors constructed from AB, UV or a described type of RGB color histograms.

3.2.2 MPEG-7 COLOR DESCRIPTORS

Feature extraction techniques of two different MPEG-7 color descriptors employed in the thesis are described in this subsection. The rest of MPEG-7 color descriptors existing in the related area is also investigated during the experimental runs of the thesis; however, they are not involved with the scope of the thesis since their accuracies are worse than of the ones being in the scope of this thesis.

3.2.2.1 MPEG-7 COLOR STRUCTURE DESCRIPTOR

The fundamental design of MPEG-7 color structure descriptor is derived from the construction of blob histogram introduced by Qian et al. [62]. A blob histogram is constructed by adjusting the percentile of a structuring element with 1 and encapsulating the recurrences and sizes of each same type of local pixel aggregation based on textual and color characteristics of a given image [63]. MPEG-7 color structure descriptors are obtained from HMMD (Hue-Max-Min-Diff) color space [64]. The item of max defined in HMMD color space is computed as the maximum value placing in three channels of RGB color space [64]. Likewise, the minimum value of three RGB channels is set as min for HMMD color space [64]. The item of hue in HMMD color space is defined exactly same with the one defined in HSV color space [64]. The value of Diff can be computed by subtracting the value of Min from the value of Max [64]. In addition, the summation can be defined as an additional item and calculated by dividing the summation of the value of Min from the value of Max by 2 [64]. Hence, five items can be determined by using the color space of HMMD although the letters of its name contain only four items. Moreover, all of these items of HMMD color space are nonlinear and non-uniform quantization of this color space is employed for the extraction of MPEG-7 color structure descriptors [64].

Initially constructed blob histogram is 256-bin histogram and if the length of the MPEG-7 color structure descriptor is required to be 128, 64 or 32, then the length of bins is reduced by application of bin unification [64]. Bin unification groups 256 bins into new length of bins equally and sums the values of bins placing in the same groups in order to compute the values of the regarding updated bins [64]. Therefore, while the length of bins is being reduced, the average of bin values increases in the updated histogram, which is constructed by applying bin unification to initial histogram. In this thesis, bin unification is employed in order to reduce the length of bins from 256 to 32. Since performing nonlinear quantization on amplitudes of MPEG-7 color structure descriptors increases the accuracies of object models built by these descriptors [63], this process is also executed. Figure 3.16 shows histograms constructed during the extraction process of MPEG-7 color structure descriptor from a sample image segment presented in Figure 3.15 as well as a nonlinear quantization function utilized for the extraction process. Similar to segments used for the extraction of CEDD, MPEG-7 color structure descriptors are also extracted from the segments, which are obtained with their rectangular boundaries from CIELUV color space by applying mean shift segmentation.

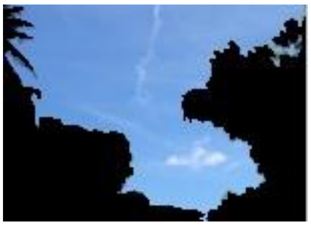


Figure 3.15: A sample image segment obtained by mean shift segmentation and extracted with rectangular boundary by setting background region with black color in order to be used for the extraction of MPEG-7 color structure descriptor.

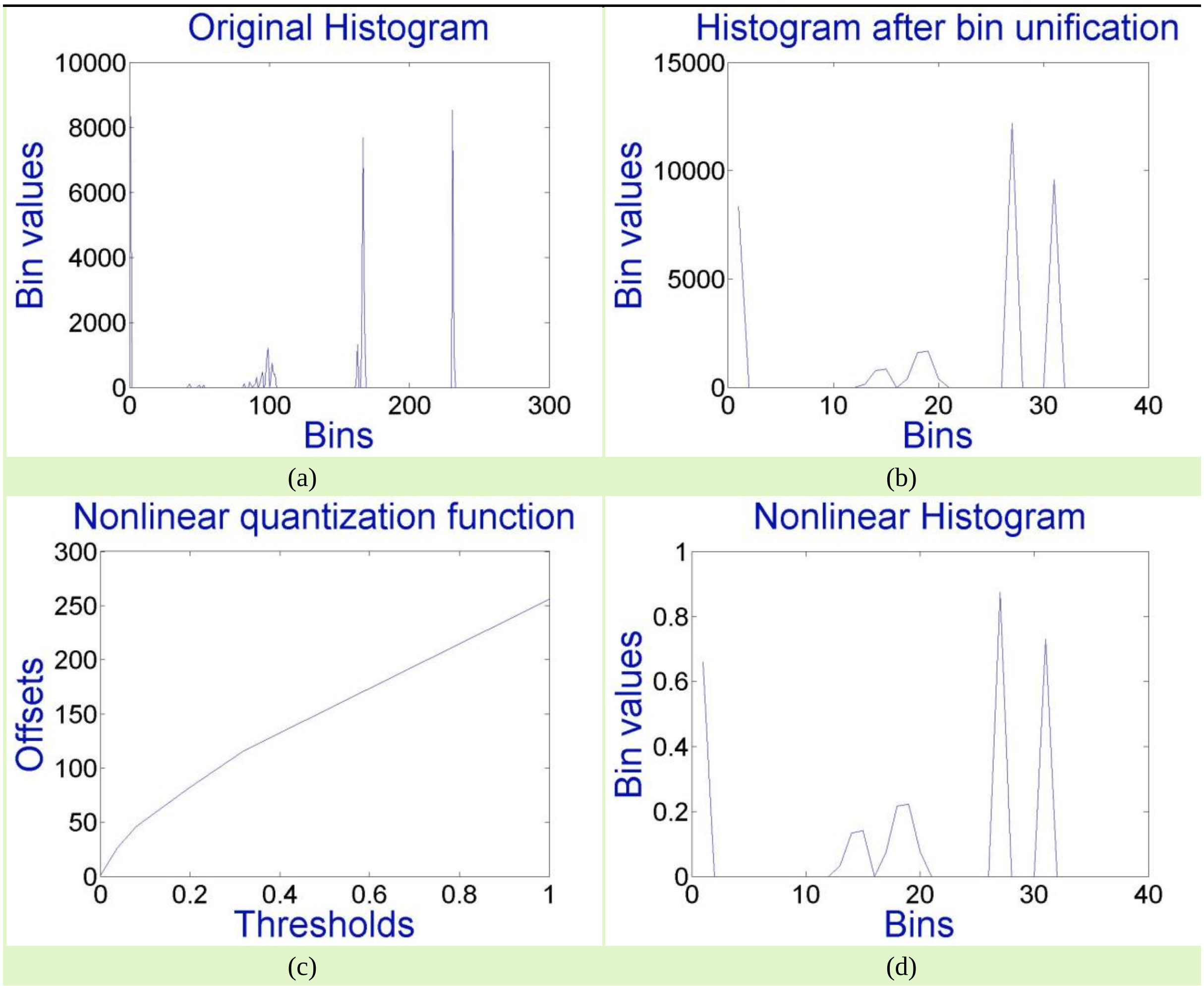
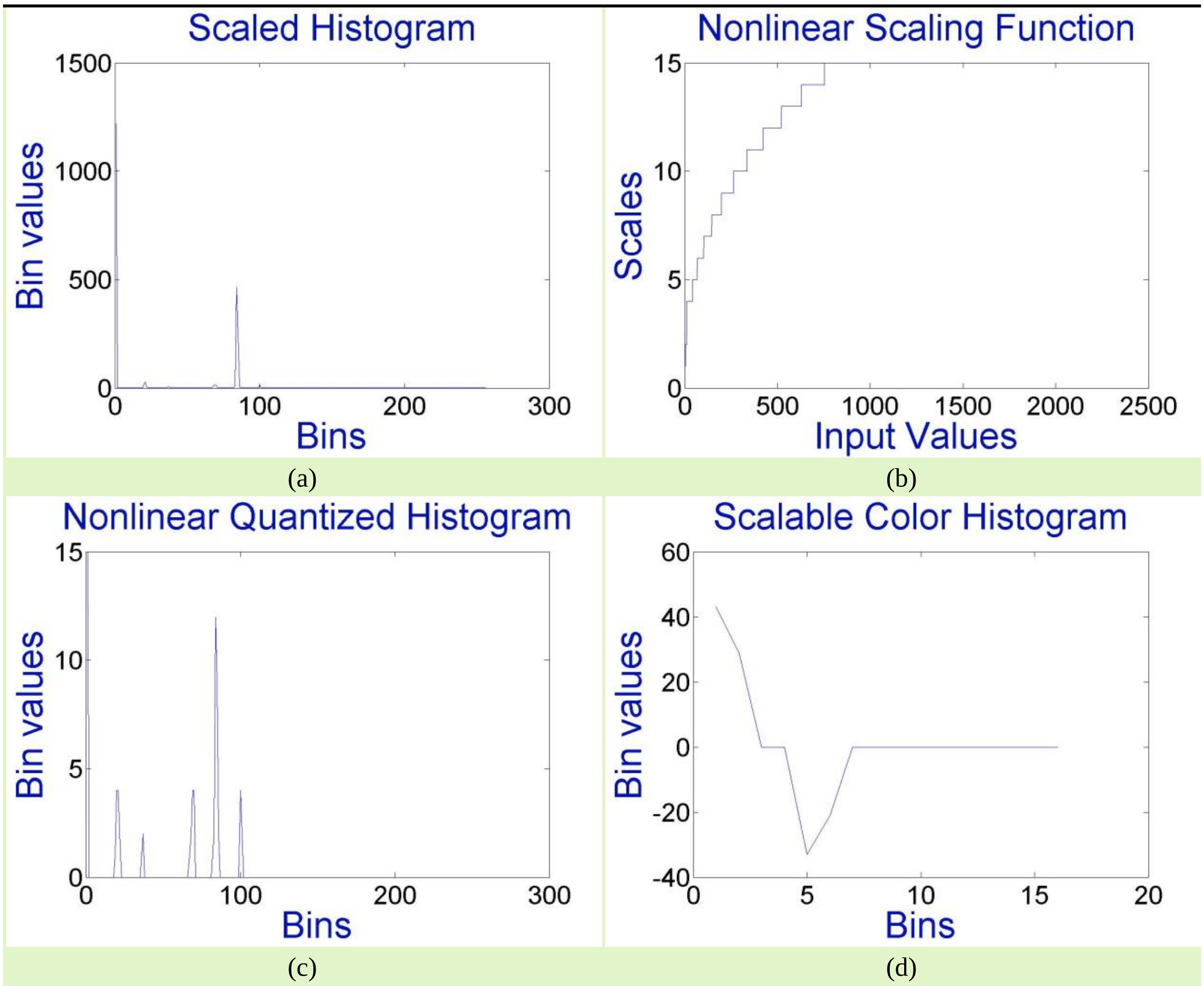


Figure 3.16: Histograms constructed from beginning to the end of extraction process of MPEG-7 color structure descriptor from the image segment shown in Figure 3.15 in addition to a performed nonlinear quantization function.

3.2.2.2 MPEG-7 SCALABLE COLOR DESCRIPTOR

Extraction of MPEG-7 scalable color descriptor from a given image is started by quantizing its color values placing in HSV color space into 256 bins uniformly [64]. Afterwards, this uniformly quantized 256-bin histogram is encoded by performing Haar transformation in order to reduce the length of bins from 256 to 128 in a single iteration [64]. In this thesis, the length of bins of the constructed uniformly quantized histogram is compressed from 256 to 16 bins by performing Haar transformation with 4 repetitions since reducing the dimensions to 16 bins causes an increase in the accuracy of object recovery using MPEG-7 scalable color descriptors. In order to extract MPEG-7 scalable color descriptor from a sample image segment presented in Figure 3.18, the histograms shown in Figure 3.17 are constructed one after another by applying the operations described above. Moreover, a nonlinear scaling function, which is employed to construct a nonlinear quantized histogram by taking a uniformly quantized 256-bin histogram as an input, is demonstrated in Figure 3.17. The same segments used by the extractions of MPEG-7 color structure descriptors are likewise used to perform extractions of MPEG-7 scalable color descriptors.



(a) (b)

(c) (d)

Figure 3.17: Histograms constructed from beginning to the end of extraction process of MPEG-7 scalable color descriptor from the image segments shown in Figure 3.18 in addition to a performed nonlinear scaling function.

Figure 3.18: A sample image segment obtained by mean shift segmentation and extracted with rectangular boundary by setting background region with black color in order to be used for the extraction of MPEG-7 scalable color descriptor.

3.2.3 COLOR AND EDGE DIRECTIVITY DESCRIPTOR (CEDD)

Color and edge directivity descriptor (CEDD) is proposed as an alternative feature descriptor containing only 54 bytes by Chatzichristofis et al. [65]. The size of 54 bytes for each feature descriptors corresponds to 144 numbers of dimensions. The main idea of the construction of CEDD for each given image is stated by Chatzichristofis et al. [65] as extracting a feature descriptor consisting of both color and texture information locating in the image and holding similar size of dimensions to MPEG-7 color descriptors. For this purpose, Chatzichristofis et al. [65] firstly extend the construction scheme of their previously introduced Fuzzy-Linking histogram in [66] from 10-bins histogram to 24-bins histogram by employing 4 more fuzzy rules in order to extract the color information from a given image formed in HSV color space. Chatzichristofis et al. [65] upgrade the limit calculations of each HSV channel value in [66] as distinct from [65] by detecting the vertical edges placing in these channels. In order to emphasize the vertical edges placing in a given image, coordinate logic filtering and the operation of AND are applied sequentially. Computing the differentiation between inputs and outputs of this vertical edge emphasizing procedure finds the positions of the vertical edges. Moreover, an extraction of texture information placing in a given image is accomplished by employing 5 digital filters presented by Park et al. [67] for their usage in MPEG-7 Edge Histogram descriptor. These filters enable to cluster image segments by assigning either a single or multiple texture classes. Consequently, corresponding 144-bin histograms are constructed for each given image by utilizing this texture clustering.

Extraction of CEDD is applied in this master’s thesis to image segments obtained with mean shift segmentation technique, which is performed in CIELUV color space. The minimum segmentation area is set as 200 pixels instead of 300 pixels different than other investigated feature extraction techniques taking inputs as segments obtained by application of mean shift segmentation in CIELUV color space since this change improves the accuracies of object recovery using CEDD as feature extraction technique. The obtained segments are extracted with their outer rectangular boundary and the color of regions between the boundaries and the segment is set with black color. The sample segments extracted by this method are shown in Figure 3.6. The dimension reduction is not performed for the extracted feature descriptors of CEDD from given image segments since dimension reduction decreases the accuracies of object recovery using CEDD.

3.2.4 BAG OF VISUAL WORDS (BOVW)

The bag-of-visual-words model is derived from the bag-of-words model. The bag-of-words model compresses the representation of the series of words [68]. To this end, only the frequencies based on the words appearing in a sample text are kept in a discrete format [68]. Therefore, a fixed vocabulary is utilized in order to obtain these frequencies [68]. In other words, the semantics and syntax of the compressed text are lost with this discrete representation obtained by bag-of-words [68]. For the purpose of deriving the bag-of-visual-words model from the bag-of-words model, the components of the bag-of-words model identified as the words lying in a set of texts are replaced with the local feature descriptors extracted from a set of images. Moreover, the bag-of-visual-words model supplies an additional clustering process during the construction of the vocabulary [69].

Csurka et al. [70] are the inventors of the bag-of-visual-words model but name this model as the bags of keypoints. The bag-of-visual-words also constructs a single dimensional histogram to provide a global representation for each instance (image or image segment). For this purpose, a clustering algorithm for all the gathered local feature descriptors from a set of training images is operated as a pre-learning process in order to build a visual vocabulary [71]. After a visual vocabulary is obtained from a selected set of training images, each extracted feature descriptor from a set of test instances can be mapped to one of the classes defined in this visual vocabulary [71]. The number of occurrences of these classes forms a single dimensional histogram regarding each instance as a global feature descriptor [71]. Since the bag-of-visual-words model requires the use of a clustering algorithm for the discrete data denoted as the input, the bag-of-visual-words model is a vector quantization technique [71].

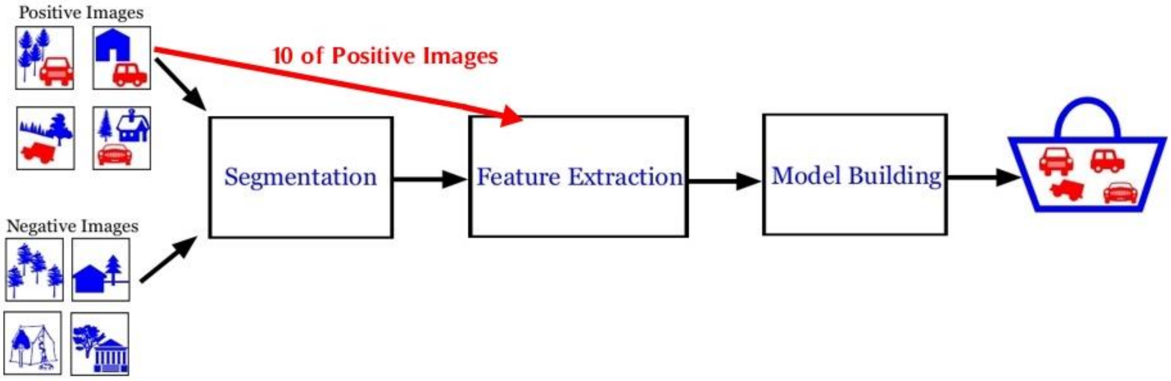


Figure 3.19: The illustration of the updated design of Figure 3.1 with the additional usage of 10 positive images selected from the initial positive image set in order to construct a visual vocabulary in the processing step of feature extraction.

K-means clustering is applied in this thesis in order to construct a visual vocabulary from a set of local feature descriptors extracted from a selected training dataset in case the bag-of-visual-words model is employed as a feature extraction tool in a research path involved in the thesis. In order to extend the experimental setups used in this thesis, the different values of k are selected as the additional research options in the tool of k-means clustering such as 50, 100, 150, 200 and 250. The definition of k-means clustering is given in Subsection 2.1.1. One should note that 10 positive images are selected as a training dataset from 30 positive training images used in the processing step of segmentation in order to be utilized to build a visual vocabulary. In other words, the same 10 training images exploited for the building of a visual vocabulary are also used as the inputs of the segmentation included in the full execution path, which produces the parts of the target visual objects shown in Figure 3.19. Therefore, the experimental setup including the bag-of-visual-words model in the processing step of feature extraction updates the design shown in Figure 3.1 as Figure 3.19 presented above.

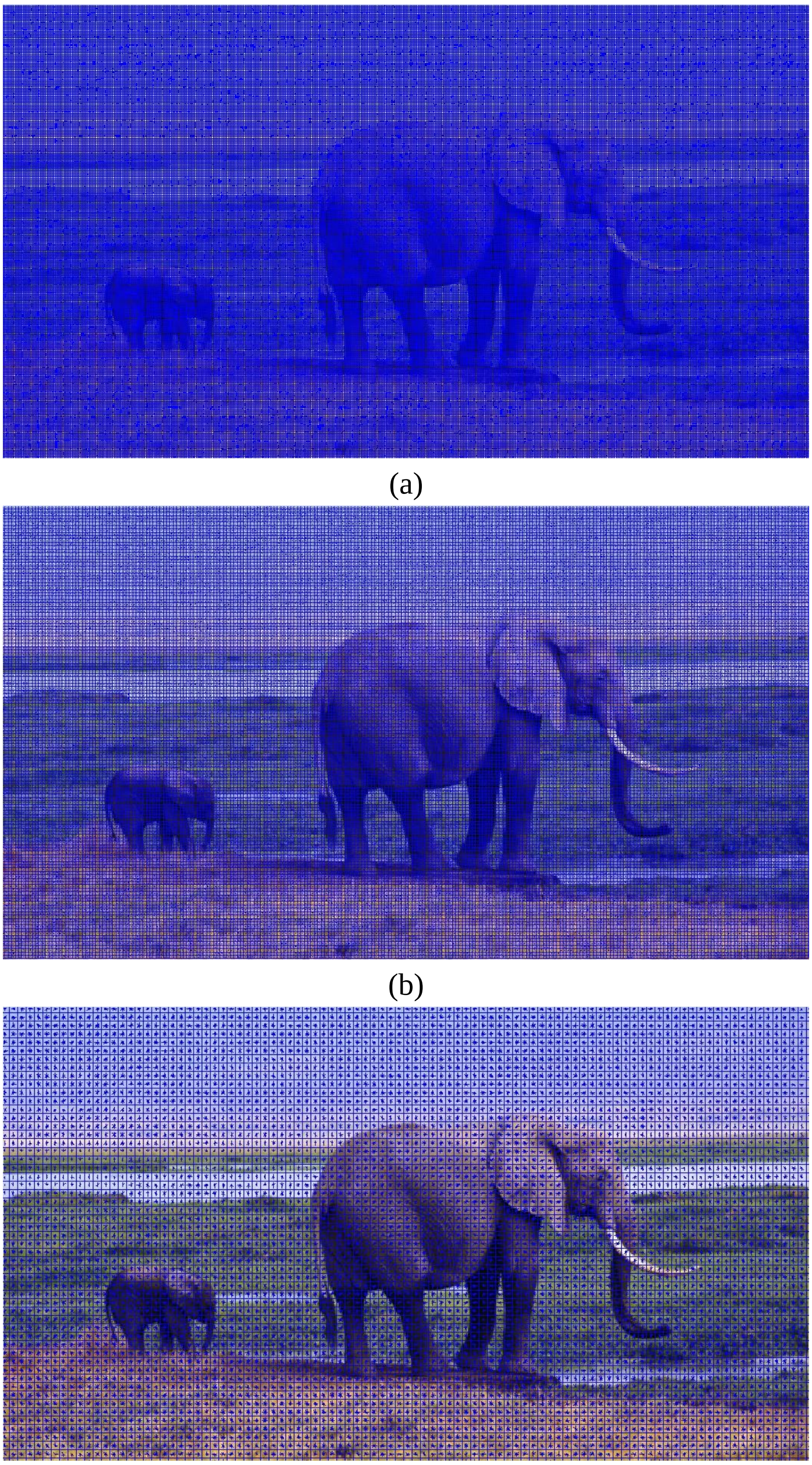
3.2.4.1 DENSE SIFT AND PHOW DESCRIPTORS

Bosch et al. [69] establish the verification of the higher accuracies for the classification of the visual objects with the supplied experimental evaluations by extracting the SIFT descriptors from the regular dense grid points rather than the sparse interest points. Moreover, Bosch et al. [69] name the descriptors obtained by this method as dense SIFT descriptors. The root cause of the improvements in the experimental results is that the SIFT descriptors computed over the dense grid points provide more information about the contents of the respective images due to the larger set of the extracted local feature descriptors rather than the SIFT descriptors computed at sparse interest points.

In order to construct a visual vocabulary when the dense SIFT descriptor is selected as the feature descriptor for one of the research purposes of this thesis, the descriptors are extracted from the entire image frame of each selected positive training image by traversing the Gaussian window. The size of each sampling step with respect to the Gaussian window is set as an optional case, which is single or 2 pixels, and the research results are collected for both cases. Apart from these two cases, the sampling at each 5 pixels is likewise investigated for the extraction of VLAD descriptors described in the following subsection. Moreover, whichever case is selected for the sampling step size, the size of each spatial bin locating in the Gaussian window is aligned as 2 pixels. One should bear in mind that before the computation of dense SIFT descriptors, the input images are converted from RGB color images to greyscale images. Figure 3.20 demonstrates the computation of SIFT descriptors at the dense grid points located by traversing the Gaussian window along one of the selected positive images. One should note that the bin borders and the keypoints of the Gaussian window is colored with blue for the demonstration purpose. Figure 3.20 (a) is the case when the sampling step size is determined as a single pixel. Additionally, Figure 3.20 (b) and Figure 3.20 (c) respectively present the illustration conditions for the selection of sampling step size as 2 and 5 pixels in order to settle the grid points for the computation of the SIFT descriptors by utilizing the Gaussian window.

The size of each dense SIFT descriptor is 1×128. After the construction of a visual vocabulary by applying K-means clustering to these extracted dense SIFT descriptors, the dense SIFT descriptors are extracted this time from the whole training data. The newly extracted dense SIFT descriptors are assigned to the respective segments based on their fellowships with the regions of the segments. Eventually, single dimensional histograms indexing these segments are built by assigning these extracted dense SIFT descriptors to their closest visual words defined in the visual vocabulary constructed during the previous computation step and counting the number of the occurrences of the visual words for each segment.

The magnified demonstration of the extraction of dense SIFT descriptors from an instance of the image segments is presented below in Figure 3.21. The only descriptors, which are in a relation with the region of the considering image segment, are taken into account in order to build a single dimensional histogram indexing this image segment. Performing l_1 normalization on the histograms constructed using dense SIFT descriptors based on bag of visual words is also investigated as well as non-normalized ones.



(a)

(b)

(c)

Figure 3.20: The depiction of dense SIFT computation over one of the selected positive training images with different sampling step sizes (a single pixel (a) or 2 pixels (b) or 5 pixels (c)).

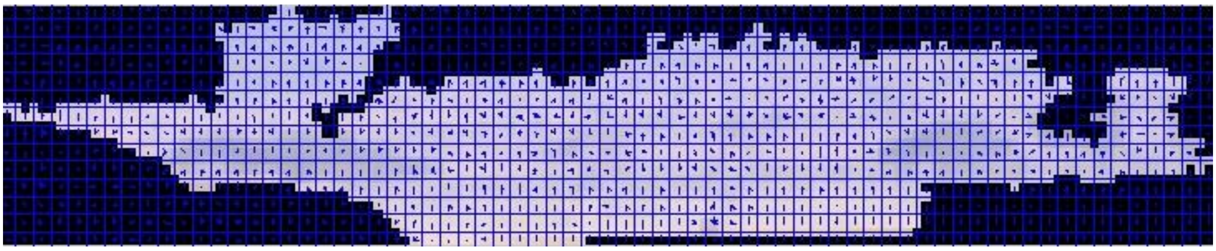


Figure 3.21: The enlarged representation of the gathering dense SIFT descriptors from a segment sample.

Bosch et al. [72] introduce Pyramid Histogram of Visual Words (PHOW) by pursuing and extending the computation method of the SIFT descriptors at the dense grid points proposed again by Bosch et al. [69]. According to this newer approach derived from the dense SIFT descriptor, the extraction of SIFT descriptors is performed for four times at each grid point and the sizes of the annular areas employed for the computation of the SIFT descriptors differ from each other at the same grid point. The value of the radius is set with one of the followings pixels presented in the paper introducing PHOW [72] to impose the annular areas possessing different pixel sizes on the extraction of the SIFT descriptors: 4, 8, 12 or 16 pixels. Apart from this additional extraction possibility of the SIFT descriptors obtained densely with 4 different scales instead of only a single scale, the PHOW descriptors are applicable to the color images in addition to the greyscale images [72]. Bosch et al. indicate in their paper [72] that the dense SIFT descriptors are extracted with the specified scales for each channel of the input color image defined in the HSV color model. RGB color space forms HSV color space by projecting the color values denoted in the RGB cube onto the HSV cylinder and this projection enables a system using HSV color space to gain human perceptual differentiation [73]. Afterwards, the descriptors which are extracted from the different channels with the same scales and at the same grid points are jointed as PHOW descriptors. Therefore, the size of each PHOW descriptor is denoted as 128×3.

To obtain PHOW descriptors from the selected color images, RGB is used instead of HSV for the SIFT computation. Furthermore, the gathered dense SIFT descriptors from the different color channels are concatenated if they are extracted at the same grid points and with the same scales. Hence, the size of the extracted each PHOW descriptor is 1×384. Moreover, 4 different scaling operations similar to the above-mentioned experimental runs executed in the related original paper [72] with the following spatial bin sizes are used: 4, 6, 8 and 10 bins.

The extraction process of PHOW descriptor is succeeded by traversing a Gaussian window along the images same as dense SIFT descriptors described in the previous pages. Thus, the Gaussian window is also aligned with the spatial bins and each of the spatial bin keeping the size of 2 pixels for the computation of the PHOW descriptors. Apart from that, the same sampling sizes which are investigated during the computation of dense SIFT descriptors, are also analyzed during the extraction of PHOW descriptors. However, the images are not converted from RGB color images to greyscale images. Instead, the extraction process of PHOW descriptors is operated directly to the RGB color images. Since more descriptors are extracted due to the additional computations with respect to the scaling and the color information, the PHOW descriptors are more powerful scale invariant features and contain more clues placing in the color information rather than the dense descriptors. Based on the experimental checks, a normalization process decreases the score of precision of each model built by the histograms constructed using PHOW descriptors. Therefore, these histograms are sent as representing indices for the respective image segments to the model building without processing normalization on PHOW descriptors. The regions of image segments are determined by performing mean shift segmentation based on CIELAB color space.

3.2.4.2 VECTOR OF LOCALLY AGGREGATED DESCRIPTORS ENCODING (VLAD)

Jégou et al. [74] present a new generation of image mining technique, which is called Vector of Locally Aggregated Descriptors Encoding (VLAD). This new approach is constructed by being inspired to find a simplified image mining technique, which keeps the structure of both Fisher vector and the bag-of-visual-words model [74]. The description of the Fisher vector is given in Subsection 2.3.1. The first advantage of VLAD against bag-of-visual-words model is that VLAD keeps the more detailed information with respect to the distinction of the classified local descriptors. In other words, the bag-of-visual-words model assigns the local descriptors to the closest visual words placing in the visual vocabulary; however, VLAD stores the distances between the local descriptors and their assigned closest visual words in addition to the assignment [75]. The second advantage of VLAD against bag-of-visual-words model is that the dimensions of the whole encoded data obtained by operating the technique of VLAD can be reduced dramatically without decreasing the quality of distinguishability by utilizing PCA [74]. The explanation regarding the method of PCA is provided in Subsection 2.3.1. These advantages are perceived as evidences to supply high-quality products with lower costs and decreased computation time by the industrial point of view. In this thesis, the number of visual words is determined as 16 to gather VLAD descriptors from given a set of image segments, which is collected by employing mean shift segmentation technique based on CIELUV color space. Same as the construction visual vocabularies for both dense SIFT and PHOW descriptors, 10 positively image-level labeled training images are selected. These selected training images are used to detect 16 numbers of cluster centers as visual words in a visual vocabulary. The dimension of each extracted VLAD descriptor is reduced from 6144 to 12 by applying PCA. Afterwards, the obtained principal components are processed by l_2 and power normalization. The idea of performing l_2 and power normalization during the construction of VLAD descriptor increases the precision scores of the object models constructed by using VLAD descriptor. These constructed final descriptors are sent to the processing step of model building.

3.3 MODEL BUILDING

After all the global feature descriptors are extracted from the corresponding segments by applying one of the feature extraction techniques described in Section 3.2, a non-parametric discriminative learning scheme on the basis of information gain proposed by Hao et al. [23] is adapted in the thesis. The elaborated description of this learning strategy and its usage in [23] is provided in Subsection 2.2.1. For this purpose, first of all, the similarity measurement for each pair of feature descriptors obtained from the segments of the complete set of training data is computed.

3.3.1 SIMILARITY MEASUREMENT

In order to extend the pathways of the research, eight different similarity functions apart from CK-1 distance function introduced in [28] shown as below list are involved in the scope of the investigation.

- Diffusion distance
- Euclidean (l_2) distance
- Jensen-Shannon divergence
- Kolmogorov-Smirnov test
- Kullback–Leibler divergence
- Chi square statistics
- Histogram intersection
- Matching distance

The definitions of these distance functions and their connections are given as follows. Diffusion distance computation between histogram pairs is an imitation of the natural diffusion flows through inner spaces of histograms by reducing potential bin-to-bin conflicts [76]. Euclidean distance between a pair of histograms is calculated by simply summing up each bin-to-bin Euclidean distance measurement [77]. One of differences between Euclidean and diffusion distance functions is that diffusion distance function has to compute most intensive flows only through inner spaces of a pair of histograms while the computation path of Euclidean distance function can contain both inner and outer spaces of this pair of histograms. Therefore, Euclidean distance function does not embed the adjacent bins into the computation of the measurement based on the input bins [77]. A distance quantity between empirical distribution functions of two instances is measured by Kolmogorov-Smirnov test [78]. Jensen-Shannon divergence measures a distance quantity between two probability distributions by utilizing Shannon entropy while Kullback–Leibler divergence computes a non-symmetric distance measurement (relative entropy) between a pair of probability distributions [79] [80]. Chi square statistics measures the similarity of the histogram pairs by calculating sum of the squared of difference (error) of histogram bins corresponding to standard normally distributed variables [81]. In addition, histogram intersection firstly sums up minimum distances between the histogram bins and then normalize these summations. Therefore, histogram intersection computes likewise the similarity measurement by involving the adjacent bins counter to Euclidean distance function [82]. Moreover, matching distance enables the combinatorial execution of the similarity measurement between histograms pairs by using size functions [83].

3.3.2 APPLICATION OF CK-1 DISTANCE

All the elaborated description of CK-1 distance is provided both in Subsection 2.2.1 and Section 6.1. CK-1 distance introduced in [28] is used for the experimental runs of this thesis without any changes in its implementation. The only difference is that the input segments are greyscale images in [23], while input segments are color images in this thesis. The input segments are gathered by either sliding window or mean shift segmentation (CIELUV color space is used to perform mean shift segmentation) techniques as inputs for CK-1 distance. Since CK-1 distance function is applicable to only a pair of images which have exactly the same shape, additional reshaping process is implemented to be executed for a pair of segments obtained by mean shift segmentation if they do not have same shape. If only widths or heights of segment pairs are different, then the smaller heights or widths are extended to the lengths of the bigger heights or widths with the addition of black pixels by this reshaping process. If both widths and heights of segment pairs are different, the differences between widths and heights are compared. If the difference is bigger for widths than heights, then the segment containing the bigger width is reshaped based on the smaller width by keeping the aspect ratio. Likewise, the segment which has bigger height than the compared one is reshaped based on the smaller height by keeping the aspect ratio if the difference is bigger for heights than widths. After the reshaping by keeping aspect ratio, if the lengths of cross sides do not match, shorter cross sides are extended to longer cross sides by adding black pixels. Moreover, smaller length for either width or height is 15 pixels to be applicable for CK-1 distance. Therefore, there is an additional alignment in reshaping process to avoid reshaping widths or heights smaller than 15 pixels. This additional alignment adds black pixels to extend the lengths of either widths or heights to 15 pixels if one of them has a length smaller than 15 pixels before or after the reshaping operation described above.

The major difference between all the above-mentioned distance functions and CK-1 distance function regarding their applications is that CK-1 distance function directly measures similarity between image segments while other distance functions listed in the scope of the thesis distribute the quantities of similarity comparisons between feature descriptors instead of image segments. Consequently, each kind of feature extraction techniques presented in Section 3.2 is required to be applied to image segments to compute their similarity measurements by using their corresponding feature descriptors if one of the investigated distance functions apart from CK-1 distance function is selected for the similarity comparison.

3.3.3 APPLICATION OF INFORMATION GAIN

In order to build a discriminative learning scheme on the basis of information gain, the distance measures between each segment from positively image-level labeled training images (set as a candidate segment) and the rest of the segments from both positively and negatively image-level labeled training images are computed by employing either one of eight similarity functions or CK-1 distance function. Afterwards, the values of information gain and threshold are calculated for each of these candidate segments by using these computed distance measures. The segments are sorted based on their values of information gain in a descending order. The highest value of information gain provides an intuition for the discrimination between positive and negative segments. Hence, object models are built with the numbers of 250, 200, 150, 100, 50 and 25 segments having the highest values of information gain by storing their values of information gain and thresholds as well as their indices, which show their locations in the respective training images. For instance, the first 25 segments locating in that sorted list are collected with their above-mentioned data to build an object model with the 25 best fingerprints. These selected segments for the recovery of object models are called fingerprints in this thesis like in [23].

Once object models are built with the 250, 200, 150, 100, 50 and 25 best fingerprints, the precisions of these models are calculated by matching gathered fingerprints with the ground truth data. If the intersection of a gathered fingerprint with the ground truth data contains minimum 80% of entire region of the fingerprint, then this gathered fingerprint is set as true positive; otherwise, set as false positive. Figure 3.22 illustrates the precision calculations for the recovered object models with the 250, 200, 150, 100 and 50 best fingerprints. Based on the precision results of recovered object models, optimal models can be determined. The segments of images showing the results of information gain are colored in a greyscale form from lightest grey color to darkest grey color based on the values of their information gain. The value of information gain varies between 0 and 1. For instance, if the information gain value of a segment is 1 then the respective segment is colored by setting the respective pixels with 0s. On the other hand, a segment holding 0 value of information gain is colored by setting its pixels with 255s. Likewise, the mid values between 0 and 1 regarding information gain supply percentages to determine pixel values between 0 and 255. Moreover, the images presenting the best fingerprints show the gathered fingerprints for the respective images based on the selected number of the best fingerprints for the recovered object models.

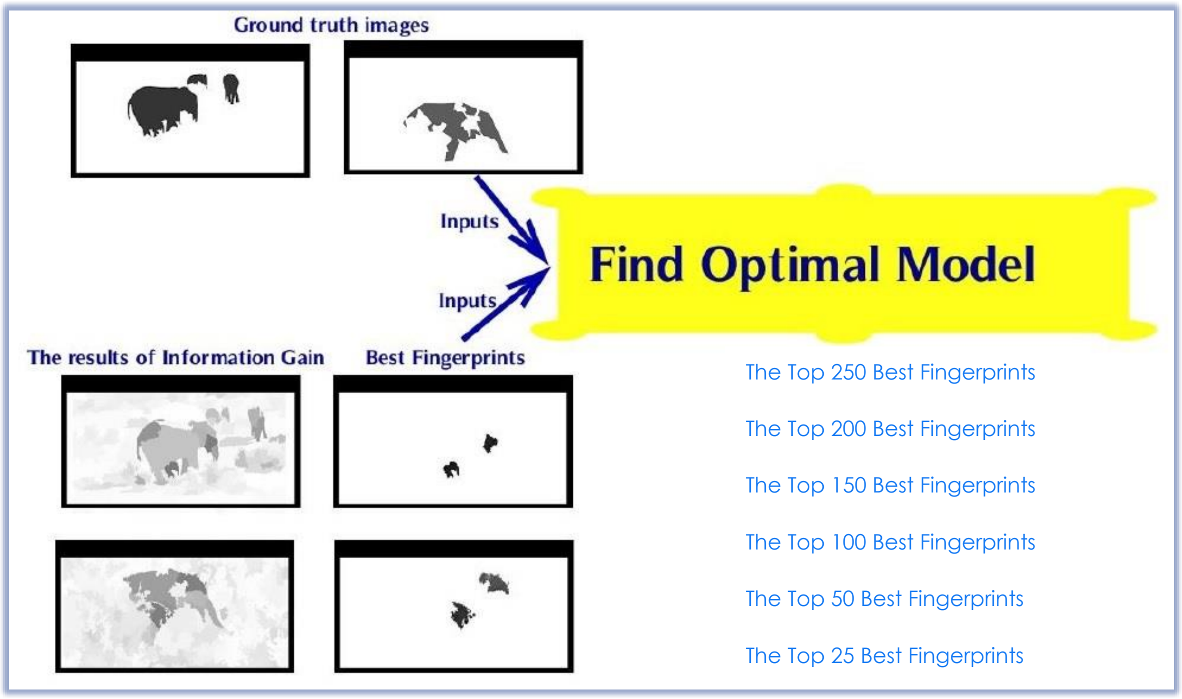

Ground truth images
Inputs
Inputs
Find Optimal Model
The results of Information Gain
Best Fingerprints
The Top 250 Best Fingerprints
The Top 200 Best Fingerprints
The Top 150 Best Fingerprints
The Top 100 Best Fingerprints
The Top 50 Best Fingerprints
The Top 25 Best Fingerprints

Figure 3.22: Demonstration of precision calculations for the recovered object models in order to select optimal ones.

3.4 DETECTION OF TARGET VISUAL OBJECTS

This section describes three different implemented approaches to detect segments of target visual object from a given set of positively image-level labeled test images by using one of the determined optimal object model. A Naïve Bayes classifier is employed as the first approach of target segment detector. Naïve Bayes classifier is the combination of a probabilistic model which utilizes the Bayes' Theorem under the assumption of the independency of feature sets

[84]. The fundamental of Bayes' Theorem is to compute the posterior probabilities by using prior probabilities and class-conditional densities [9]. In this thesis, the positive samples required for the training of this Naïve Bayes classifier are obtained from a selected object model. Furthermore, the negative samples used for its training are negative image segments, which are randomly collected during the segmentation process of object model recovery. The number of negative samples is aligned based on the number of positive samples because the training of this classifier utilizes an equal number of positive and negative samples. After the training process of Naïve Bayes classifier is completed, it is used to classify the segments collected from the test dataset. For this purpose, the segmentation and feature extraction methods which are exactly the same ones utilized for the selected object recovery process are performed on the test images. Afterwards, the segments are classified based on their feature descriptors by the already trained Naïve Bayes classifier.

The second implemented technique is a linear SVM classifier which establishes a hyperplane to separate two different classes by maximizing the margin for both sides in the feature space [22]. This classifier is also initially trained by using an equal number of positive samples from a pre-trained object model and randomly gathered segments from negative training dataset same as the previous approach. In addition, an optional supportive training process is employed in order to re-train the already trained linear SVM classifier with hard examples (false positives) same as presented by Dalal et al. [50]. For this purpose, the feature descriptors are extracted from the segments of 10 randomly selected negative test images by employing the same segmentation and feature extraction techniques as the one utilized during the construction of the pre-defined object model. Afterwards, the extracted feature descriptors are classified by the already trained linear SVM classifier. The hard examples, which are classified as positives, are collected and used to train the SVM classifier once again. A set of positively image-level labeled test images is also segmented by executing the same segmentation technique and then the same feature extraction technique is applied to these obtained segments. Finally, the ultimate version of this classifier is used to detect target object segments by classifying their feature descriptors. (If this optional training process is not set as activated, then the linear SVM classifier is used for the detection of target object segments once its core training process, as described above, is completed.)

The third alternative approach does not require any training process to start detecting target object segments from positively image-level labeled test images. Firstly, the feature descriptors are collected from these test images in exactly the same way as in the previously described approaches. The distance measures between these feature descriptors extracted from the test images and the feature descriptors of the segments placing in the determined object model are computed with the same distance function used during the process of object model recovery. Each of these distance measurements is normalized between 0 and 1 by aligning based on the segments having the greatest distances to the respective fingerprints during the information gain process of a selected visual object recovery. Once the distance measurements are completed, if the lengths of distances between a feature descriptor obtained from test images and the feature descriptors of at least a selected percent of the segments placing in the determined object model are less than the threshold values of the respective feature descriptors from the object model, then the test segment where the feature descriptor is extracted from, is detected as positive; otherwise, negative. Figure 3.23 illustrates the threshold spaces utilized for the detection of test segments with the way explained in the previous sentence. Therefore, one should note that the spherical surface of each spherical threshold space constructed by using a selected fingerprint from a determined object model and its threshold value is defined as a negative zone. Moreover, Figure 3.24 presents the fingerprints, which are used to generate threshold spaces based on their threshold values. As it is explained in the previous section 3.3, the thresholds of the segments placing in the object model are collected during the learning process on the basis of the information gain.

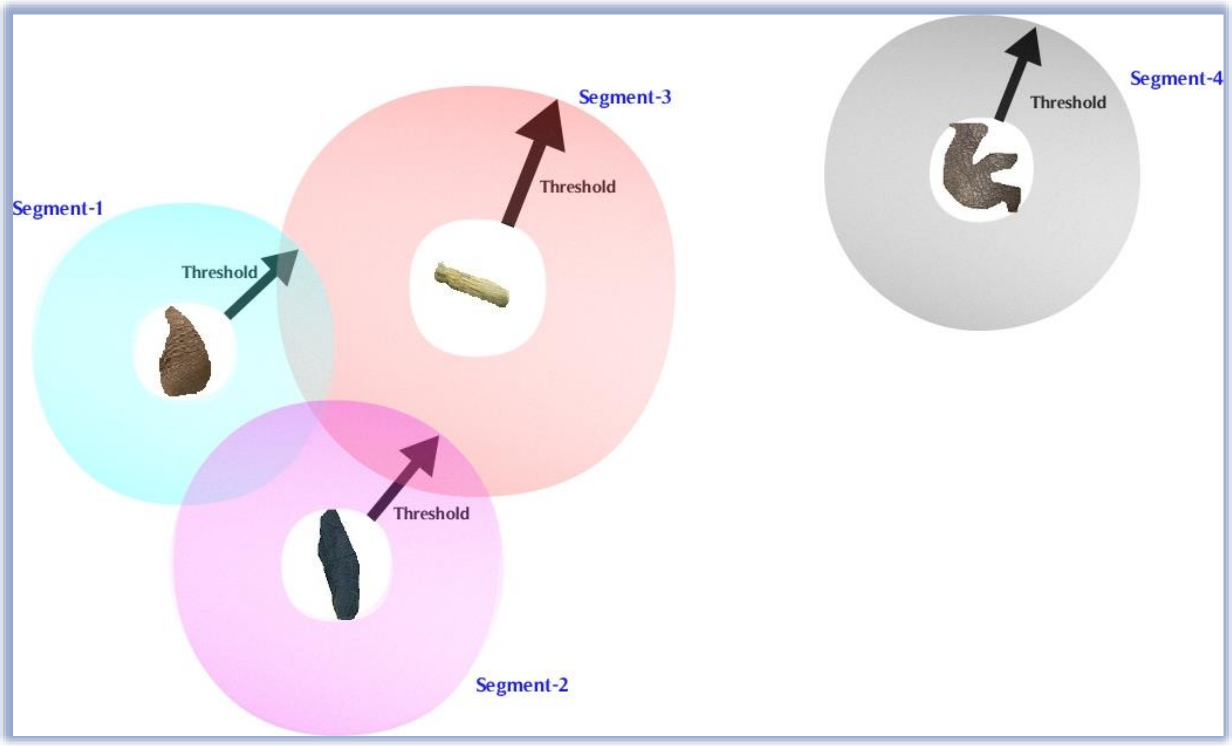


Figure 3.23: The visualization of the threshold spaces created by different fingerprint samples shown in Figure 3.24 in order to detect a test sample as positive or negative.

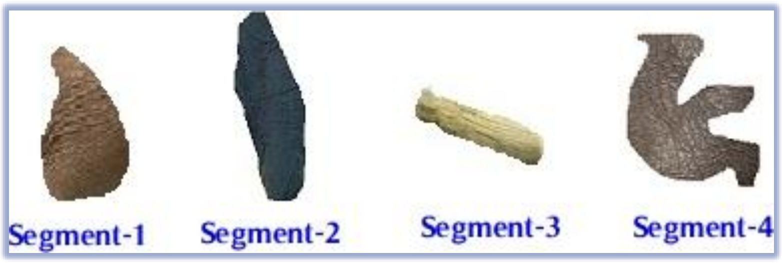


Figure 3.24: 4 different fingerprint samples of an optimal object model.

The detected segments whose boundaries intersect partially or fully, are merged into a single segment. After the merging process, the merged segments are visualized with boundary boxes.

CHAPTER 4

4 EVALUATION

The details of overall settings and ingredients in order to run each experiment existing in the experimental scope of the thesis are expressed in this chapter. Two different datasets are collected in this thesis. The dataset used for the development is called development dataset or elephant dataset. Another dataset is used to evaluate generatability to other visual objects and it is called tiger dataset. Moreover, this chapter lists the degrees of freedom for each experimental run and ends with the explanation how observations are assessed.

4.1 DATASETS

Each of the following subsections gives the details of one of the datasets utilized in this thesis. The development dataset is used for all the experimental runs. Moreover, only the selected experimental runs are launched by using the tiger dataset in order to verify the stability of their results which are previously obtained by using the development dataset.

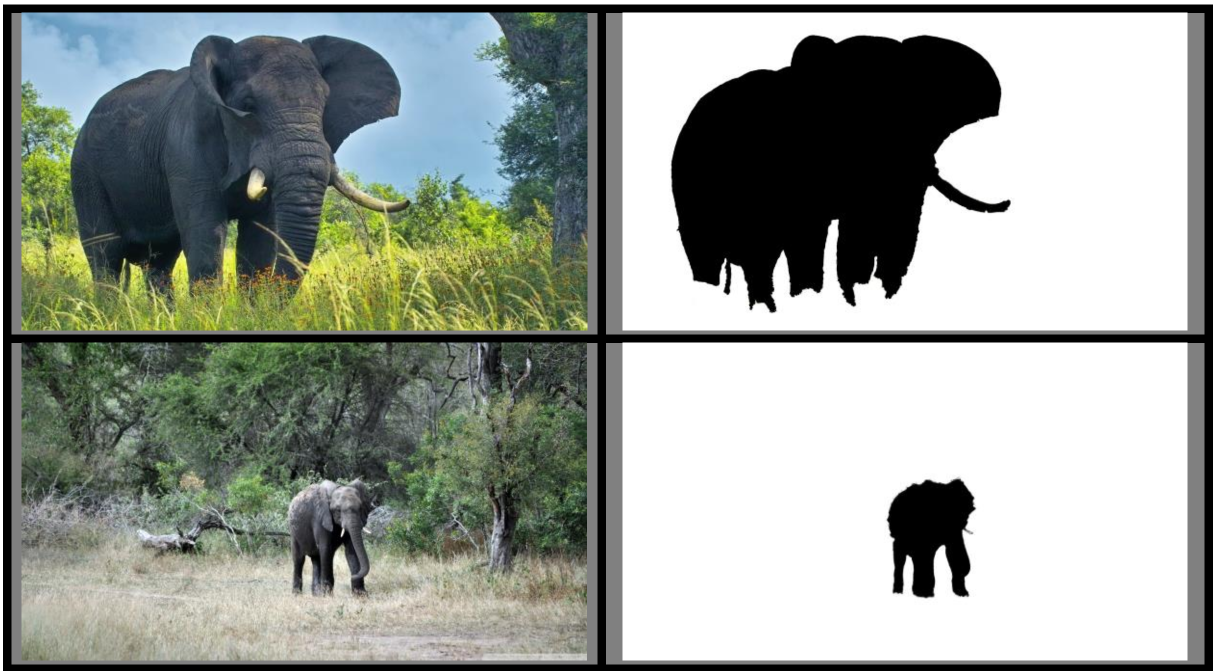
4.1.1 DEVELOPMENT DATASET

The development dataset comprises of color images, which are 1920 pixels in width and 1080 pixels in height HD images (1920×1080) and image-level labeled as positives or negatives. The images which contain at least one segment presenting partial or entire view of single or multiple elephants are labeled as positives. The rest of the images existing in the dataset is labeled as negative. Therefore, this dataset is also called elephant dataset. There are 50 positively and 40 negatively image-level labeled images in the development dataset.

A ground truth dataset related to the development dataset is created by using Adobe Photoshop CS6 as shown in Figure 4.1. This ground truth dataset represents the pixel-level labels of all the positively image-level labeled images of the development dataset.

The development dataset is partitioned into a training dataset and a test dataset. 30 positively and 30 negatively image-level labeled images of the development dataset forms the training dataset. These image-level images of the training dataset are used for building the object models. Furthermore, a set of segments collected from the training dataset is also used to train either an SVM classifier or a Naïve Bayes classifier employed in the detection of target visual objects. The images which form the half of the training dataset are taken personally by the supervisor of the thesis in the real wild environments of elephants. The rest of the training dataset is gathered from Flickr [11].

The test dataset contains 20 positively and 10 negatively image-level labeled images and is utilized only to evaluate the accuracies of investigated three different detection approaches based on different recovered object models. The complete test dataset is also collected from Flickr [11]. The negatively image-labeled images of the test dataset are used only to catch “hard examples” for the already trained SVM classifier and then train the classifier once again with these hard examples. Since this is a supportive and optional training process, all the evaluations of detection approaches do not require the negatively image-level labeled images of test dataset.



(a) Positive image samples from the elephant dataset

(b) The respective ground truth data for (a)

Figure 4.1: The presentation of two positive sample images from the elephant dataset and their ground truth data.

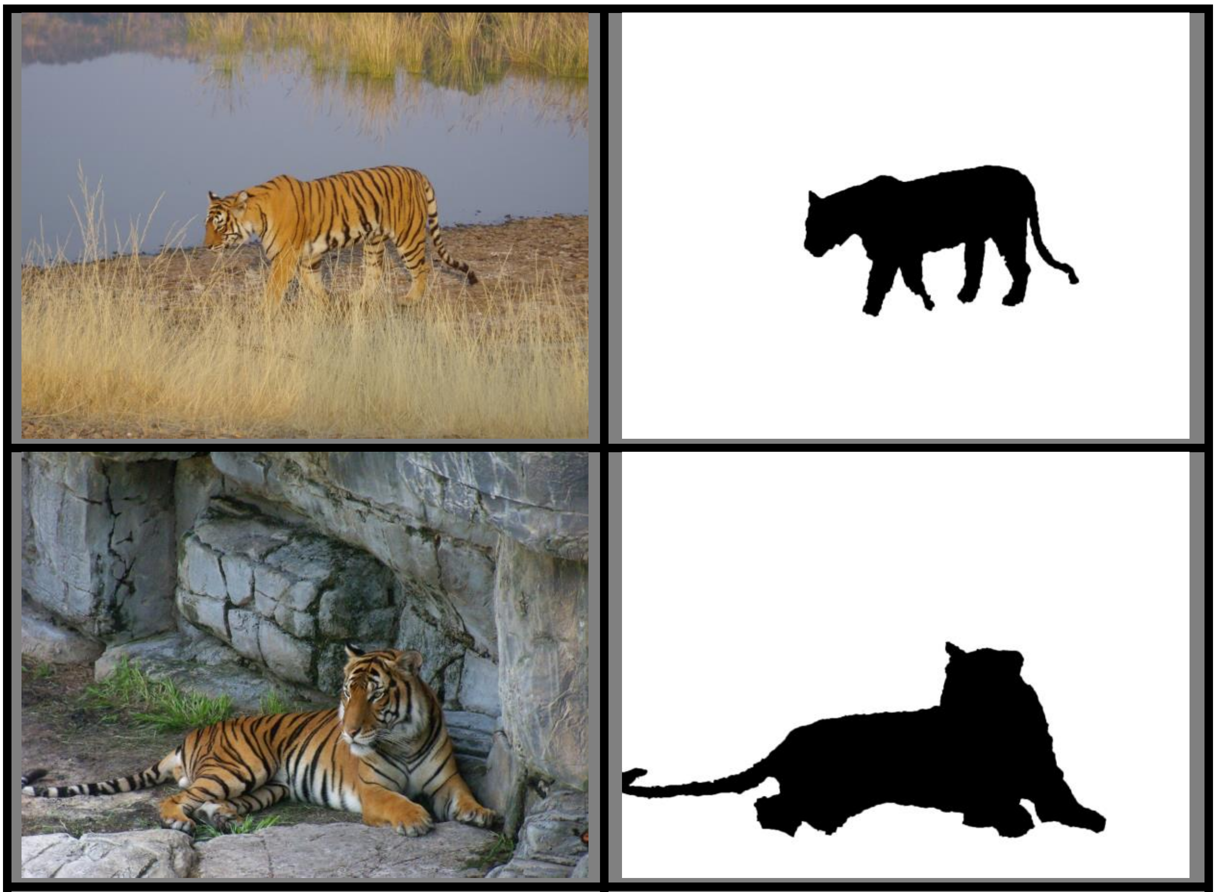
4.1.2 VERIFICATION OF GENERALIZABILITY

A tiger dataset is used for the verification of generalizability. The sizes of all the color images existing in the tiger dataset are not the same, contrary to the ones belonging to the development dataset but most of them are 1920 pixels in width and 1280 pixels in height HD images (1920×1280) or very close to these dimensions. All of these color images are also gathered from Flickr [11]. The labeling structure of the tiger dataset are exactly the same with the one of the development dataset. The only difference in the tiger dataset than in the development dataset is that the species of tiger is selected as a positive visual object rather than the species of elephant. 50 positively image-level labeled images forms the tiger dataset.

A ground truth dataset is also created by using the application of Adobe Photoshop CS6 in order to have pixel-level labels of the tiger dataset. Figure 4.2 shows two samples from the positively image-labeled training images of the tiger dataset and their ground truth data.

The tiger dataset is likewise partitioned into a training dataset and a test dataset. The number of images pertaining to the training dataset is 30 and all of these images are positively image-level labeled. When an experimental run is launched to build a tiger model, the negative images belonging to the training images of the development dataset are used as a negatively image-level training dataset.

The test dataset contains only 20 color images which are negatively image-level labeled. This test dataset does not contain any additional set of negatively image-level labeled images in order to be used as “hard examples”, unlike the test dataset existing in the development dataset.



(a) Image samples from the tiger dataset | (b) The respective ground truth data for (a)

Figure 4.2: Two samples from the positive images of the tiger dataset with their ground truth data.

4.2 PARAMETER SELECTION

EDISON [85] implementation of mean shift segmentation is employed for the investigation of this thesis. Timofte et al. [86] respectively set the bandwidths of spatial and color spaces to the default values determined in EDISON tool [85] as 7 and 6,5 for the mean shift implementation used in their research. Wang et al. [87] introduce the highest accuracies regarding their initial experiments with the employment of mean shift segmentation running for a large dataset by setting the following parameters as follows: spatial space, color space, and minimum segmented region are respectively set to 49, 30.5 and 7000 pixels. Based on these parameter settings, Wang et al. [87] suggest three different sets of parameters, which can be set with different combinations to obtain the satisfactory accuracies from the mean shift segmentation. The values of bandwidths of spatial and color spaces set by Timofte et al. [86] also exist in two of parameter sets suggested by Wang et al. [87]. Therefore, based on these two references and default selections in [85], the same bandwidth values of spatial and color spaced set in [86] are also assigned to the same parameters of the mean shift segmentation employed in this thesis as specified above. A parameter is selected from the third parameter set suggested in [87] as 200 pixels in order to set the minimum area of the segmented region for some experimental runs of this thesis. Moreover, the minimum segmented region is set to 300 for other experimental setups to speed up their executions although this value is not included in the parameter set presented in [87] regarding the minimum segmented region. If CEDD or MPEG-7 color descriptor is selected as the feature extraction technique or CK-1 distance is employed to measure distances among segments, the minimum area of the segmented region is set to 200 pixels; otherwise to 300 pixels. The radius of the gradient window is set to 2 since this value is also determined as the default value for this parameter in [85]. The rest of the parameters required to be set for mean shift segmentation is also selected and assigned based on the default parameter selections in [85] and alignment made in [86] as follows: activation of synergistic segmentation, setting 1 to the level of speed up and the assignment both mixture parameter and the strength of the edges to 0.3.

According to the comparisons among the precision scores of recovered object models using MPEG-7 color descriptors by setting all combinations of possible parameters, the ones which provide the highest precision scores are determined as the values of parameters to set for MPEG-7 color descriptors. Consequently, the number of coefficients is set to 32 for MPEG-7 color structure descriptor and set to 16 for the MPEG-7 scalable color descriptor. In addition, the side length of structure element is aligned with 8.

The number of principal components is also determined for specific feature descriptors by selecting the highest precision scores gathered from some of the object models which are built during the experimental runs with about dozens of different numbers of principal components for each selected feature extraction technique using entire training dataset. If the application of PCA increases the precision scores of the recovered object models for a feature extraction technique, a PCA process is performed based on a selected number of dimension reduction for the respective feature descriptors. Therefore, the dimension reduction by performing PCA is only applied to AB, UV, third algorithm of RGB color histograms (described in Subsection 3.2.1.4) and VLAD encodings. The dimensions of AB color histograms are reduced to 6, while the number of principal components for UV color histograms is 8 after the dimension reduction by the application of PCA. Moreover, PCA reduces both dimensions of RGB color histograms described as the third algorithm in Subsection 3.2.1.4 and VLAD encodings to 12. The number of visual words placing in the visual vocabulary is determined as 16 for the extraction of VLAD descriptors by referencing from [75].

The third approach for the detection of target visual objects classifies the segments as positives if they lie in the intersection of the distance spaces generated by at least 60% of the segments existing in the selected optimal object model with their thresholds as explained in Section 3.4.

4.3 EXPERIMENTAL SETUP

This section firstly presents a complete set of experimental setups which are used to launch the experimental runs for the visual object model recovery. Next, all the experimental setups used in this thesis for the detection of target visual objects are presented.

4.3.1 EXPERIMENTAL SETUP OF VISUAL OBJECT MODEL RECOVERY

The overall configuration used in this thesis to recover a visual object model is represented in Figure 4.3. The configuration represented in Figure 4.3 shows all the possible degrees of freedom of the generally defined processing steps as well as the dependencies between the processing steps for the recovery of a visual object model. Each processing step is demonstrated as a black rectangular and defined with a red colored headline. In addition, an inner rectangular of each processing step shows its degrees of freedom and one of them has to be selected in order to setup an experiment. In the processing step of model building, information gain is the constant process, while a distance function has to be selected for the similarity measurements. For instance, the segments of the input training datasets are obtained by applying either mean shift segmentation or sliding windows.

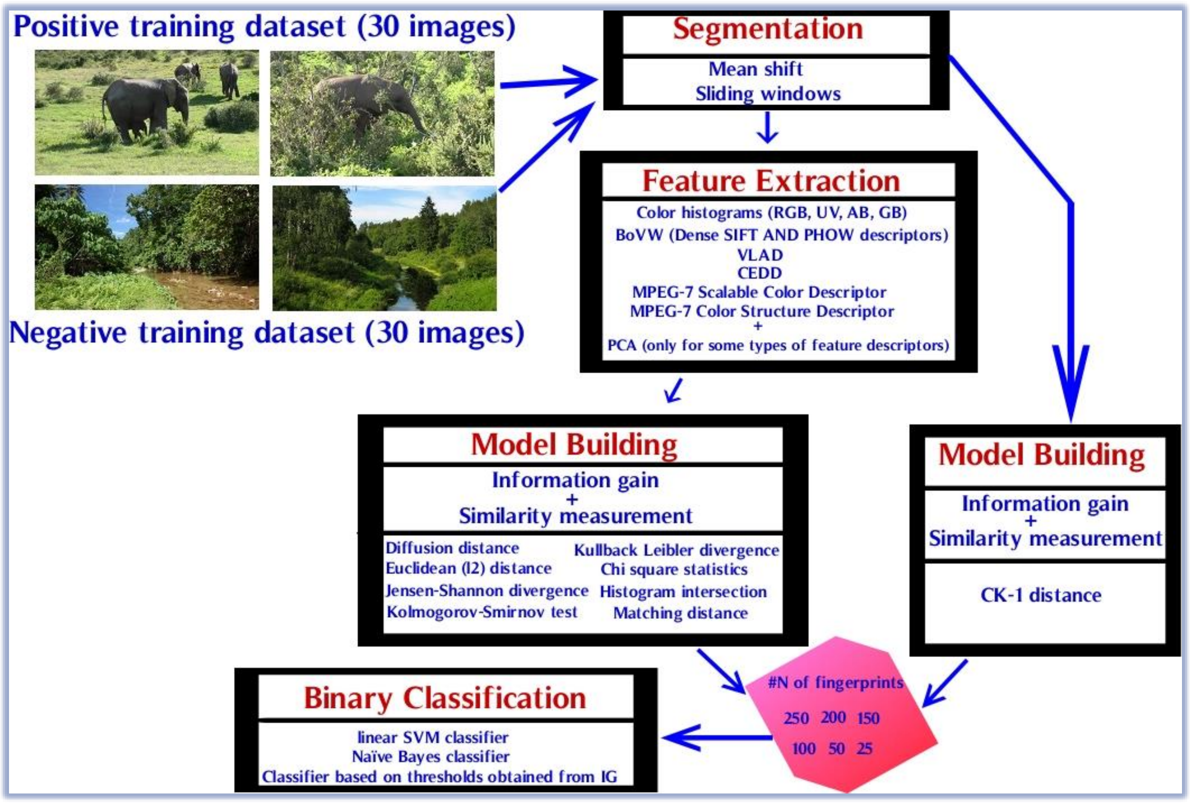


Figure 4.3: Demonstration of the dependencies between the processing steps and their degrees of freedom to setup an experiment with respect to an object model recovery.

Moreover, there are two different methods of sliding windows and one of them has to be selected when an experimental run obtains the segments from a given training dataset by operating sliding windows. The first method of sliding windows produces the segments of a given image which contain only unprecedented image parts. On the other hand, the sampling step size of second sliding windows method is smaller than the dimensions of sliding windows; therefore, the segments which are sequentially obtained by this type of sliding windows method, contain the same image regions.

As it is shown in Figure 4.3 that using CK-1 distance function for the similarity measurement of the image segments bypasses the process of each kind of feature extraction technique. If a feature extraction process is used for an experimental run, a kind of feature extraction techniques presented in Figure 4.3 is selected. As it is mentioned in the previous section, four of these extraction techniques are followed by the execution of PCA. After the feature extraction process is completed during an experimental run, a distance function is selected from eight different kinds of options. The selected distance function is performed to compute the similarity measurements between the feature descriptors which are the global visual features of the image segments collected from both positive and negative training datasets. These similarity measurements computed among the global feature descriptors extracted from the image segments are employed to calculate the values of information gain and threshold for each image segment extracted from the positive training dataset during an experimental run. Afterwards, these image segments are sorted in descending order according to their information gain values. Finally, in keeping with the size determined for the construction of an object model, the first required number of the image segments from these sorted ones is selected as an object model.

In the following subsections, the details (degrees of freedom) of the required processing steps and their experimental setups for the recovery of the object models depending on different feature extraction methods are represented by using tables. Each row except the first one of these tables belongs to a single processing step.

4.3.1.1 VLAD-BASED MODEL RECOVERY

VLAD-based model recovery	DoF	Comments
Segmentation	• Mean Shift segmentation based on CIELUV • Mean Shift segmentation based on CIELAB • Sliding windows with occlusions • Sliding windows without occlusions	No additional process is required.
Feature extraction	• VLAD descriptors with sampling steps of {1,2,5} pixels	
Model building	Information gain and one of eight different distance functions: • Diffusion distance • Euclidean (l_2) distance • Jensen-Shannon divergence • Kolmogorov-Smirnov test • Kullback–Leibler divergence • Chi square statistics • Histogram intersection • Matching distance	

Table 4.1: The representation of all the experimental setups with respect to the object model recovery containing VLAD extraction.

VLAD descriptors are constructed by clustering its local PHOW descriptors based on the visual vocabulary (with the size of 16). Moreover, VLAD descriptors stores distances between the local PHOW descriptors and their assigned closest visual words. The dimension of each VLAD descriptor is reduced to 12 by performing PCA. L_2 and power normalization is performed on the obtained principal components. Table 4.1 shows all the experimental setups utilizing VLAD descriptors in their processing step of feature extraction in order to build object models.

The total number of the extracted dense SIFT descriptors from the selected 10 positive training images is 1251360 when the sampling step of the Gaussian window is set to a single pixel. When the sampling step is switched to 2 or 5 pixels from the single pixel, the total number of the extracted dense SIFT descriptors is respectively decreased from 1251360 to 312840 and 50350.

4.3.1.2 COLOR HISTOGRAMS-BASED MODEL RECOVERY

All the experimental setups consisting of one of the color histograms as the feature descriptors existing in the research scope of the object model recovery are elaborated in Table 4.2.

Color histograms-based model recovery	DoF	Comments
Segmentation	• 4 different segmentation methods same as the ones shown in Table 4.1	No additional process is required.
Feature extraction	• 2-dimensional GB color histograms • UV color histograms • 3-dimensional RGB color histograms • AB color histograms • 1-dimensional RGB color histograms which are generated by concatenating the 2d- histograms of RG, GB and BR channels. • 1-dimensional RGB color histograms which are generated by concatenating the vectors of RGB channels.	
Model building	• Information gain and 8 different distance functions same as the ones shown in Table 4.1	

Table 4.2: The representation of all the experimental setups with respect to the object model recovery containing the extraction of one of color histograms.

2-dimensional GB color histograms are extracted by using only the green and blue channels of the segments and then converted in the form of normalized column vectors. 6 and 8 principal components of the normalized column vectors which are respectively generated from AB and UV color histograms are the outputs of these feature extraction methods.

4.3.1.3 DENSE SIFT AND PHOW DESCRIPTORS-BASED MODEL RECOVERY

There are 30 different experimental setups presented in Table 4.3 with respect to dense SIFT descriptors depending on their segmentation method, sampling step size, vocabulary size and normalization. If SIFT descriptors are extracted at each two pixels of the image segments, they are either normalized or not after their extractions as twenty different experimental setups. The remaining experimental setups contain the extraction of SIFT descriptors from each pixel of the image segments. The dense SIFT descriptors extracted from each image pixel are not normalized. As it is shown in the last row of Table 4.2, all 8 different distance functions described in Section 3.3 are employed for the computations of the similarities of the feature descriptors for each experimental setup presented in this table. After the similarity computations, the information gain values of the segments extracted from the positive training dataset are calculated for each set of similarities computed by using 1 of 8 different distance functions. Moreover, 10 different experimental setups utilizing the PHOW descriptors are shown in Table 4.3.

Dense SIFT and PHOW Descriptors-based model recovery	DoF	Comments
Segmentation	• Mean Shift segmentation based on CIELUV • Mean Shift segmentation based on CIELAB	No additional process is required.
Feature extraction	• Dense SIFT descriptors with sampling step size of single pixel and vocabulary sizes of {50,100,150,200,250} • {Normalized, Non-Normalized} Dense SIFT descriptors with sampling step size of 2 pixels and vocabulary size {50,100,150,200,250} • PHOW descriptors with sampling steps of {1,2} pixels and vocabulary sizes of {50,100,150,200,250}	
Model building	• Information gain and 8 different distance functions same as the ones shown in Table 4.1	

Table 4.3: The representation of all the experimental setups with respect to the object model recovery containing the extraction of either dense SIFT or PHOW descriptors.

4.3.1.4 CK-1-BASED MODEL RECOVERY

Table 4.4 presents the details of the required processing steps and their experimental setups for the recovery of the object models using CK-1 distance function. As seen in Table 4.4, when the mean shift segmentation method is selected for the processing step of segmentation, the segments are collected with their closest outer rectangular boundaries. Afterwards, the temporary resizing process is performed for each pair of segments in order to compute their CK-1 distances. One should note that the resizing process is performed on the segments for the equalization of their sizes during only their related computation of CK-1 distance function and after their distance is computed, the segments remain in their original sizes.

Moreover, it is seen in Table 4.4, since the computation of CK-1 distance function does not require any feature extraction process as a predecessor processing step, a single process step of model building containing the sequential computations of CK-1 distance function among the image segments and the information gain values of the segments extracted from only the positive training dataset based on their computed CK-1 distances. Therefore, the information gain values are computed as the successor of the computation of CK-1 distance function among the image segments in a single processing step. Furthermore, performing one of the methods of sliding windows in a processing step of segmentation does not require additional successor processes for the boundary selection and resizing of the image segments.

CK-1-based model recovery	DoF	Comments
Segmentation	• 4 different segmentation methods same as the ones shown in Table 4.1	If segments are gathered by one of mean shift segmentation methods, the segments are extracted with their closest outer rectangular boundaries and resizing segments is processed.
Feature extraction	Not required	Not required
Model building	• CK-1 distance function and information gain.	Similarity measurements between segments are calculated by CK-1 distance. Object models are built by learning similarity measurements based on the information gain.

Table 4.4: The representation of the experimental setups for the employment of CK-1 distance function.

4.3.1.5 MPEG-7 COLOR DESCRIPTORS-BASED MODEL RECOVERY

Table 4.5 elaborates the degrees of freedom, supplementary and exceptional processes existing in the experimental setups used for the investigation of the effects obtained from the feature descriptors such as MPEG-7 color descriptors.

MPEG-7 Color Descriptors-based model recovery	DoF	Comments
Segmentation	• 4 different segmentation methods same as the ones shown in Table 4.1.	If segments are gathered by one of mean shift segmentation methods, the segments are extracted with their closest outer rectangular boundaries as an only successor supplementary internal process of the segmentation.
Feature extraction	• MPEG-7 color structure descriptors • MPEG-7 scalable color descriptors	No additional process is required.
Model building	• Information gain and 8 different distance functions same as the ones shown in Table 4.1	Shannon divergence and Kullback–Leibler divergence are excluded from the similarity measurements of MPEG-7 scalable color descriptors because they are not applicable to this type of descriptors.

Table 4.5: The representation of all the experimental setups with respect to the object model recovery containing the extraction of one of MPEG-7 color descriptors.

4.3.1.6 CEDD-BASED MODEL RECOVERY

CEDD descriptors can be extracted from the segments having rectangular boundaries same as MPEG-7 color descriptors. Therefore, the requirement of a supplementary process is likewise defined for mean shift segments in the details of experimental setups utilizing CEDD descriptors shown in Table 4.6.

CEDD-based model recovery	DoF	Comments
Segmentation	• 4 different segmentation methods same as the ones shown in Table 4.1	If segments are gathered by one of mean shift segmentation methods, the segments are extracted with their closest outer rectangular boundaries as an only successor supplementary internal process of the segmentation.
Feature extraction	• CEDD descriptors	No additional process is required.
Model building	• Information gain and 8 different distance functions same as the ones shown in Table 4.1	

Table 4.6: The representation of all the experimental setups with respect to the object model recovery containing the extraction of CEDD.

4.3.2 EXPERIMENTAL SETUP OF TARGET VISUAL OBJECT DETECTION

The overall schema of the experimental installations used for launching the experimental runs regarding the target visual object detection as well as the elaborations of their dependencies are demonstrated in this subsection. Figure 4.4 illustrates the degrees of freedom for each processing step employed for the detection of target visual object in an overall schema. As seen in Figure 4.4, the segments of the images existing in the positive test dataset are collected by performing the mean shift segmentation on them. Afterwards, the global feature descriptors are extracted from these segments. Selection of the type of feature descriptors is made based on the type of the ones utilized for training or construction of the determined classifier. The selected object models containing these global feature descriptors as fingerprints are shown in the violet cube presented in Figure 4.4 and each of them is used by a classifier.

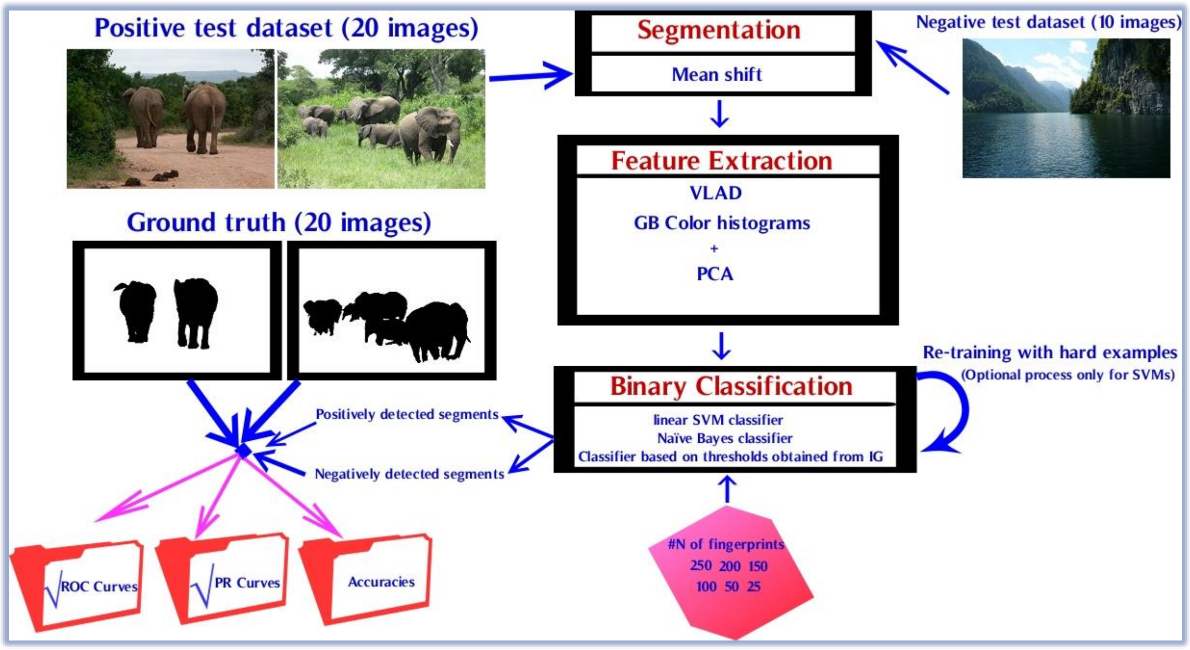

Positive test dataset (20 images)
Segmentation
Mean shift
Negative test dataset (10 images)
Feature Extraction
VLAD
GB Color histograms
+
PCA
Ground truth (20 images)
Re-training with hard examples
(Optional process only for SVMs)
Binary Classification
linear SVM classifier
Naïve Bayes classifier
Classifier based on thresholds obtained from IG
Positively detected segments
Negatively detected segments
#N of fingerprints
250 200 150
100 50 25
ROC Curves
PR Curves
Accuracies

Figure 4.4: Demonstration of the dependencies between the processing steps and their degrees of freedom to setup an experiment with respect to the detection of target visual objects.

Only the sets of object models having the highest mean precision scores are selected in order to train a binary classifier. Since the object models owning the highest mean precisions scores are recovered by extracting either VLAD descriptors or GB color histograms, they are the only options in the processing step of feature extraction employed for the detection of a target visual object from the test dataset. Moreover, PCA is performed on both of these global feature descriptors shown in Figure 4.4 same as in the object model recovery. Similarly, since both VLAD descriptors and GB color histograms are extracted from the segments obtained by applying only mean shift segmentation, it is set as the only possible method in the processing step of segmentation. Once the execution of PCA is completed, the obtained principal components are sent to a selected classifier. If a linear SVM classifier or a Naïve Bayes classifier is selected in the processing step of classification, the input data coming from the positive test dataset is used for the core training of the selected classifier. There is an optional re-training process only for the linear SVM classifiers, which is executed as a successor process of the core training by using the negative test dataset. For this purpose, the feature descriptors of the negatively image level-labeled test dataset are firstly extracted by using exactly the same experimental setup which is employed in order to collect the input data for the core training. Afterwards, these collected descriptors are classified by the already trained SVM classifier and the false positives are gathered in order to re-train the classifier by using them. On the other hand, if the selected classifier of the respective experimental setup is the one working based on the thresholds of the fingerprints obtained from information gain, it directly classifies the global feature descriptors extracted from the positive dataset by using these thresholds as explained in detailed in Section 3.4. These foregoing fingerprints are also sent to the processing step of classification from the selected object model placing in the violet cube shown in Figure 4.4. Finally, ROC curve, PR curve and accuracy belonging to a classifier trained based on an experimental setup and a selected object model are computed by comparing its classification results with the ground truth. One should bear in mind that each experimental setup contains a single type of classifiers and each of them is trained by using one object model from the selected set of object models. Therefore, each experimental setup constructs or trains six classifiers. Consequently, each of these classifiers from the same experimental setup receives its own ROC curve, PR curve and accuracy.

The processing details of all the experimental setups in the scope of the thesis with respect to the detection of a target visual object from a given set of color test images are presented in Table 4.7, Table 4.8 and Table 4.9. Each of these tables shows the degrees of freedom and supplementary details of the experimental setups containing a separate classifier approach. Inferring the configuration details of the experimental setups represented in these three tables is the same way how the previous tables of Subsection 4.3.1 are inferred.

4.3.2.1 NAÏVE BAYES-BASED OBJECT DETECTION

Naïve Bayes-based Object Detection	DoF
Segmentation	• Mean Shift segmentation based on CIELUV
Feature extraction	• VLAD descriptors with sampling steps of {1,2,5} pixels • 2-dimensional GB color histograms
Classification	➢ Trained with an object model recovered by using VLAD descriptors and one of three different distance functions: • Diffusion distance (for the ones sampled at each {2,5} pixels) • Euclidean (l_2) distance (for the ones sampled at each 2 pixels) • Chi square statistics (for the ones sampled at each {1,5} pixels) ➢ Trained with an object model recovered by using GB descriptors and Euclidean (l_2) distance

Table 4.7: The configuration details of each experimental setup utilizing only Naïve Bayes classifier in its processing step of classification are demonstrated.

The experimental setups presented in Table 4.7 utilizes only Naïve Bayes classifier in their processing step of classification. As seen in Table 4.7, when a Naïve Bayes classifier is employed in the processing step of classification, either GB color histograms or VLAD descriptors are extracted from the segments of the test images in the processing step of feature extraction. One should note that the employed Naïve Bayes classifier is trained with an object model recovered by using exactly the same feature extraction setup with the one used for the segments of the test images. Apart from that, the segments used in the processing step of feature extraction are collected by applying mean shift segmentation to the test images based on CIELUV color space. There are three different options regarding the sampling steps for the extraction of VLAD descriptors utilized in each experimental installation shown in Table 4.7. For instance, if the extractions of VLAD descriptors from the test images are sampled at each two pixels, it means that the object model training the Naïve Bayes classifier is built by utilizing either diffusion distance or l_2 distance. When the sampling step size is set to single pixel, the fingerprints of the object model are selected by using chi square statistics. Furthermore, sampling at each five pixels referring the Naïve Bayes classifiers trained by using either diffusion distance or chi square statistics. On the other hand, if GB color histograms are extracted in the processing step of feature extraction, l_2 distance is performed during the object model recovery of the fingerprints utilized for the training of Naïve Bayes classifier. Moreover, any supplementary segmentation process such as boundary selection and resizing is not necessary for the experimental setups shown in Table 4.7.

4.3.2.2 SVM-BASED OBJECT DETECTION

Table 4.8 shows all the installation details regarding the processing steps cooperating with linear SVM classifiers. The extraction of the VLAD descriptors from the segments by sampling each two pixels exists in the processing step of feature extraction of each experimental setup presented in Table 4.6. Consequently, the linear SVM classifiers are employed to classify the principal components obtained from these VLAD descriptors. The core training of each of these classifiers is performed with an object model whose recovery contains similarity measurement by using either diffusion distance or l_2 distance. In addition, re-training with the hard examples is also investigated for the linear SVM classifiers which are recovered by using l_2 distance.

SVM-based Object Detection	DoF
Segmentation	• Mean Shift segmentation based on CIELUV
Feature extraction	• VLAD descriptors sampled at each 2 pixels • 2-dimensional GB color histograms
Classification	➢ Trained with an object model recovered by using VLAD descriptors and one of two different distance functions: • Diffusion distance • Euclidean (l_2) distance {No additional re-training, Re-trained with hard examples}

Table 4.8: The configuration details of each experimental setup utilizing only linear SVM classifier in its processing step of classification are demonstrated.

4.3.2.3 THRESHOLDS-BASED OBJECT DETECTION

As shown in Table 4.9, the experimental setups containing the classifier constructed based on the thresholds of the fingerprints entails only VLAD descriptors extracted by sampling each two or five pixels for the processing step of feature extraction. During the calculation of these thresholds by using information gain, diffusion distance is employed. Furthermore, mean shift segmentation is performed on CIELUV color space in the experimental setups of Table 4.9 without any additional process same as the ones shown in Table 4.7 and Table 4.8.

Thresholds-based Object Detection	DoF
Segmentation	• Mean Shift segmentation based on CIELUV
Feature extraction	• VLAD descriptors with sampling steps of {2,5} pixels
Classification	• The classifier which works based on the thresholds of the fingerprints calculated by employing information gain. The fingerprints are obtained from the object model recovered by diffusion distance.

Table 4.9: The configuration details of each experimental setup containing only the classifier constructed by using the thresholds of selected fingerprints in its processing step of classification are demonstrated.

4.4 EVALUATION MEASURE

Since the target of recovering visual object models is to collect only the most discriminative candidate positive segments based on the selected target visual object, the evaluations are performed only on true and false positives by comparing the selected segments with the ground truth data. These selected segments which are the most discriminative candidate positive segments for each object model are called fingerprints as described in Subsection 3.4. The precision score of each investigated object model is computed by using the numbers of true and false positives of its fingerprints. The equation of precision is shown in the equation 4.1.

$$precision = \frac{TP\ (the\ number\ of\ true\ positives\ )}{TP\ (the\ number\ of\ true\ positives\ ) + FP\ (the\ number\ of false\ positives\ )} \quad 4.1\ [22]$$

As specified in Section 3.3, while calculating the precision scores of object models, fingerprints are set as true positives if and only if the intersections of fingerprints with ground truth data cover minimum 80% of complete regions of fingerprints. There are two purposes to set minimum covering region for the fingerprints as 80%. The first purpose is to provide a challenging examination to deeply evaluate capabilities of introduced segmentation techniques regarding cooperation with other components as well as segmenting pure segments. The second purpose is to recover object models by gathering pure fingerprints excluding the background data as much as possible and with the maximum possible number of such pure fingerprints.

Apart from that, the size of each object model is defined based on the number of fingerprints which are stored in the object model. For each experimental setup of visual object model recovery, six different sizes of object models such as 25, 50,100, 150, 200 and 250 are recovered. The mean value of the precision scores gathered from these six different object models is computed. This computed mean value is called mean precision. If the mean precision obtained from a set of object models is equal or greater than 0.85, then this set of object models is evaluated as highly satisfactory. Furthermore, if a set of object models is denoted as satisfactory or closely satisfactory, then its mean precision score is [0.75, 0.85) or [0.65, 0.75), respectively. Otherwise, if the mean precision calculated for a set of object models lies somewhere less than 0.65, then this set of object models is evaluated as unsatisfactory.

ROC curves with respect to the true and false positive rates of the investigated binary classifiers used for the detection of target visual objects from the test dataset depending on the selected object models are demonstrated to evaluate the recognition capabilities of the investigated binary classifiers by checking the areas under ROC curves (AUCs). Since the preferable values of true and false positive rates are respectively 1 and 0, enlarging the area under the ROC curve obtained from a classifier identifies that the recognition capabilities of this classifier are strengthened. Moreover, if the misclassification probability of a point locating on a given ROC curve for a positive instance is equal to the one referring to a negative instance, this point can be used to compute a value of equal error rate (EER) for the given ROC curve. The value of equal error rate can be utilized as an alternative identifier with respect to the recognition capabilities of a classifier. The accuracy (ACC) of each selected classifier is computed by checking its results based on the ground truth dataset of the test dataset. The equation 4.2 shows the equation of accuracy. The labels TP and FP respectively refer to true positives and false positives in the equation 4.2 same as in the equation 4.1. Apart from that, true negatives and false negatives are respectively labeled as TN and FN in the equation 4.2. For each experimental setup of target visual object detection, six different sizes (25, 50,100, 150, 200 and 250) of object models previously constructed by the training process are used. Consequently, for each experimental setup regarding the detection process, a mean value of 6 different accuracies gathered depending on all these object models is computed and this mean value is called mean accuracy. The classifiers are evaluated as closely satisfactory if their accuracies are equal or greater than 0.5 and less than 0.7. Furthermore, the accuracies received from the satisfactory or highly satisfactory classifiers are respectively in the ranges of [0.7, 0.8) and [0.8, 1.0].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad 4.2\ [88]$$

4.5 RUN TIME AND STORAGE EVALUATION

The both of tech and financial companies located all over the world, which either provide the IT services or supply the software services, software development, maintenance, publication and the activities categorized as Research and Development, firstly deploy the new software features in the many development and quality environments and then execute the Non-Regression testing (NRT), gray box testing or visual testing on these environments before releasing these new software features to the production applications. If a totally new software application is planned to be published, then the functional and non-functional testing methods are processed in the development and quality environments before its publication. The NRT, which is a type of black box testing approach same as the functional and non-functional testing methods, detects the inappropriate behaviors of the software application after the deployment of new changes or additions existing in the scope of the release [89]. For this purpose, the benchmark data and the post deployment data is reconciled to find the differences between the outputs produced by the post deployment software application and the software application denoted as the benchmark. These detected differences have to be explained by the experts whether they are expected or not according to the release. The black box testing is a testing approach inspecting the whole set of functions supplied by the selected software asset and does not have any information about the source code of the software [89]. Apart from this, the gray box testing externally examines the functions of the selected software asset same as the black box testing but with the full knowledge of the internal implementation enabling the testing approach to execute the tests with respect to the design of the selected software asset [89]. Furthermore, the visual testing enables the testers to see the errors with the respective comprehensible information at the place occurred in the source code [89].

In particular, the required data, which has to be stored either in the databases or in the file systems of all these environments, constitutes an expenditure item for the firms. Therefore, the firms regularly open research projects to optimize this expenditure item. The number of the utilized development and quality environments differ based on the size of the projects running by the respective firms, the number of the features supported by the software assets, etc. Accordingly, the required number of the development and the quality environments can be even up to one hundred in order to support only one software asset providing the services for the clients in the production environment. The number of the databases used by a commercial software application is mostly more than one. Please note that the databases of these development and quality software applications are loaded with the dump of the databases of the production software application to have a benchmark application or application to be deployed with the new updates. Therefore, the sizes of the databases of the development and quality environments used as the benchmark applications are identical with the sizes of the databases of the production environment in case there is no additional data purging process executed in the databases of these test environments after the refreshment of their databases with the snapshot of the production databases. For instance, if the decreased size of the data stored either in the databases or the file system is 10 GB in the production environment thanks to an implemented optimization approach, then the total number of decreased data placing in the quality and test environments is approximately 1 TB when there is an assumption that the considered company is using one hundred numbers of development and quality environments. Consequently, the smaller sizes for the investigated feature descriptors are evaluated as an advantage in the scope of this thesis. If the execution time frame of a similarity measurement between a type of extracted feature descriptors is less than 90 minutes, then it is denoted as satisfactory based on its execution performance. One should note that the processor of the exploited workstation is Intel ® Core™ i7 CPU and its RAM is 4,00 GB. Moreover, the operating system installed in this workstation is 64-bit Windows 7 Professional. The execution time frames of the similarity measurements of the entire training dataset are 87 minutes, 44 minutes, 25 minutes and 56 minutes in order to recover the object models demonstrated in Figure 5.16 and related to diffusion distance, histogram intersection distance, l_2 distance and chi square statistics, respectively. In addition, the executions of the same distance functions respectively take 73 minutes, 30 minutes, 13 minutes and 45 minutes for the calculations of the similarity measurements utilized while building the respective object models shown in Figure 5.17. Therefore, the similarity measurements among the GB color histograms obtained from CIELAB color space take more time than the ones extracted from CIELUV color space. Consequently, the percent of the time difference between the executions of these two setups changes depending on each of these distance functions. Moreover, computing the similarity measurements for VLAD descriptors takes less time than GB color histograms, which are obtained from either CIELAB color space or CIELUV color space.

Since the AB color histograms are reduced to 6 principal components and UV color histograms are reduced to 8 principal components, the distance measurements take less time among the principal components obtained from AB color histograms than the ones obtained from UV color histograms. For instance, the similarity computation of the principal components of UV color histograms extracted from the entire dataset takes about 51 minutes by using diffusion distance while it takes respectively about 45 minutes and 44 minutes for the principal components of AB color histograms and RGB 1-dimensional histograms by using the same distance function. Therefore, the execution time frames for the similarity measurements of the feature descriptors with respect to AB color histograms and RGB 1-dimensional histograms are very close to each other.

The execution time frames obtained from the similarity computations using Kolmogorov-Smirnov test among either dense SIFT or PHOW descriptors change from about 6 minutes to 9 minutes depending on the size of selected visual vocabulary. When the size of the selected visual vocabulary becomes bigger, the respective execution time frame also becomes bigger. The similarity measurements of the VLAD descriptors which are extracted for the recovery of the object models shown in Figure 5.1 take about 8 minutes in total by performing Kolmogorov-Smirnov test. Moreover, diffusion distance runs about 51 minutes and chi square statistics runs about 9 minutes while computing the similarities among either dense SIFT or PHOW descriptor. Apart from that, the runtime of chi square statistics is also 9 minutes for VLAD descriptors extracted based on the experimental setup of Figure 5.1. Consequently, the execution time frames received from the similarity computations of dense SIFT, PHOW and VLAD descriptors using both diffusion distance and chi square statistics are very similar to each other. They are not exactly equal to each to other due to the differences of the execution time frames only in the seconds. The runtime of chi square statistics is about 5 times longer for GB color histograms than dense SIFT, PHOW and VLAD descriptors.

The runtimes of CK-1 distance function for the mean shift segments, the sliding windows with occlusions and the ones without occlusions obtained from the complete training dataset of the thesis are respectively about 2 and half days, 3 days and 6 days. Therefore, the execution performance of CK-1 distance function is definitely unsatisfactory for each kind of these segments. The runtimes of all the distance functions performed with VLAD descriptors are about 23% shorter than the ones performed with CEDDs and about 41% shorter than the ones used for similarity measurements of MPEG-7 color descriptors.

CHAPTER 5

5 RESULTS

This chapter presents both qualitative and quantitative results which are collected by running experiments depending on two different groups of the setups elaborated in Section 4.3 (experimental setups of visual object recovery and target visual object detection). Firstly, the chapter starts with the demonstration of the qualitative results obtained from the experimental runs which are launched according to the setups prepared for the investigation of object model recovery from the development dataset described in Subsection 4.1.1. Secondly, the chapter continues with the representation of the quantitative results which are also obtained by using both the experimental setups of visual object model recovery and the development dataset. Thirdly, the qualitative experimental results of detecting a target visual object from the development dataset are illustrated in this chapter. Fourthly, its quantitative results gathered from the same dataset are represented. Finally, Chapter 5 concludes with the demonstration of the results gathered based on the selected experimental setups by using the tiger dataset shown in Subsection 4.1.2.

5.1 RESULTS OF VISUAL OBJECT MODEL RECOVERY

The results demonstrated in this subsection are produced by using the development dataset which contains positively and negatively image-level labeled images depending on the existence of the species of elephant. The models built by using the same distance function is specified with the same color in all the figures presenting the precision scores of the recovered visual object models. Apart from that, the nodes presenting the precision scores of the object models which are built by using the same distance function are labeled with the same sign and jointed with a single line segment. Consequently, these line segments allow the readers of this thesis to see the fluctuations of the precision scores among the object models which are built based on exactly the same experimental setup but store different numbers of fingerprints. If there is bigger number of fingerprints in an object model than another one, then this object model is called a bigger object model than another one due to the increased number of stored fingerprints. Likewise, if there is less number of fingerprints in an object model than another one, then this model is called a smaller object model than another one. The results collected from dense SIFT-based and PHOW descriptors-based model recovery is described and illustrated in Appendix A.3.

5.1.1 VLAD-BASED MODEL RECOVERY

Figure 5.1 shows the precision scores of the models built by using VLAD descriptors which possess an extraction procedure from the interest points sampled at each single pixel of a given image. As it is seen in Figure 5.1 that Kolmogorov-Smirnov Test generates continuously increasing precision scores with each shift from a bigger object model to a smaller object model while the number of stored fingerprints decreases. Moreover, excluding the shift from the object model keeping 50 fingerprints to the object model keeping 25 fingerprints, the precisions of the object models built by using matching distance function progressively increase while shifting from a bigger object model to a smaller object model. Although the shift from the object model keeping 50 fingerprints to the smallest presented object model regarding matching distance function does not increase the precision score, it also does not decrease the precision score. In other words, the precision score stays the same when the number of fingerprints stored in the object model built by using matching distance function decreases from 50 to 25, and it is 0.88. Therefore, if the matching distance function is selected for the computations of the similarity measurements based on the above-mentioned experimental setup of Figure 5.1, then the object model keeping 25 fingerprints contains 22 true positives and the object model keeping 50 fingerprints contains 44 true positives.

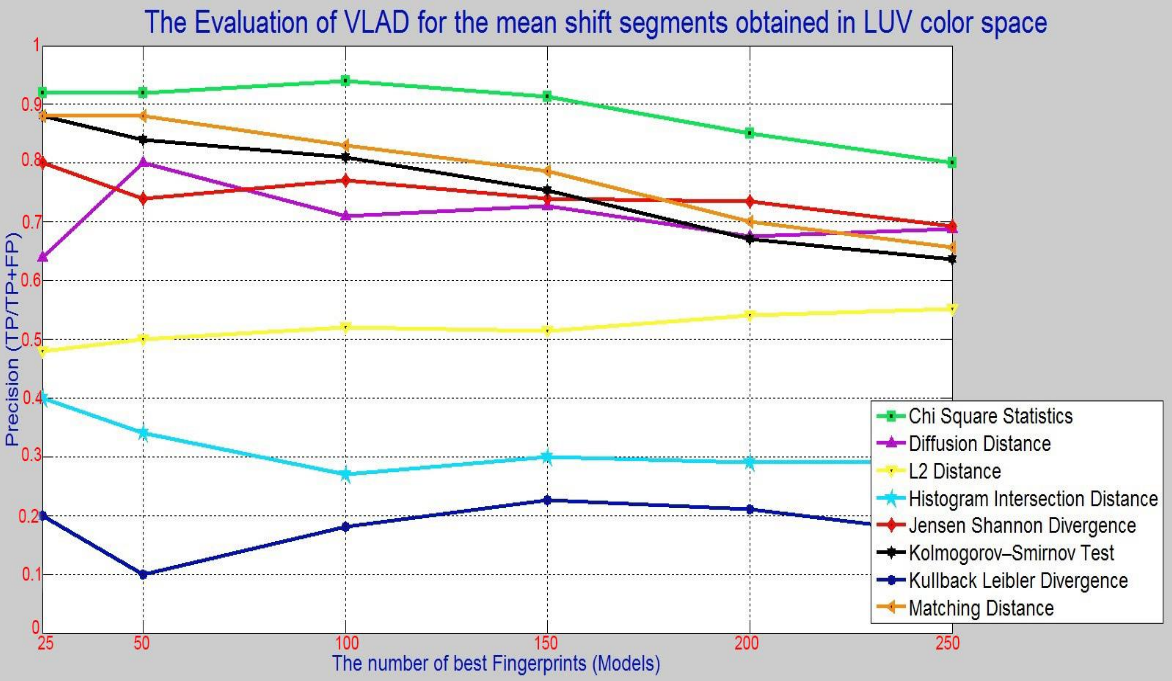


Figure 5.1: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each single pixel from the mean shift segments.

The highest mean precision is obtained from the models built by using chi square statistics when the experimental installation is based on the extraction of VLAD descriptors from the dense grid points sampled at each pixel location of a given image. The fluctuation of the curve among these object models built by using chi square statistics shows in Figure 5.1 that the precision score slightly decreases only during a single shift. Nevertheless, since the mean precision of this set of object models is 0.8905, this set of object models is evaluated as highly satisfactory and so can be employed for one of the investigated detection approaches. Moreover, the mean precision scores of the previously mentioned sets of object models built by using either Kolmogorov-Smirnov Test or matching distance are respectively 0.7648 or

0.7887. Therefore, both of these sets of object models are evaluated as satisfactory and can be taken into account in order to be utilized by an investigated target visual object detection approach. In addition, if the similarity measurements of the visual descriptors are computed by Jensen Shannon divergence, the set of object models which is based on the same experimental configurations used for obtaining the results presented in Figure 5.1 is evaluated as closely satisfactory. This evaluation is made based on its mean precision score which is 0.7439. The remaining sets of object models presented in Figure 5.1 are unsatisfactory since they are under the value of 0.65.

In Figure 5.2, the object models are built by using VLAD descriptors which are extracted from dense grid points sampled at each two pixels. Although the only change is the sampling step size during the building processes of object models regarding Figure 5.1 and Figure 5.2, the majority of the obtained precision scores of the object models is different with respect to each distance function based on the same number of stored fingerprints shown in Figure 5.1 and Figure 5.2. Based on the black curve presented in Figure 5.2, the precision scores of the object models built by Kolmogorov-Smirnov Test increase if there is a shift to a smaller object model of this set. Therefore, the tilt directions of the line segments placing in the black curve presented both in Figure 5.1 and Figure 5.2 are the same while their tilt angles are different. Furthermore, the mean precision of the set of object models regarding Kolmogorov-Smirnov Test based on the experimental setup of Figure 5.1 is greater than the ones based on the experimental setup of Figure 5.2. Consequently, the mean precision score which is obtained by using Kolmogorov-Smirnov Test is 0.4137. Hence, this set of object models is registered as unsatisfactory for the usage of object detection. Likewise, the set of object models with respect to matching distance produces an unsatisfactory mean precision, although the shifting to each smaller object model of this set increases the respective precision score.

According to all the sets of object models presented in Figure 5.1 and Figure 5.2, the highest mean precision score is received from the set of object models which is built based on the experimental installation of Figure 5.2 by using diffusion distance. Apart from that, the tilts of the line segments placing in the curve colored with violet identifying the diffusion distance in Figure 5.2 increases to the left upwards as the most preferable situation of the curve. The second highest mean precision score from Figure 5.2 belongs to the set of object models built by using l_2 distance function, which is 0.8612. Furthermore, this mean precision score is the third highest one from all the mean precision scores of the sets of object models represented in Figure 5.1 and Figure 5.2. Since the precision score is greater than 0.85, this set of object models is evaluated as highly satisfactory. The drawback of this set of object models is that there is a decline in the precision score while shifting from the object model keeping 50 fingerprints to the object model keeping 25 fingerprints which is 0.04. The most stable distance functions seem as histogram intersection distance and Jensen Shannon divergence based on Figure 5.1 and Figure 5.2 because the precision scores using histogram intersection distance and Jensen Shannon divergence fluctuate in the regions [0.16, 0.4] and [0.68, 0.8] for both of these experimental setups, respectively. In other words, the region between the curves of either histogram intersection distance or Jensen Shannon divergence from Figure 5.1 and Figure 5.2 is tight enough to decide that these distance functions provide stable behaviors based on both experimental setups with respect to the precision scores of the respective recovered object models. However, these precision scores gathered from object models using histogram intersection distance are quite disappointing (unsatisfactory) and these object models cannot be employed for the investigated detection approaches of target visual objects to collect satisfactory test results.

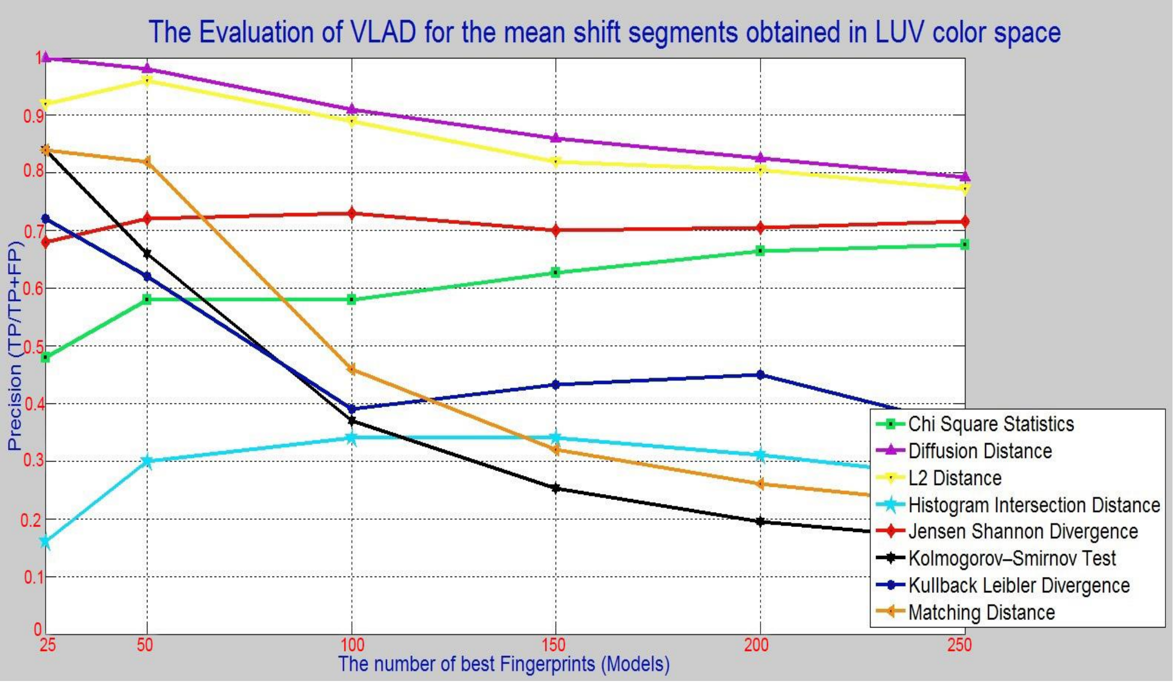


Figure 5.2: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

As seen in Figure 5.2, the set of object models with respect to Jensen Shannon divergence possessing a mean precision score (0.7085) which is a closely satisfactory result. Therefore, the closely satisfactory status of the mean precision obtained from the set of object models built by using Jensen Shannon divergence is stabilized even the experimental setup of Figure 5.2 is interchanged with the experimental setup of Figure 5.1.

In Figure 5.3, the object models are recovered based on the VLAD descriptors which are extracted from the dense grid points determined by moving 5 pixels after each sampling of Gaussian window around the given image segments. The remaining part of experimental installation is exactly the same with the ones used for the recovery of the object models presented in Figure 5.1 and Figure 5.2.

The highest mean precision score is 0.9029 from the mean precision scores of the sets of objects presented in Figure 5.1, Figure 5.2, and Figure 5.3. The set of object models recovered by using chi square statistics based on the experimental setup used for the representation of Figure 5.3 possessing this highest mean precision score. However, this set of object models contains the same drawback which is a decline in the precision score which occurs during the shift from the object model storing 50 fingerprints to smallest investigated object model of this set. Nevertheless, the effect of this decline in the precision score is less sharp than the one occurred during the same type of the shift among the object models recovered based on the experimental configurations of Figure 5.2 by using l_2 distance function. In other words, this decline in the precision score is 0.02 for the objects with respect to chi square statistics instead of 0.04 which is related to latter considered set of object models. Moreover, the diffusion distance provides a highly satisfactory mean precision score for the set of respective object models illustrated in Figure 5.3 while the object models built based on the experimental setup of Figure 5.3 by using l_2 distance function produce a satisfactory result for the score of its mean precision. According to the curve constructed based on the models built by using diffusion distance, the precision score of the model storing 150 fingerprints is 0.067 greater than the precision score of the one storing 100 fingerprints. Since the difference of these precision scores is very small, this drawback can be ignored. Apart from that, the constructed curve visiting the models built by using l_2 distance function has preferable tilt directions for its line segments. As it is seen in Figure 5.3, the mean precision scores with respect to the rest of the distance functions are unsatisfactory.

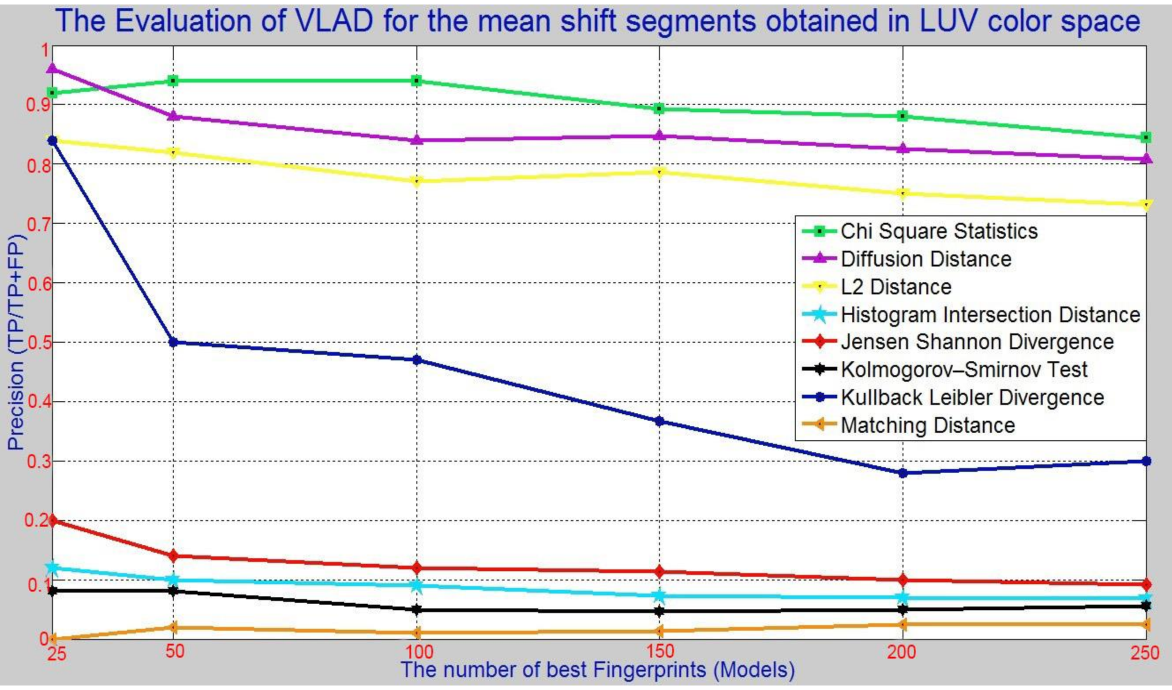


Figure 5.3: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 5 pixels from the mean shift segments.

Figure 5.4 demonstrates the curves traversing precision scores pertaining to the object models built by using VLAD descriptors as well. However, these VLAD descriptors are extracted from each sliding window instead of each mean shift segment. Moreover, the step density between the selected dense grid points of a given image is 5 pixels for the computation of the SIFT descriptors, which are used for the construction of VLAD descriptors for the respective sliding windows. One should note that the sampling step sizes of sliding windows are equal to its dimensions. According to all the mean precision scores of the sets of objects presented in Figure 5.4, all of them apart from the ones built by using Kolmogorov-Smirnov test have unsatisfactory mean precision scores. The mean precision score of the set of object models built by using Kolmogorov-Smirnov test is 0.8217 and evaluated as satisfactory. However, this mean precision score is less than the mean precision scores of 5 different object model sets presented in previous figures.

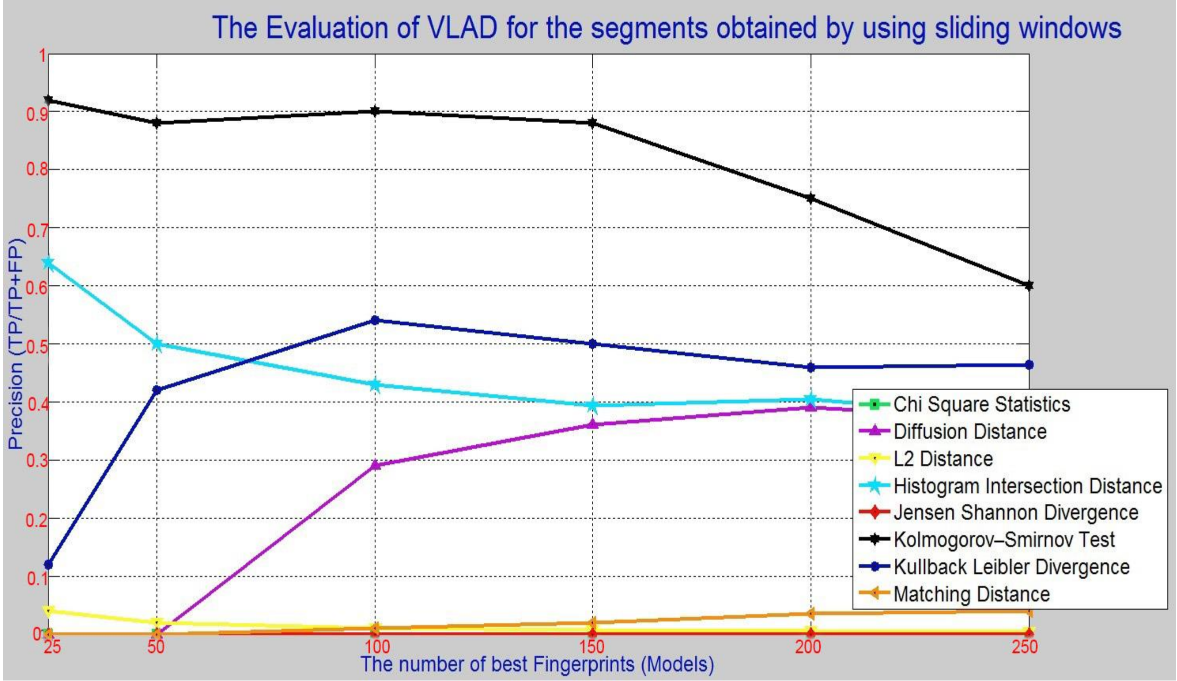


Figure 5.4: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 5 pixels from the sliding windows.

The precision scores of the object models illustrated in Figure 5.5 are received by changing only the sampling density from 5 pixels to 2 pixels for the extraction of SIFT descriptors and keeping the rest of the experimental configurations as they are described for the recovery of the object models presented in Figure 5.4. However, the mean precision scores collected from all the sets of objects shown in Figure 5.5 are less than 6.5 and so evaluated as unsatisfactory. The interesting point is that the models which are shown in Figure 5.4 and Figure 5.5 and built by using Kullback Leibler Divergence have exactly the same precision scores when they store the same numbers of fingerprints. Therefore, their mean precision scores are equal to each other. Apart from that, the precision scores of all the objects models presented in Figure 5.4 and Figure 5.5, and built by using either Jensen Shannon Divergence or chi square statistics are equal to 0. Hence, all of these object models are completely disappointing and evaluated as unsatisfactory.

As it is previously explained that each decrease occurring in the precision score while sifting to the smaller object models in a set belonging to a selected distance function is an undesired sign of the respective curve. The root causes of such undesired signs can be some disorder possibilities appearing during each processing step or information transferring among the processing components. In order to understand and analyze either desirable or undesired fluctuations of a curve, a visualization tool is implemented in the thesis.

This tool illustrates four different images for each training image which is used for the recovery of a visual object model. The first image is the original training image given to the visualization tool and called original image. The second image is the image demonstrating the segments of the original training image after performing the utilized segmentation method by coloring each segment region properly and called segmented image. Apart from them, the third image is called fingerprint candidate rankings and contains the segment regions colored with the greyscale values based on their values of information gain. Furthermore, the fourth image illustrates only the segments which are the fingerprints based on the selected number of the best fingerprints from the complete set of training images and locate in the demonstrated original image based on then given training image to the visualization tool. In other words, if the selected number of the best fingerprints is 50 for a recovered object model, then this means that only the 50 best fingerprints are selected from the complete set of segments extracted from all the training images based on their values of the information gain. Therefore, it is possible that all the 50 best fingerprints can be selected from the segments which are extracted from only one image of the complete training dataset. This also means that each image in the training dataset does not have to have a fingerprint for a recovered object model. Hence, the fourth illustrated image of the visualization tool for a given training image is called the best fingerprints and shows each segment with its original texture and color if this segment is selected as a fingerprint in the recovered object model which is visualized by the tool.

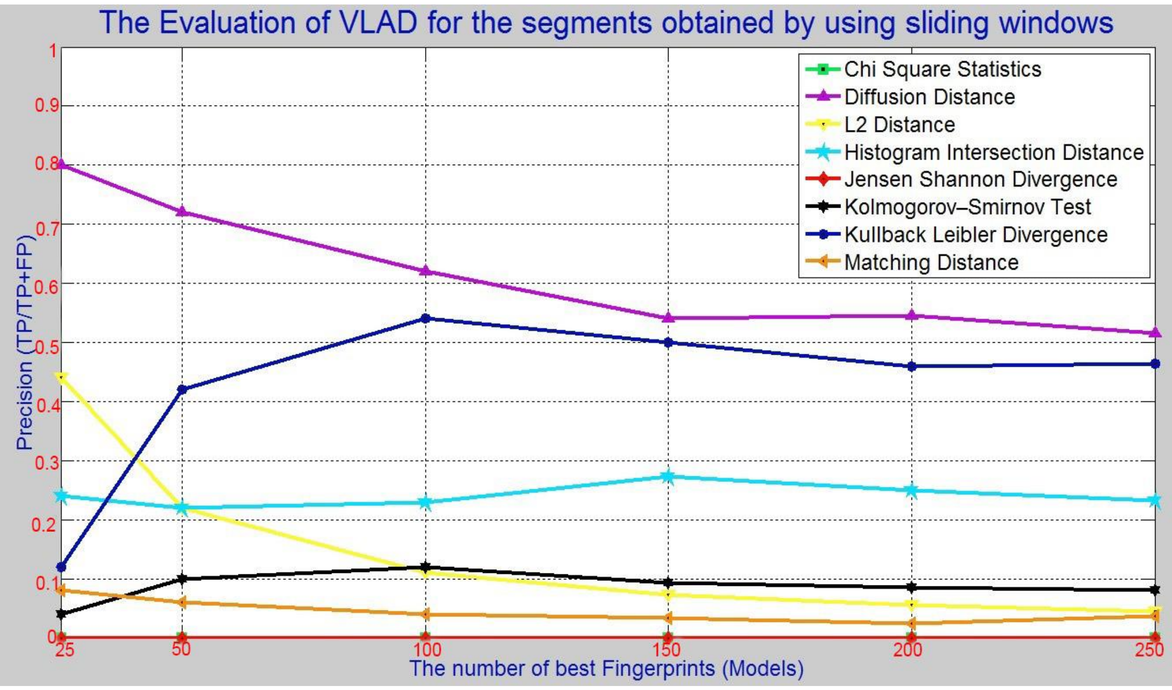


Figure 5.5: The representation of the precision scores received from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the sliding windows.

Coloring the segments placing in the image of fingerprint candidate rankings is made by setting darker grey pixels based on the bigger information gain values. Hence, the darker segments in the images of fingerprint candidate rankings show that these segments hold bigger values of the information gain and vice versa. If the information gain value of a segment is 1 (maximum), then its pixels are set with 0s. The pixels of a segment are set with 255s if its information gain value is 0 (minimum). The pixel values for a given training image are likewise aligned based on the mid-values of the information gain.

Figure 5.6 shows the results obtained by the visualization tool for a sample training image based on four different object models. Each of these object models stores only 50 fingerprints and their experimental installations are selected based on the highest four mean precision scores from all the sets of object models presented in Table 5.2 and Table 5.3. One should note that the training dataset used for the recovery of these object models contains this sample training image. The illustrations of the first two images by the visualization tool are the same images for all these four object models because the selected sample training image and their utilized segmentation methods are the same. Therefore, an original image with respect to all these four models is shown in the top leftmost part of Figure 5.6. Moreover, the top rightmost part of Figure 5.6 illustrates a segmented image for all the models presented in Figure 5.6. In the second row of Figure 5.6, two images representing the fingerprint candidate rankings and the best fingerprints which are received from the object model built based on the experimental installation of Figure 5.2 and diffusion distance by performing visualization tool for the sample training image. The only difference between the illustrations of the second row and the third row of Figure 5.6 is that l_2 distance function is used for the recovery of the object model demonstrated in the third row regarding the sample training image instead of diffusion distance; therefore, the rest of the experimental installation is the same. In addition, the fourth row of Figure 5.6 shows both images of the fingerprint candidate rankings and the best fingerprints which are created by visualizing the object model recovered based on the experimental setup utilized in Figure 5.3 by using diffusion distance. In order to produce the illustrations locating in the bottom of Figure 5.6, chi square statistics is used for the similarity measurements among the visual descriptors instead of diffusion distance as the only difference from all the configurations used for the upper row. When the differences between the images of the fingerprint candidate rankings placing in the second and the third rows of Figure 5.6, it is visible that the segments containing the darker areas of elephants obtained by using diffusion distance mostly have bigger values of the information gain than the ones obtained by using l_2 distance function. Therefore, the segment selected as one of the 50 best fingerprints from the complete training datasets shown in the rightest image of the second row is darker than the segment presented in the rightest image of the third row. Nevertheless, it is also visible that the segments containing the shadows of elephants do not have bigger values of information gain when the object model is recovered using diffusion distance instead of l_2 distance. Furthermore, the majority of the background segments of these two illustrations containing either green or water areas has the similar values of the information gain. However, a part of the background segments containing the areas of the ground which is illustrated in the fingerprint candidate rankings placing in the second row looks lighter greys than the ones illustrated in the third row. Consequently, diffusion distance is evaluated slightly better than l_2 distance in order to distinguish textural information of the image segments depending on the selected experimental setup because the colors of the ground and elephant areas are similar while their textures are different. Since the fingerprint candidate rankings of green and water areas are similar for both diffusion distance and l_2 distance, the qualities of their color discrimination is evaluated similar.

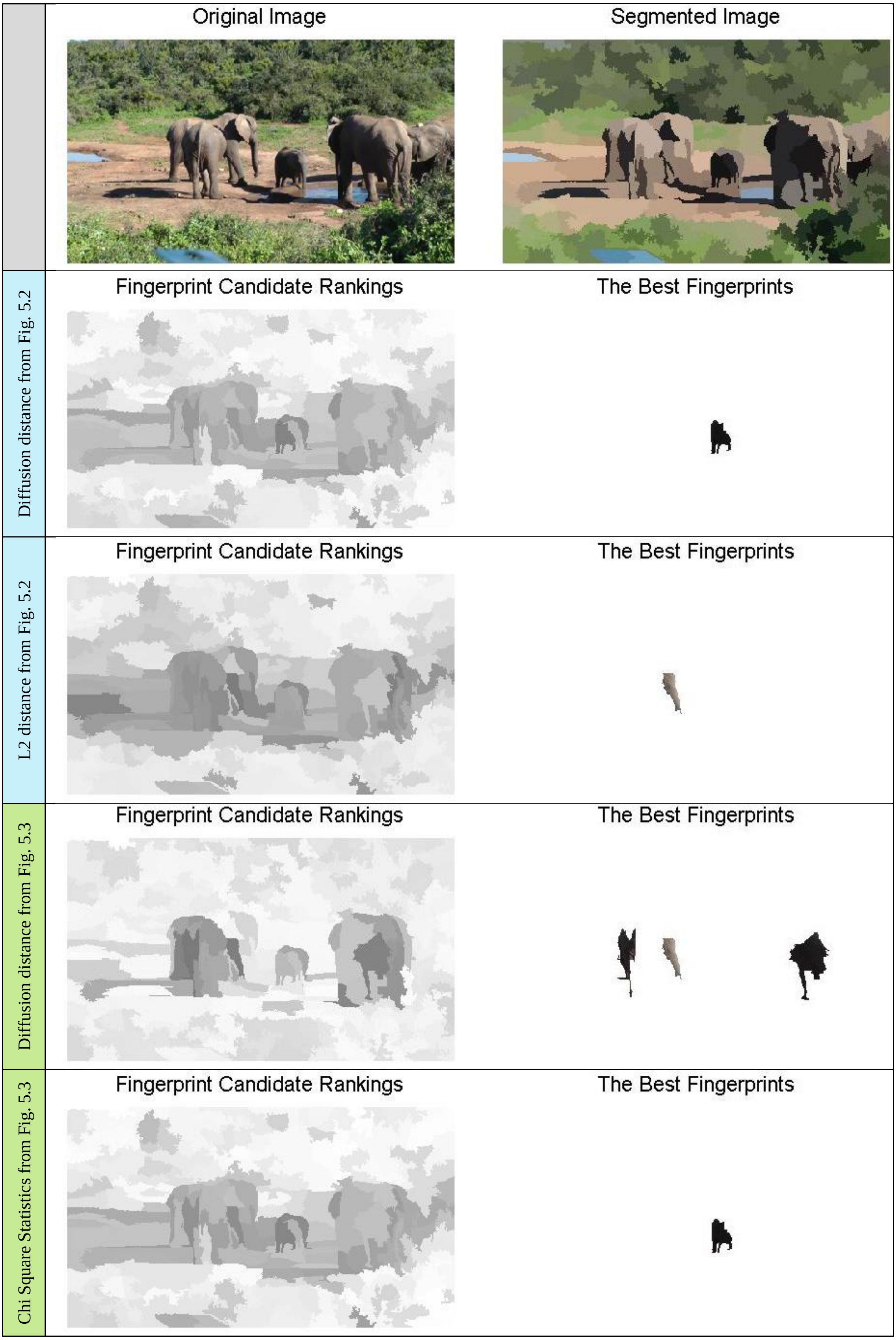

Original Image
Segmented Image
Diffusion distance from Fig. 5.2
Fingerprint Candidate Rankings
The Best Fingerprints
L2 distance from Fig. 5.2
Fingerprint Candidate Rankings
The Best Fingerprints
Diffusion distance from Fig. 5.3
Fingerprint Candidate Rankings
The Best Fingerprints
Chi Square Statistics from Fig. 5.3
Fingerprint Candidate Rankings
The Best Fingerprints

Figure 5.6: The recovery processes of 4 different object models based on the first sample training image are visualized.

It is seen that the fingerprint candidate rankings represented in the fourth row of Figure 5.6 have the most different grey color values from the ones pertaining to all the representations of the fingerprint candidate rankings. For instance, many background segments are represented with lighter grey colors in the image of fingerprint candidate rankings placing in the fourth row than the ones represented in other three images of fingerprint candidate rankings. However, some of the foreground segments are also set to the lower rankings in the fourth row than other rows. In a nutshell, the model presented in the fourth row of Figure 5.6 has the best settings in order to eliminate background segments in all the models represented in Figure 5.6. On the other hand, this model discriminates the foreground segments from the background segments with a sharper manner than all the other models. Therefore, some foreground segments are set to very low rankings in the fourth row of Figure 5.6. In addition, three segments are gathered as fingerprints demonstrated in the respective image of the best fingerprints by this model which contains 50 fingerprints from the complete set of training images.

Figure 5.7 shows the illustrations of the same models presented in Figure 5.6 which are captured from another sample training image by the visualization tool. It is seen in Figure 5.7 that all the observations explained above with respect to the fingerprint candidate rankings and the best fingerprints represented in Figure 5.6 are valid for the ones represented in Figure 5.7 as well. Consequently, this validation proofs that the qualifications of these object models with respect to object model recovery are stable for all the training images.

It is shown in Figure 5.7 that there are two selected segments as fingerprints for both of the models presented in the second and the last row of Figure 5.7 and they are the same segments. Moreover, the fingerprints gathered for these both models are dark areas of elephants while the fingerprint represented in the third row of Figure 5.7 is a light area of an elephant. The discrimination of background segments from the foreground segments is likewise sharper by the model presented in the fourth row of Figure 5.7. The texture recognition capabilities of the models presented in the second and the last row of both figures are also observable for the sample training image presented in Figure 5.7. For instance, these two models eliminate the segments containing the ground areas slightly better than the one demonstrated in the last row of Figure 5.7. This means that the difference between the ground and elephant segments are emphasized slightly better by one of these two models. One should note that the pixel colors pertaining to the majority of the elephants appearing in the training dataset is brownish and the dominant color of the ground areas is brown shown in both Figure 5.6 and Figure 5.7.

The same models receive different numbers of obtained fingerprints depending on the sample training images presented in Figure 5.6 and in Figure 5.7. Hence, this shows that there is no restriction for the models to select a specific number of fingerprints from each training image.

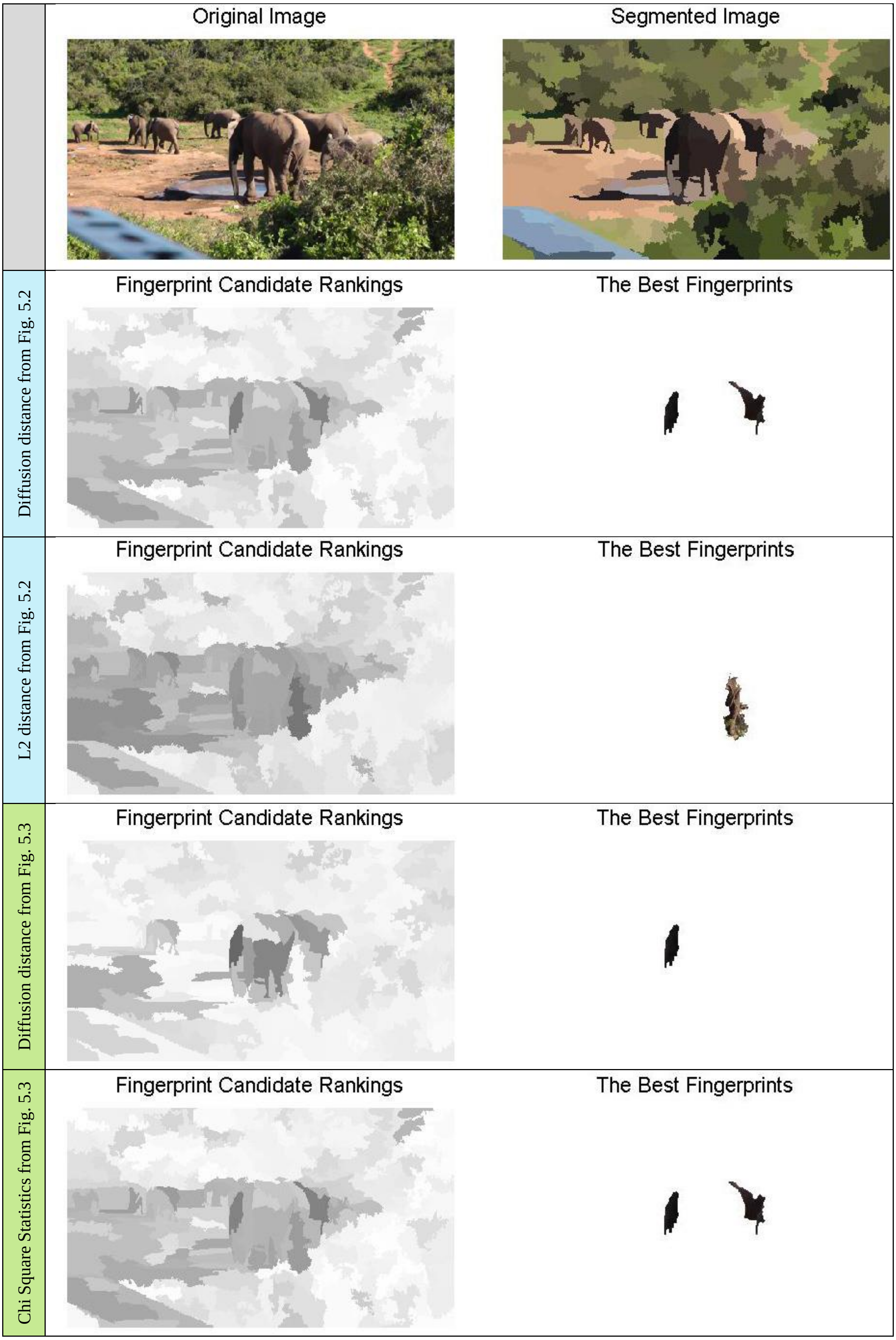

Original Image
Segmented Image
Diffusion distance from Fig. 5.2
Fingerprint Candidate Rankings
The Best Fingerprints
L2 distance from Fig. 5.2
Fingerprint Candidate Rankings
The Best Fingerprints
Diffusion distance from Fig. 5.3
Fingerprint Candidate Rankings
The Best Fingerprints
Chi Square Statistics from Fig. 5.3
Fingerprint Candidate Rankings
The Best Fingerprints

Figure 5.7: The recovery processes of 4 different object models based on the second sample training image are visualized.

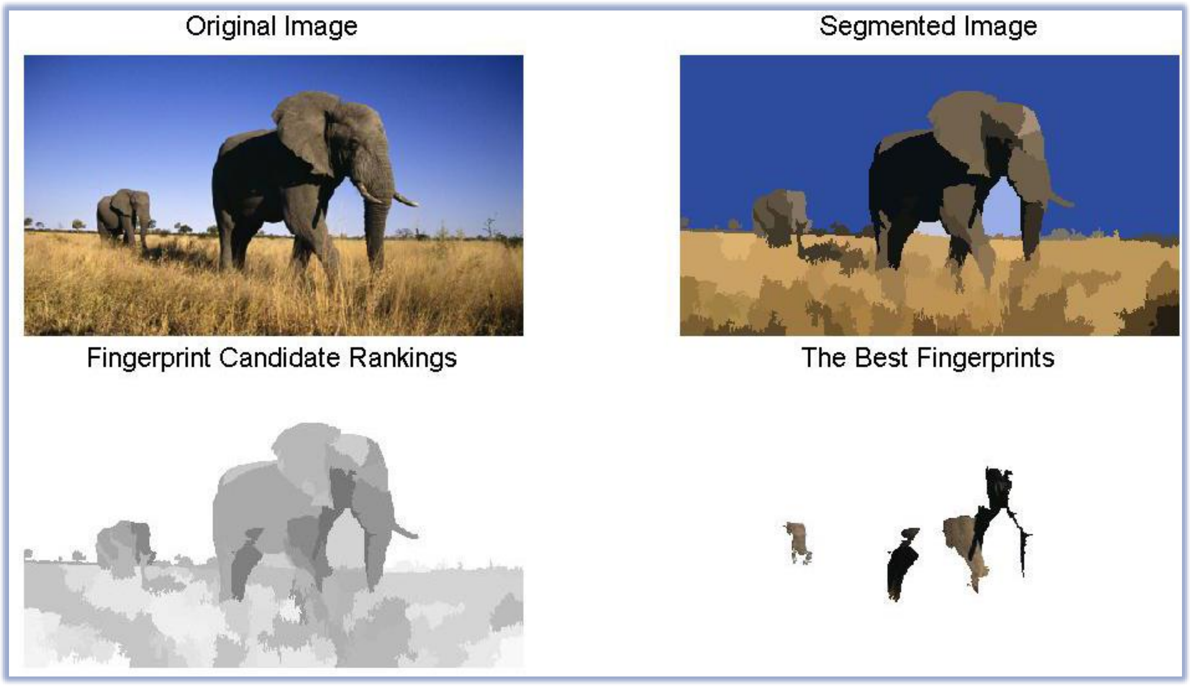


Figure 5.8: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the violet curve presented in Figure 5.2.

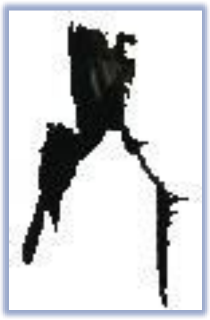


Figure 5.9: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.8.

Figure 5.8 visualizes the recovery processes belonging to the object model built by using diffusion distance with respect to a sample training image, which is different than the ones used in the last two previous figures. This object model is exactly the same object model represented in the second rows of both of these previous figures for two different sample training images. Therefore, the object model recovered in Figure 5.8 also stores 50 fingerprints. This sample training image is selected to be visualized because it possesses a segment keeping the highest information gain value from the ones gathered from all the segments in the entire training dataset based on the foregoing object model. Three segments shown in the best fingerprints part of Figure 5.7 are obtained from the training image presented in the original image part of Figure 5.7 as three fingerprints of the foregoing object model keeping 50 fingerprints in total. The obtained segment on the rightest is the one owning the highest information gain value comparing to all the other segments placing in the training dataset and its enlarged version is illustrated in Figure 5.9. As seen in Figure 5.9, the segment selected as a best fingerprint from the complete training dataset based on its information gain value also contains a dark area of an elephant similar to other fingerprints with respect to the same object model demonstrated in Figure 5.6 and Figure 5.7. In Figure 5.8, the pixel colors of the segments containing the sky are set to pure white since their information gain values are computed as 0 during the learning process of the considered object model. Moreover, the dark ground segments have slightly bigger information gain values than the light ground segments due to the similarities between the colors of the dark ground segments and the elephant segments; however, the segments containing the parts of elephants are still assigned to the bigger values of information gain than the ones belonging to the dark ground segments according to the large texture differences and the moderate color differences.

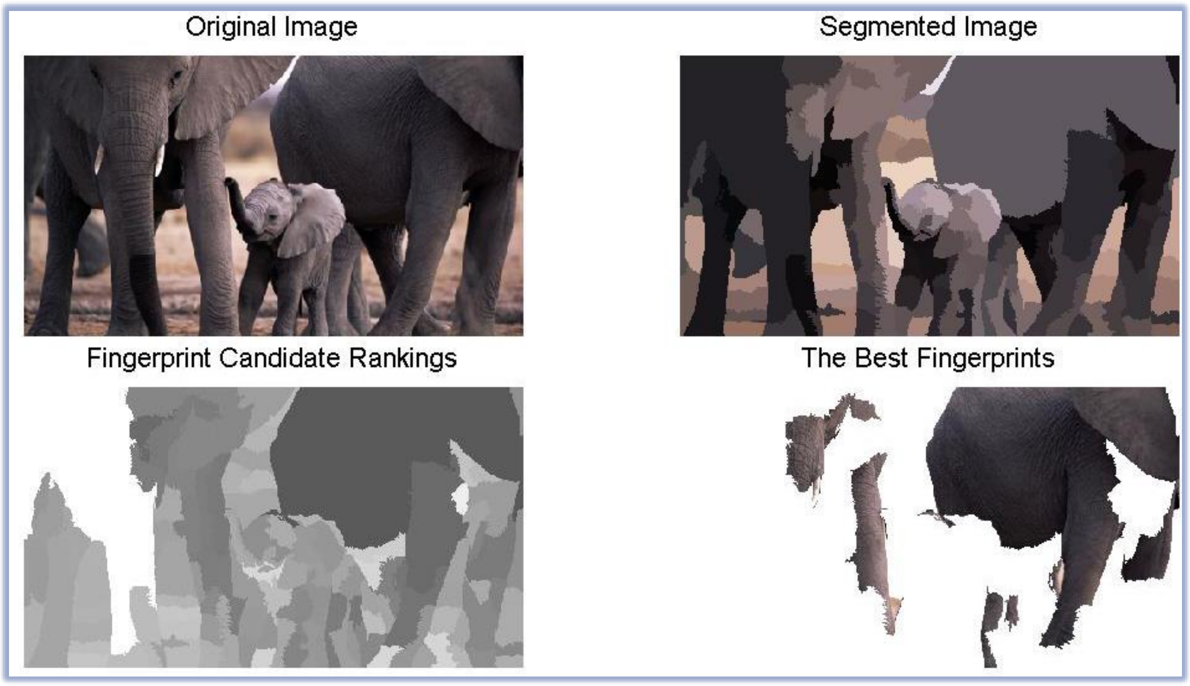


Figure 5.10: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the yellow curve presented in Figure 5.2.

The left upper image presented in Figure 5.10 is also used during the building process of the same object model presented in the third rows of both Figure 5.6 and 5.7 regarding different training images and owns a segment holding the highest information gain value by taking into account the complete training dataset. The segment keeping the biggest information gain value places at the upper rightmost of the best fingerprints image shown in Figure 5.10 and its magnified form is shown in Figure 5.11. The best fingerprint shown in Figure 5.11 is also not a dark elephant segment like the fingerprints selected from the training images of Figure 5.6 and Figure 5.7 by the same object model.

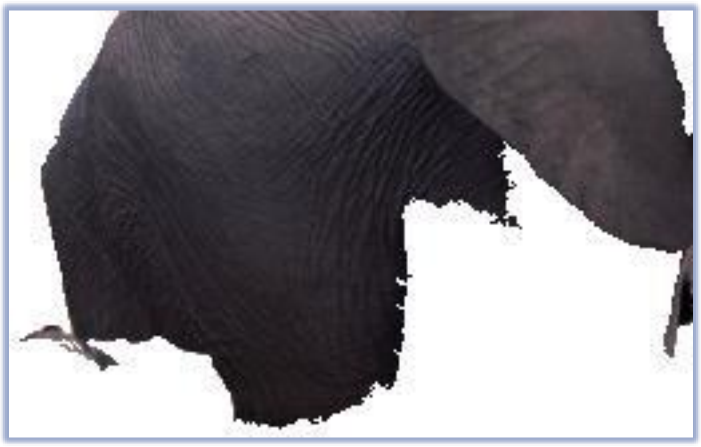


Figure 5.11: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.10.

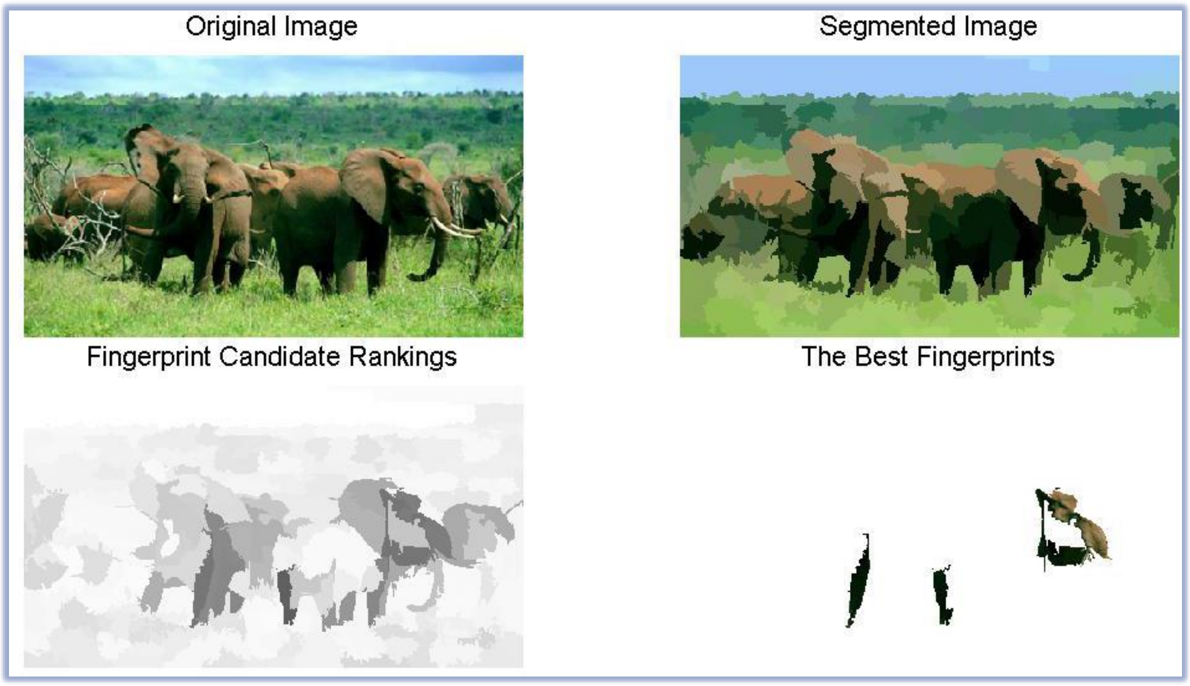


Figure 5.12: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the violet curve presented in Figure 5.3.

The object model recovery regarding two different training images presented in the fourth rows of Figure 5.6 and Figure 5.7 produces the results visualized in Figure 5.12 for the training image presented in the left upper part of Figure 5.12. This training image contains a segment keeping the highest information gain value from all the fingerprints collected in this considered object model built by using diffusion distance. As seen in Figure 5.12, the majority of the segments consisting of the sky has the lowest information gain value so they are demonstrated with pure white color. Moreover, the elimination of green segments by setting lower information gain values than the ones belonging to the segments containing the areas of the elephants is observable in the image of fingerprint candidate rankings presented in Figure 5.12. Although some parts of elephants seem greenish, they keep higher information gain values than the ones related to the segments extracted from forest regions. The segment placing in the middle of all the segments shown in the lower rightmost image of Figure 5.12 keeps the top information gain value as a selected fingerprint for the foregoing object model. In addition, Figure 5.13 presents an enlarged form of this fingerprint.

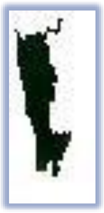


Figure 5.13: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.12.

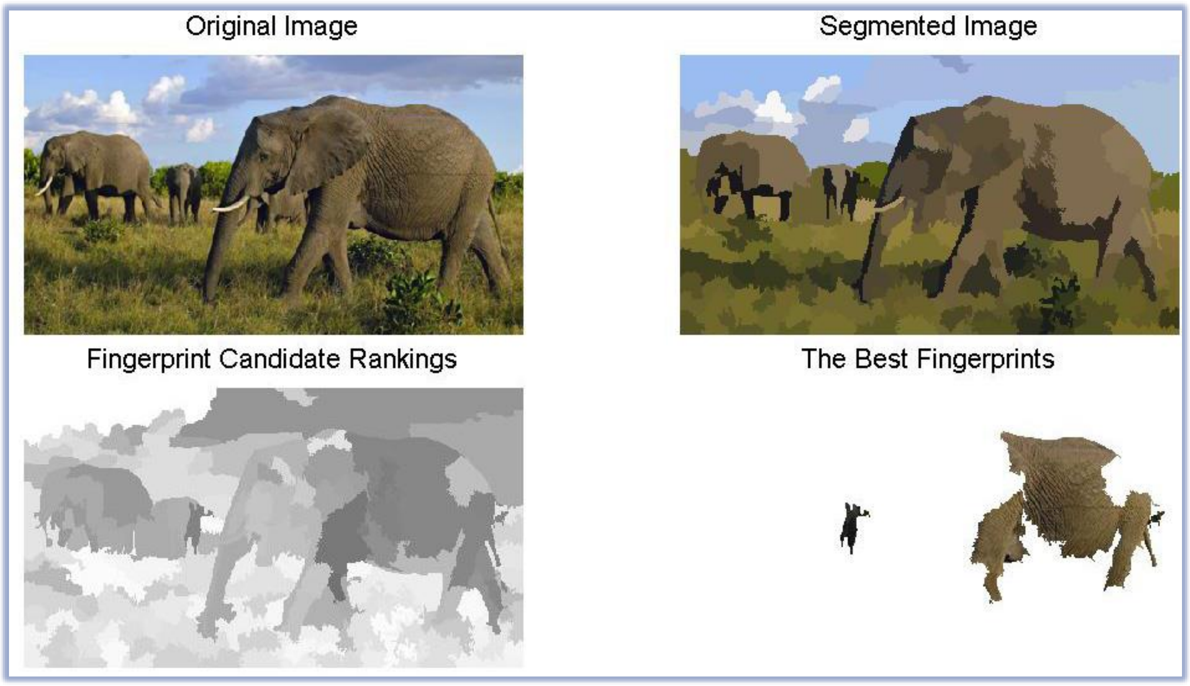


Figure 5.14: The visualization of segmentation, fingerprint candidate rankings and the selected fingerprints for the sample training image containing the fingerprint with the highest information gain value from all the training dataset based on the recovery of the object model whose precision score places on the node defining 50 fingerprints of the green curve presented in Figure 5.3.

Figure 5.14 shows the training image supplying the best fingerprint from the complete training dataset to the recovery of the object model which is also presented regarding different training images in the last rows of both Figure 5.6 and Figure 5.7. Therefore, this object model is built by using chi square statistics. All of the selected segments presented in the lower rightmost image of Figure 5.14 are true positives same as all other segments presented in the previous figures regarding other object models. The enlarged form of the segment keeping top information gain value for the object model used for the representation of Figure 5.14 is shown in Figure 5.15.

Figure 5.15: The enlarged version of the top fingerprint (keeping the highest information gain value) presented as one of the selected best fingerprints for the respective object model in Figure 5.14.

The sets of object models which are recovered based on VLAD descriptors by using the mean shift segments collected from CIELAB color space or the segments obtained from sliding windows method with occlusions mostly evaluated as unsatisfactory. Therefore, their results are not presented. Table 5.1 represents the highest mean precision scores of the sets of object models built by using VLAD descriptors.

Distance function	Chi square statistics from Figure 5.3	Diffusion distance from Figure 5.2	Chi square statistics from Figure 5.1	Diffusion distance from Figure 5.3	L2 distance from Figure 5.2	Kolmogorov-Smirnov test from Figure 5.4	Matching distance from Figure 5.1	L2 distance from Figure 5.3
Mean Precision	0.9029	0.8945	0.8905	0.8599	0.8612	0.8217	0.7887	0.7831

Table 5.1: The highest mean precisions obtained from VLAD-based model recovery

5.1.2 COLOR HISTOGRAMS-BASED MODEL RECOVERY

Figure 5.16 and Figure 5.17 show the precision scores obtained from the object models recovered based on GB color histograms, which are extracted from the mean shift segments. The difference between Figure 5.16 and Figure 5.17 is the color space utilized for the mean shift segmentation. The color space of the former is CIELAB while the latter is executed in CIELUV. The mean precisions obtained from the sets of object models represented in both of these figures regarding four different distance functions such as diffusion distance, l_2 distance, chi square statistics and histogram intersection distance are greater than 0.8. In addition, the precisions of all these object models apart from the ones built by using l_2 distance increase while shifting from bigger to smaller object models.

The sets of object models shown in Figure 5.16 and Figure 5.17 and built by using Kolmogorov-Smirnov test have the mean precision scores, which are greater than 0.65 and less than 0.75. Therefore, these sets of object models are evaluated as satisfactory in order to be utilized for the construction of a target visual object detector. Performing mean shift segmentation on CIELUV or CIELAB does not cause any significant change in the precisions of the object models built by using GB color histograms. Consequently, since the object models built by using GB color histograms and one of five mentioned distance functions provide at least closely satisfactory mean precision scores, their respective sets are also evaluated as qualitative results like the sets of object models, which are highlighted in the previous figures based on their mean precision scores and their curves having desirable shapes.

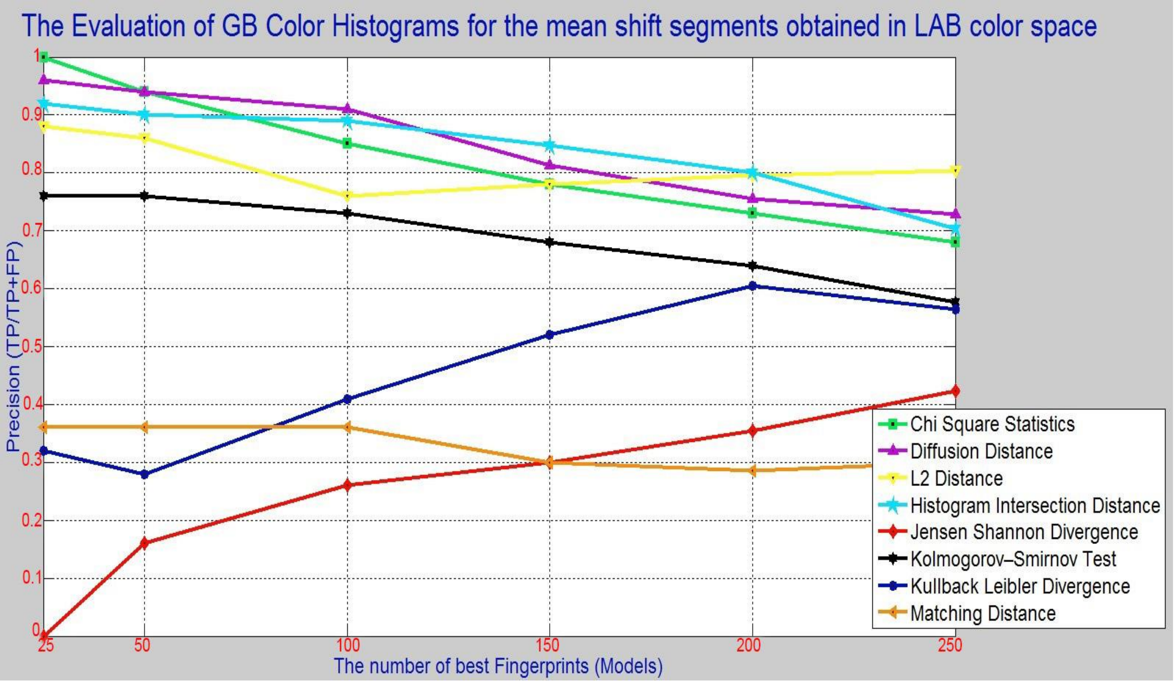


Figure 5.16: The representation of the precision scores received from the object models recovered by using GB color histograms and performing mean shift segmentation in CIELAB color space.

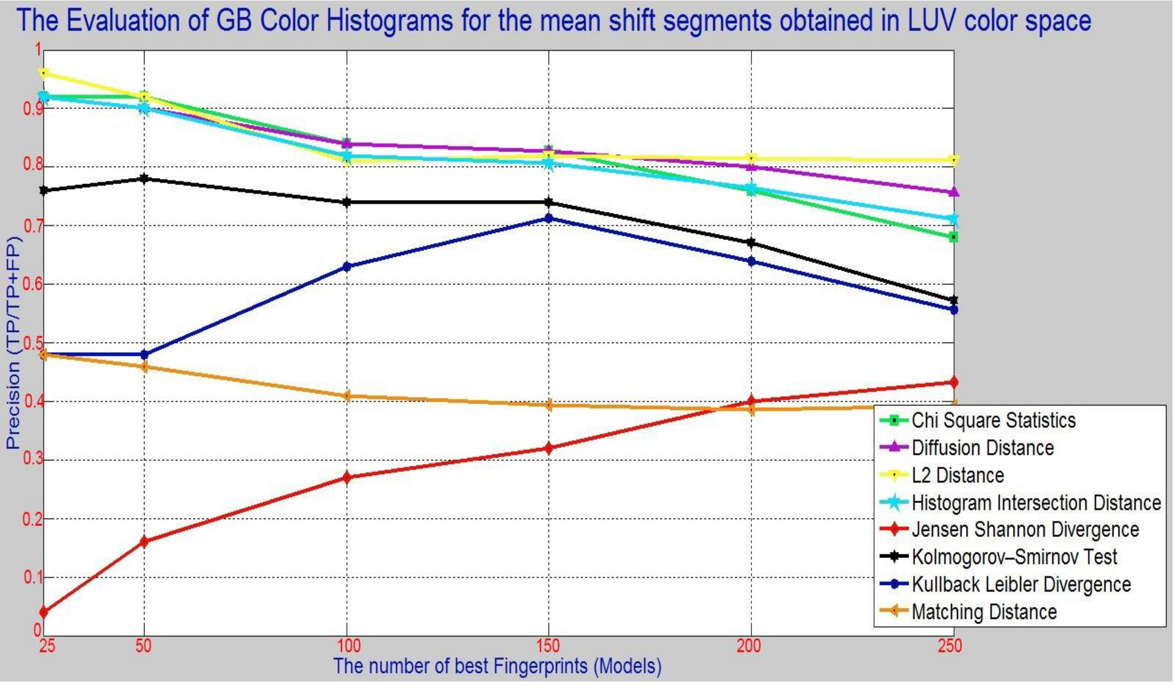


Figure 5.17: The representation of the precision scores received from the object models recovered by using GB color histograms and performing mean shift segmentation in CIELUV color space.

The set of object models recovered by using diffusion distance and presented in Figure 5.16 is evaluated as highly satisfactory since its mean precision score is slightly bigger than 0.85. Apart from that, other three sets of object models recovered by using one of the distance functions such as l_2 distance, chi square statistics and histogram intersection distance are evaluated as satisfactory because their mean precision scores are greater than 0.75 and less than 0.85.

As seen in Figure 5.17, the mean precision scores regarding the set of object models recovered by using l_2 distance is slightly bigger than 0.85; therefore, it is evaluated as highly satisfactory. Moreover, the rest of the distance functions recovering the set of object models receiving the mean precision scores being greater than 0.8 as described previously, is evaluated as a set of satisfactory distance functions for the recovery of the object model sets based on the experimental configurations used in Figure 5.17.

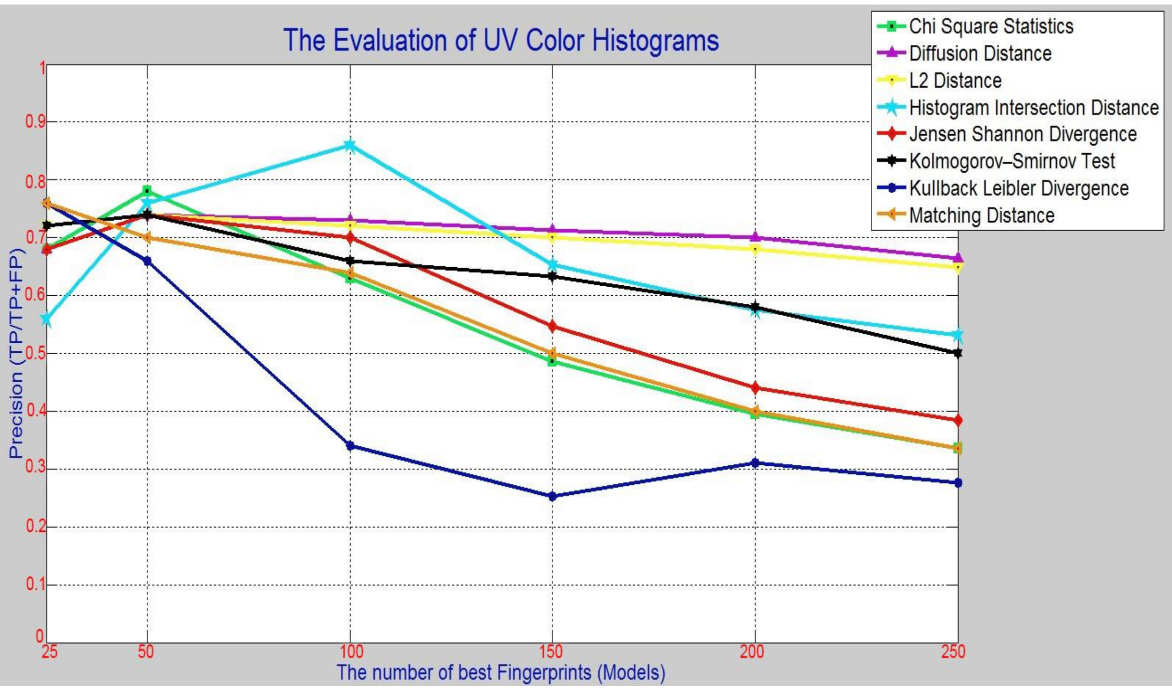


Figure 5.18: The representation of the precision scores received from the object models recovered by using UV color histograms.

The precision scores presented in Figure 5.18 pertaining to object models recovered based on the experimental setup containing UV color histograms as visual descriptors of the segments. The only curve having the desirable tilt directions for its line segments shown in Figure 5.18 is the one referring to the precision scores of the object models built by using matching distance. However, the mean precision score received from this set of object models is lower than 0.65 so it is evaluated as unsatisfactory. Moreover, the tilt directions of the line segments placing in the curve regarding histogram intersection distance are undesirable especially between the object models keeping 25 and 100 fingerprints. Nevertheless, the mean precision score belonging to this set is slightly bigger than 0.65 so it is evaluated as closely satisfactory. Apart from that, the mean precision scores of two sets with respect to l_2 distance and diffusion distance are very close to each other and both are in the range of closely satisfactory as well. The shapes of their curves seem desirable apart from the object models storing 25 fingerprints. Due to having three sets of object models evaluated as closely satisfactory based on the foregoing experimental setup, Figure 5.18 is also selected as a figure containing qualitative results. The sets of object models which are recovered based on GB color histograms by using the segments obtained from sliding windows methods mostly evaluated as unsatisfactory. Therefore, their results are not presented.

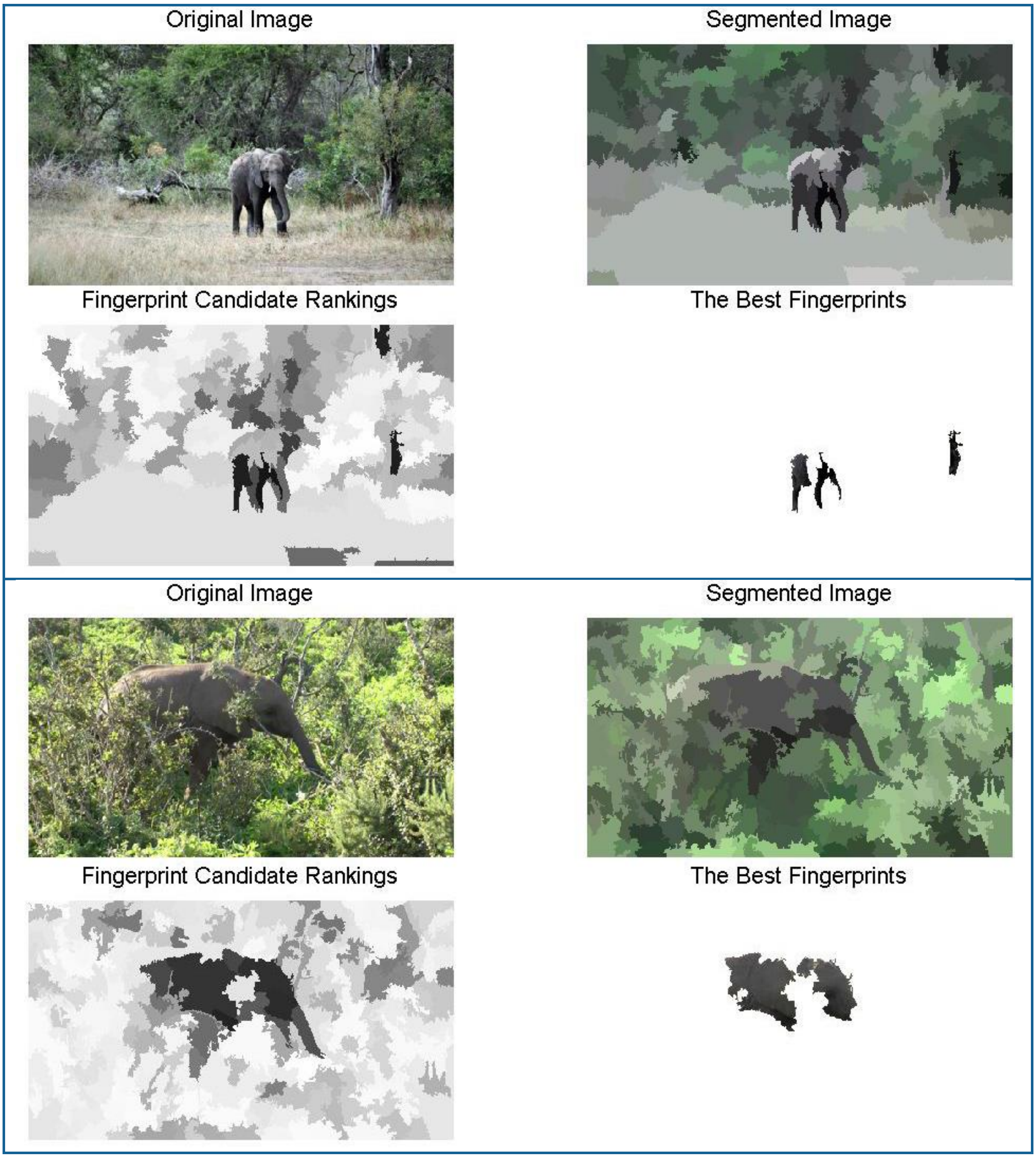


Figure 5.19: The recovery process of an object model based on two different sample training images by using UV color histograms and diffusion distance is visualized.

The recovery of an object model storing 50 fingerprints which is built by utilizing UV color histograms and diffusion distance function is visualized with respect to two different training images in Figure 5.19. These two training images are purposely selected in order to visualize the basic drawback of UV color histograms for the recovery of an object. As seen in

Figure 5.19, although the majority of the elephant segments is set to high fingerprint candidate rankings for both training images, some of segments containing either branches or trunks of tree are also set to high ranking values mostly due to their color similarities with elephant segments. Moreover, the majority of the segments consisting of the green areas has lower rankings than other segments. As a result of this, the root cause of the difference between the mean precision scores obtained from the sets of objects presented in Figure 5.18 and the others shown in the previous figures of this chapter keeping greater mean precision scores with respect to the respective distance functions is the weaker texture recognition capability of UV color histograms. The sets of object models which are recovered based on UV color histograms by using mean shift segments collected from CIELAB color space or segments obtained from sliding windows methods mostly evaluated as unsatisfactory. Therefore, their results are not presented.

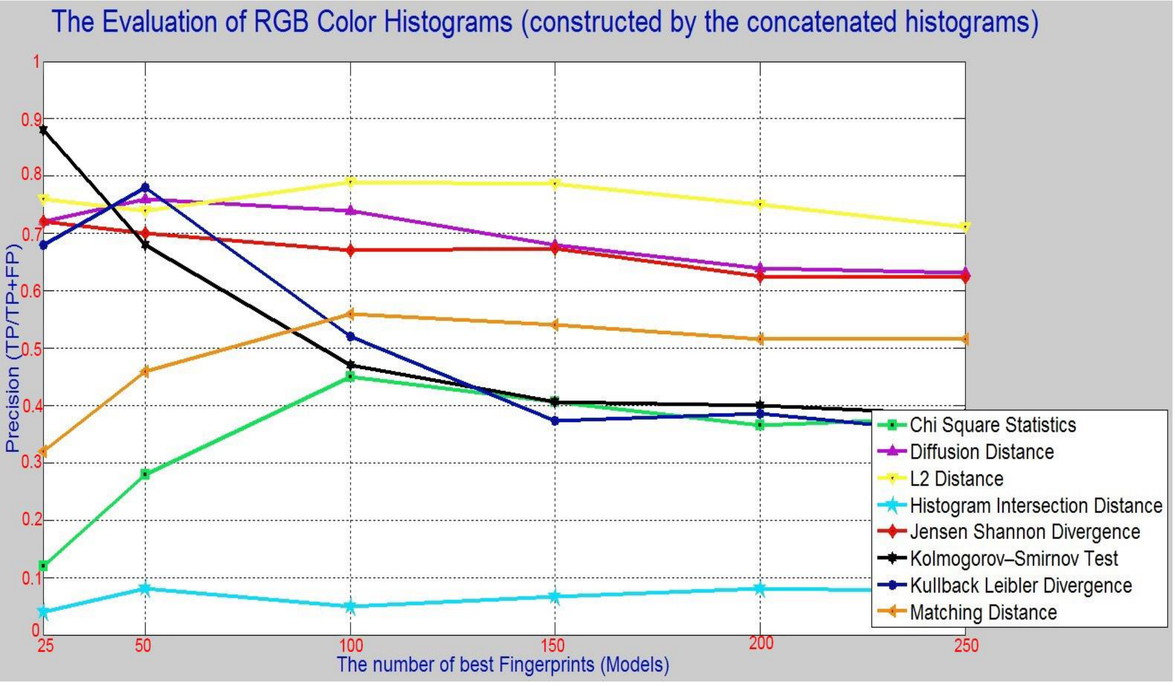


Figure 5.20: The representation of the precision scores received from the object models recovered by utilizing 1d RGB color histograms whose construction is based on the concatenation of each 2d histogram of RG, GB and BR channels into a single dimensional vector.

Figure 5.20 presents the precision scores received from the object models built by using single dimensional column vectors which are constructed based on the third type of RGB color histograms introduced in Subsection 3.2.1.4. Same as UV color histograms, there are three sets of object models constructed based on this type of RGB color histograms receiving mean precision scores which pass 0.65. One of them is recovered by using l_2 distance function and obtains 0.7564 as mean precision score. Therefore, this set is evaluated as closely satisfactory. Other two sets receive mean precision scores in the range [0.65, 0.75) so they are both evaluated as satisfactory. As presented in Figure 5.20, one of these sets receiving satisfactory mean precision score recovered by using diffusion distance while another set recovered based on Jensen Shannon divergence. The shape of the curve defining Jensen Shannon divergence is the more desirable than the ones pertaining to other two sets although its mean precision score is the lower than theirs. The precision scores collected from the object models recovered based on other two approaches of RGB color histograms are described and illustrated in Appendix A.2. The sets of object models which are recovered based on RGB color histograms by using mean shift segments collected from CIELUV color space or segments obtained from sliding windows methods mostly evaluated as unsatisfactory. Therefore, their results are not presented.

The curves presented in Figure 5.21 traverse the precision scores of object models built based on the similarities of AB color histograms computed by same distance function. There is a single set of object models presented in this figure having mean precision score is in the range [0.65, 0.75). This set is recovered by using l_2 distance and it is evaluated as closely satisfactory based on its mean precision score. However, the curve referring to this set has undesired shifts between the object models keeping 25 and 100. The rest of the sets of object models recovered based on the experimental setup consisting of AB color histograms is evaluated as unsatisfactory since their mean precision scores are lower than 0.65. Moreover, the majority of the sets presented in Figure 5.21 receives lower mean precision scores than ones presented in Figure 5.18. Therefore, utilizing UV color histograms rather than AB color histograms provides slightly better discrimination of foreground segments from background segments depending on information gain. The shapes of the curves defining distance functions of l_2 and diffusion shown in Figure 5.21 are very similar as well as the locations of their nodes specifying the precision scores of the respective object models. This similarity is also visible in Figure 5.2, Figure 5.3, Figure 5.17 and Figure 5.18. From this point of view, it can be inferred that the behaviors of diffusion or l_2 distance functions are very similar to each other in the feature spaces constructed during the building processes of the foregoing object models.

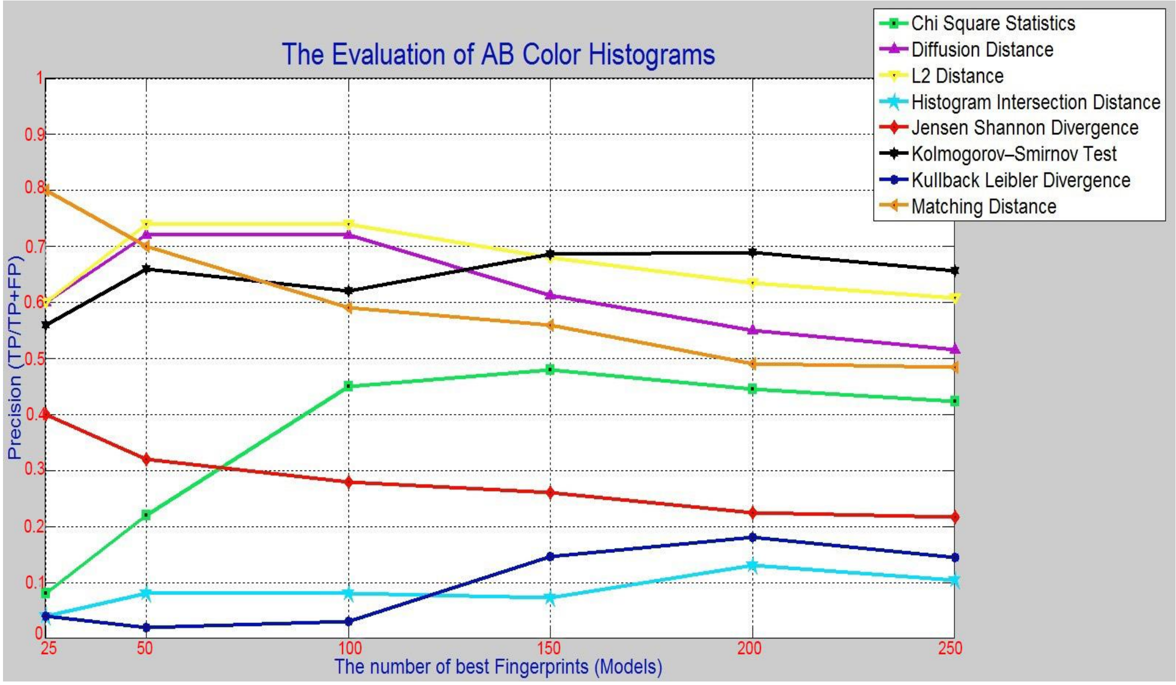


Figure 5.21: The representation of the precision scores received from the object models recovered by using AB color histograms.

The sets of object models recovered based on AB color histograms mostly evaluated as unsatisfactory like the ones collected based on RGB color histograms-based object model recovery when the mean shift segments collected from CIELUV color space or the segments obtained from sliding windows methods. Therefore, their respective results are not presented. Table 5.2 represents the highest mean precision scores of the sets of object models built by using Color Histograms.

Distance function	L2 distance from Figure 5.17	Diffusion distance from Figure 5.16	Histogram intersection distance from Figure 5.16	Diffusion distance from Figure 5.17	Chi square statistics from Figure 5.16	Chi square statistics from Figure 5.17	Histogram intersection distance from Figure 5.17	L2 distance from Figure 5.16
Mean Precision	0.8562	0.8511	0.8434	0.8404	0.8300	0.8244	0.8206	0.8132

Table 5.2: The highest mean precisions obtained from Color Histograms-based model recovery

5.1.3 CK-1-BASED MODEL RECOVERY

Figure 5.22 illustrates the precision scores obtained from the object models recovered by using CK-1 distance. As seen in this figure, there is not any set of object models having mean precision score which reaches or exceeds the boundary of 0.65; therefore, all the sets recovered based on the experimental setups containing the employment of CK-1 distance are evaluated as unsatisfactory sets. The violet and dark blue curves demonstrated in Figure 5.22 fluctuate up and down more often than green one since the segments obtained by performing mean shift segmentation contain less background data than the ones obtained by performing sliding windows. Moreover, the tilt directions of the green curve are the desirable ones apart from only the line segment joining the object models storing 25 and 50 fingerprints.

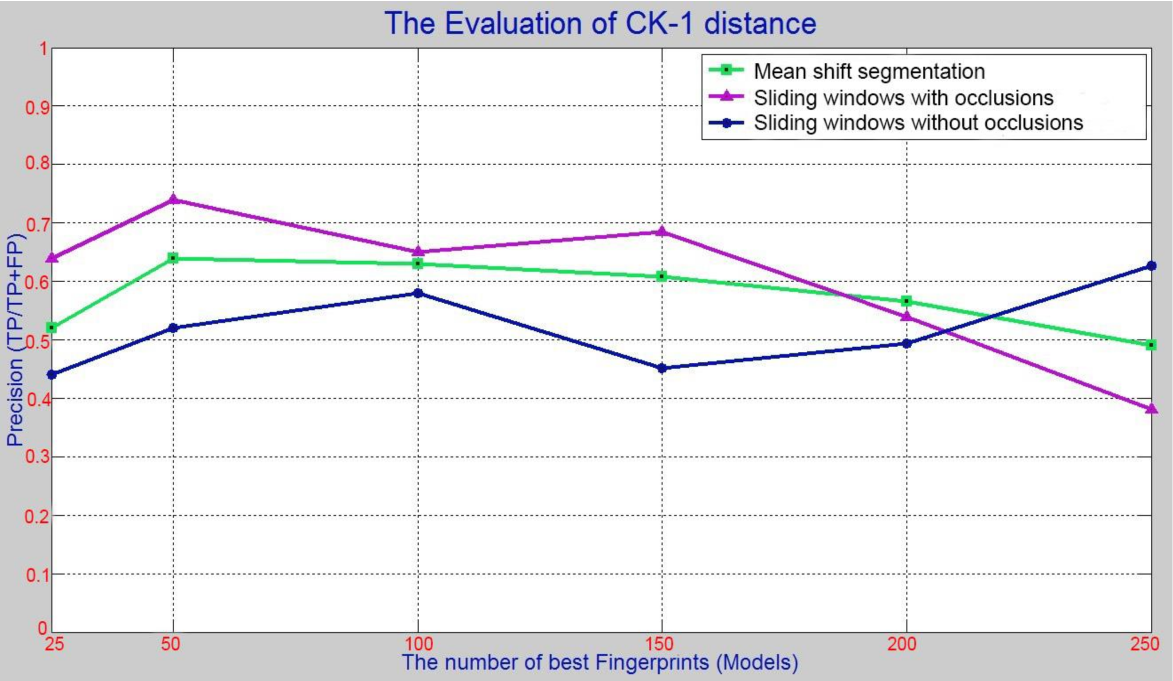
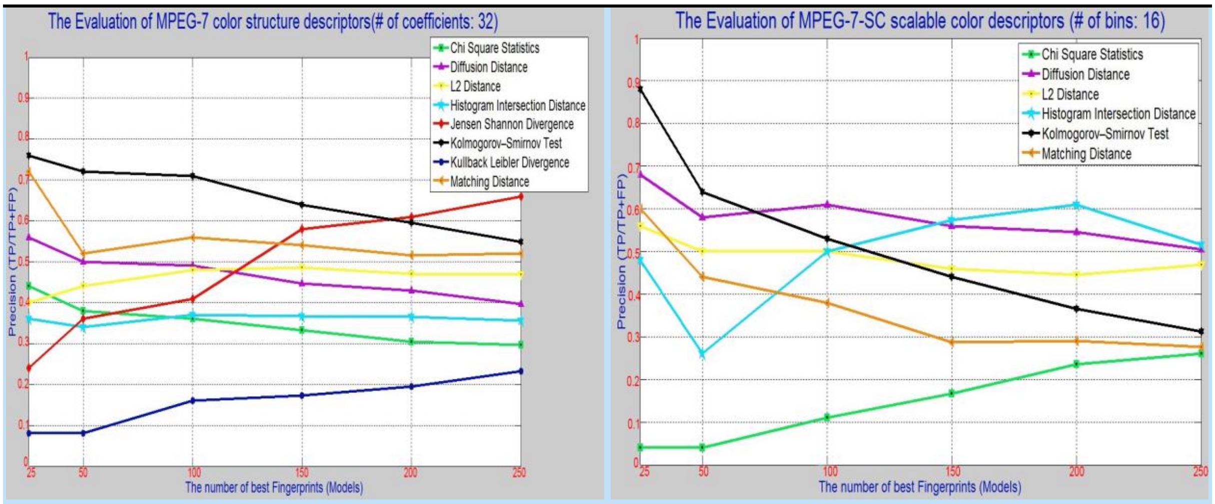


Figure 5.22: The representation of the precision scores received from the object models built by using CK-1 distance measure.

The sets of object models collected by CK-1-based model recovery and using the segments obtained from sliding windows method with occlusions mostly evaluated as unsatisfactory. Hence, their results are not shown.

5.1.4 MPEG-7 COLOR DESCRIPTORS-BASED MODEL RECOVERY

When the visual descriptors of the image segments are extracted as MPEG-7 color structure descriptors and gathered as object models based on different similarity measurements, the precision scores of all these object models are represented in (a) of Figure 5.23. Moreover, the recovery of the object models containing the process of the extraction of MPEG-7 scalable color descriptors yield the results shown in (b) of Figure 5.23. The only set of object models from all the sets presented in both subfigures of Figure 5.23 having closely satisfactory mean precision score is the one demonstrated in (a) of Figure 5.23 and recovered by using Kolmogorov-Smirnov test. All the remaining sets possess unsatisfactory mean precision scores. In addition, the curve denoting Kolmogorov-Smirnov test in (a) of Figure 5.23 has a desirable shape since the precision scores of its object models continuously increase by switching from bigger object models to smaller object models. Since Jensen Shannon divergence and Kullback Leiber divergence are not applicable to the feature space of MPEG-7 scalable color descriptors, each precision score referring to these distance functions is presented in the (b) of Figure 5.23. The computations of histogram intersection distance and Kolmogorov-Smirnov test remain very similar for the both visual feature spaces of MPEG-7 color structure descriptors and MPEG-7 scalable color descriptors.



(a) (b)

Figure 5.23: The representation of the precision scores received from the object models built by using either MPEG-7 color structure descriptors or MPEG-7 scalable color descriptors.

MPEG-7 color descriptors-based object models which are recovered by using the segments obtained from the methods of sliding windows or the mean shift segments collected from CIELAB color space mostly receive smaller precision scores than the ones presented in Figure 5.23. Therefore, their results are not presented.

5.1.5 CEDD-BASED MODEL RECOVERY

The mean precision scores collected from all the sets of object models shown in Figure 5.24 are smaller than 0.65 same as the ones recovered based on CK-1 distance. Therefore, CEDD-based model recovery is likewise evaluated as unsatisfactory. The evaluation of the precision scores of CEDD-based object models mostly worse than the ones presented in Figure 5.24 when they are recovered by using the segments obtained from the methods of sliding windows or the mean shift segments collected from CIELAB color space. Hence, , their results are not illustrated.

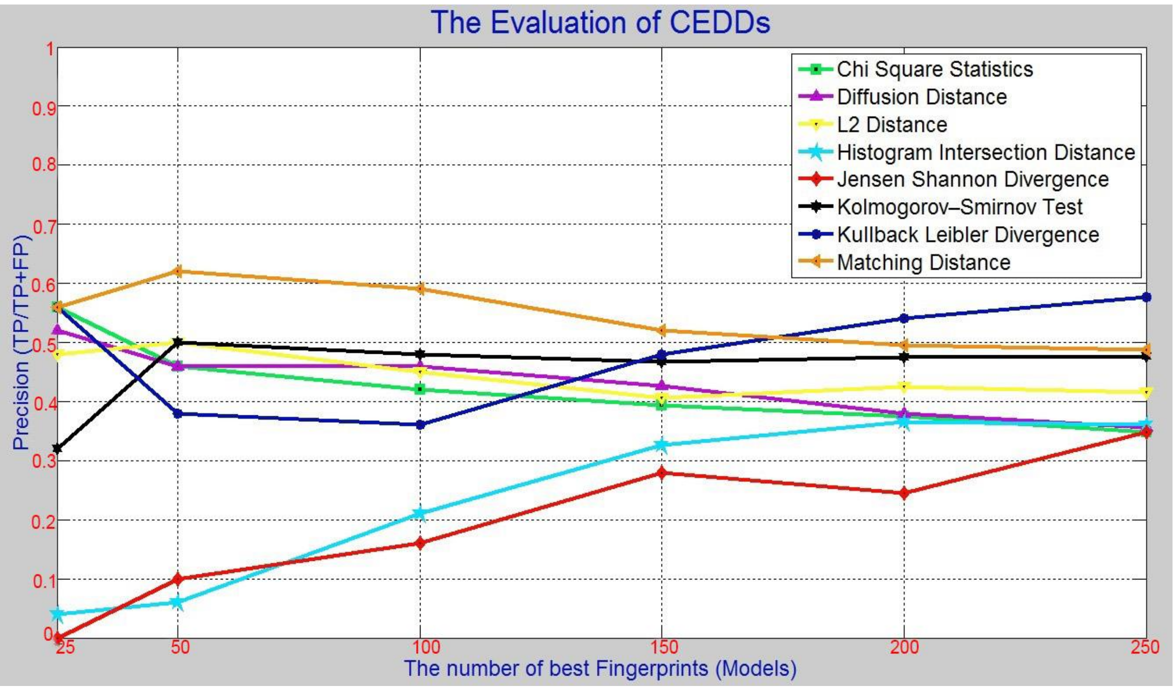
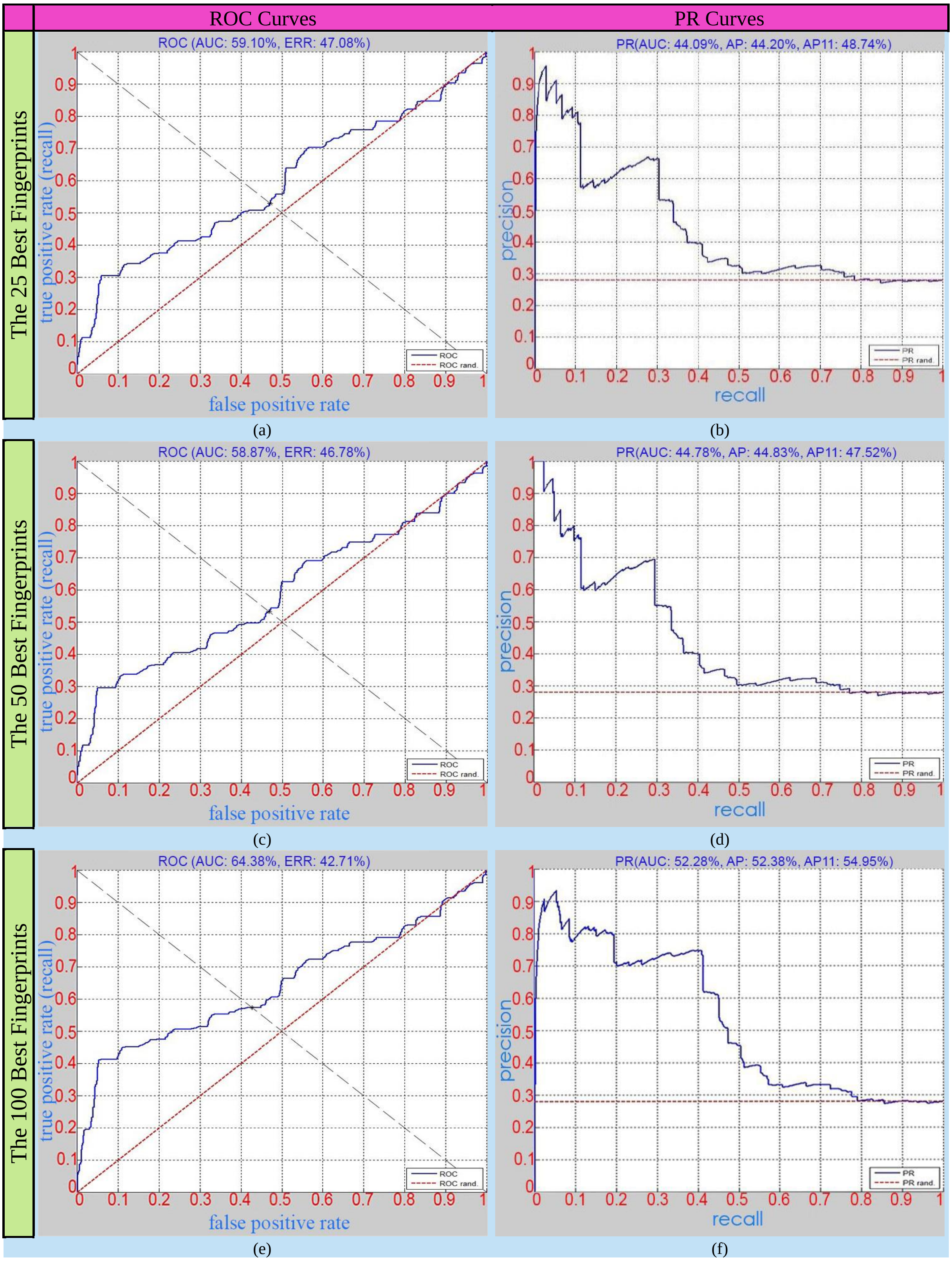


Figure 5.24: The precision scores of the object models built by using CEDDs are represented.

5.2 RESULTS OF TARGET VISUAL OBJECT DETECTION

This subsection presents the accuracies received from the classifiers existing in the scope of the experimental setups with respect to the detection of the target visual objects from the test dataset of the thesis as well as their both curves of ROC and PR. For this purpose, these different classifiers are initially trained with the most qualitative object models selected based on the mean precision scores of their respective sets and the structures of the curves traversing through them. The details of these selected object models are presented in the previous subsection 5.1. Moreover, a mean accuracy received from each classifier is computed for each set of object models. One should note that the object models are grouped into the same set if they are recovered based on exactly the same experimental setup and distance function. The only difference among the object models placing in the same set is the number of fingerprints which is stored by each of them.

5.2.1 NAÏVE BAYES-BASED OBJECT DETECTION



(a) (b)

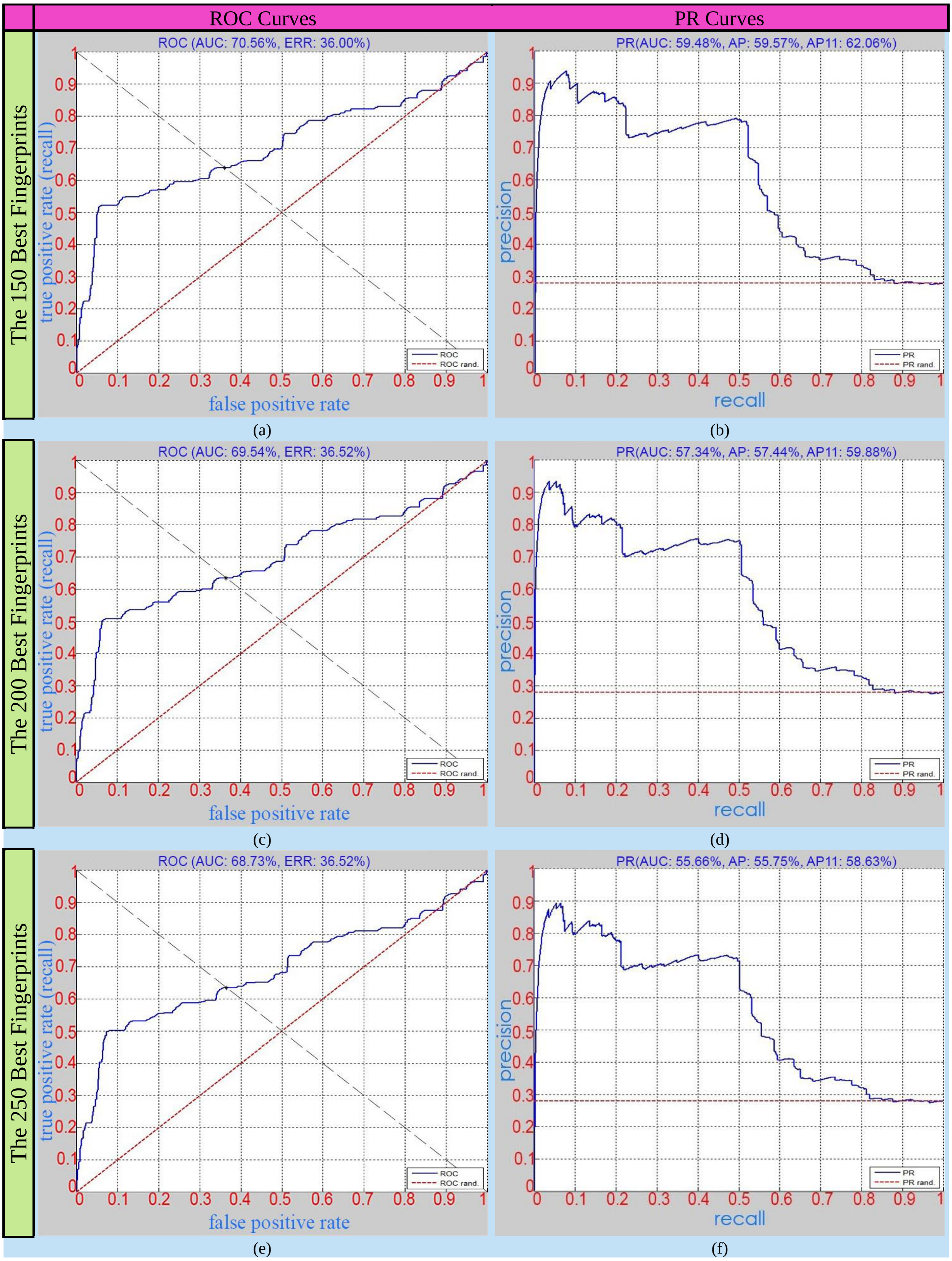
(c) (d)

(e) (f)

Figure 5.25: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

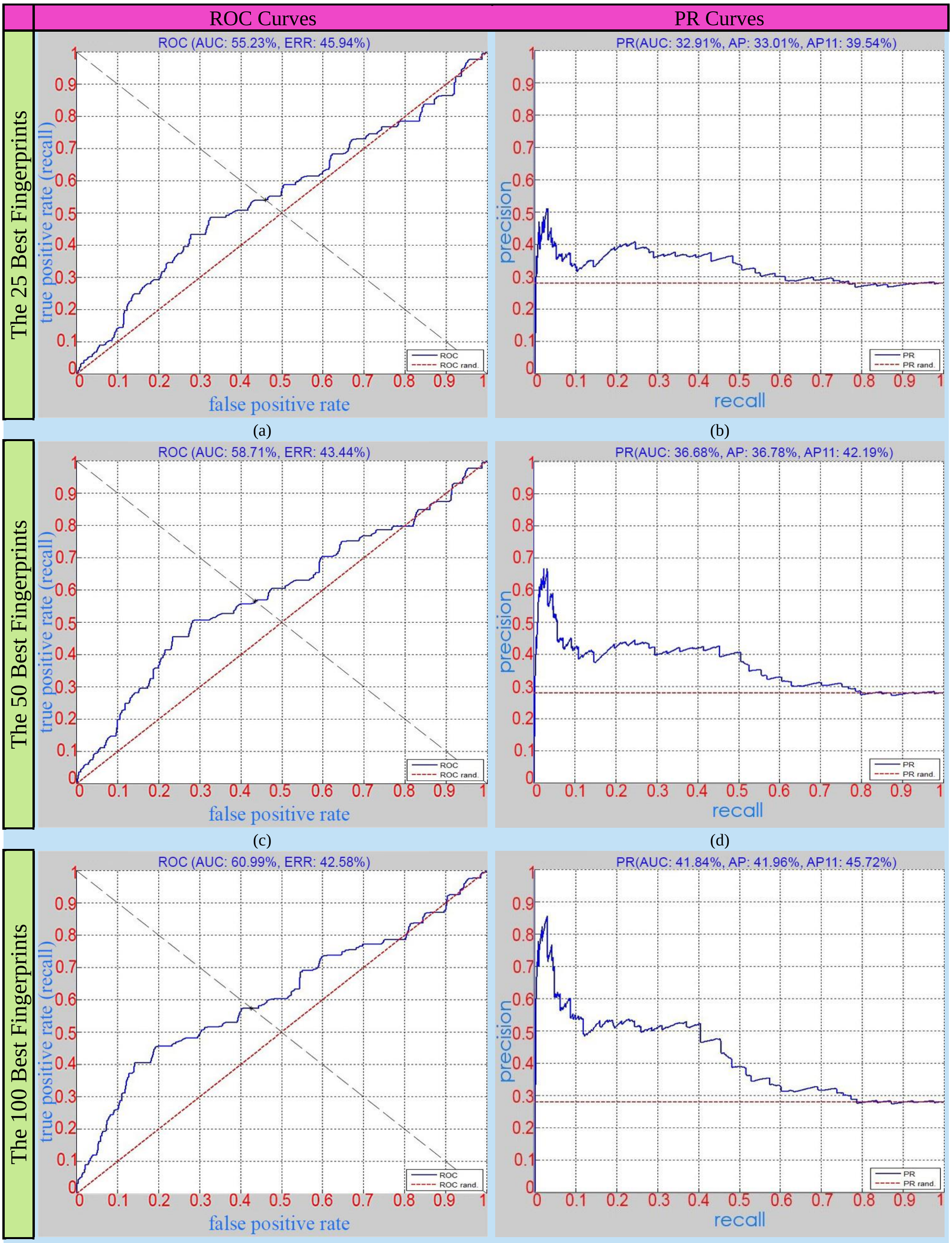
Both Figure 5.25 and Figure 5.26 present the ROC and PR curves received from Naïve Bayes classifiers which are trained by using the object models associated with the diffusion distance in Figure 5.2. Each row of the both figures belongs to an object model keeping a specific number of fingerprints; therefore, all the object models referring all the rows of these both figures utilized for the training of the Naïve Bayes classifiers are in the same set and their only difference is their sizes. The ROC and PR curves obtained from the classifiers trained by the object models storing the 25, 50 and 100 best fingerprints are shown in Figure 5.25 while the ones storing the 150,200 and 250 best fingerprints are submitted to the classifiers whose results are presented in Figure 5.26. As shown in Figure 5.25 and Figure 5.26, the areas under the ROC curves change between 58.87% and 70.56% while their equal error rates vary between 36% and 47.08%. Furthermore, the areas under PR curves presented in these figures change between 44.09% and 59.48%. According to all these results, the Naïve Bayes classifier trained by using the object model keeping the 150 best fingerprints is evaluated as the best classifier because it receives the largest areas under both ROC and PR curves as well as the smallest equal error rate. Apart from that, the highest accuracy is also obtained by the same classifier. The accuracy of this classifier is 0.8230 and the mean accuracy obtained by using the complete set of these object models is 0.7932.

The accuracies of the Naïve Bayes classifiers trained by using the object models storing the 150,200 and 250 best fingerprints from the foregoing set are greater than 0.8; therefore, these classifiers are evaluated as highly satisfactory. Furthermore, since the rest of Naïve Bayes classifiers regarding the same set of object models has accuracies in the range of [0.7 and 0.8), they are evaluated as satisfactory.



(a) (b) (c) (d) (e) (f)

Figure 5.26: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

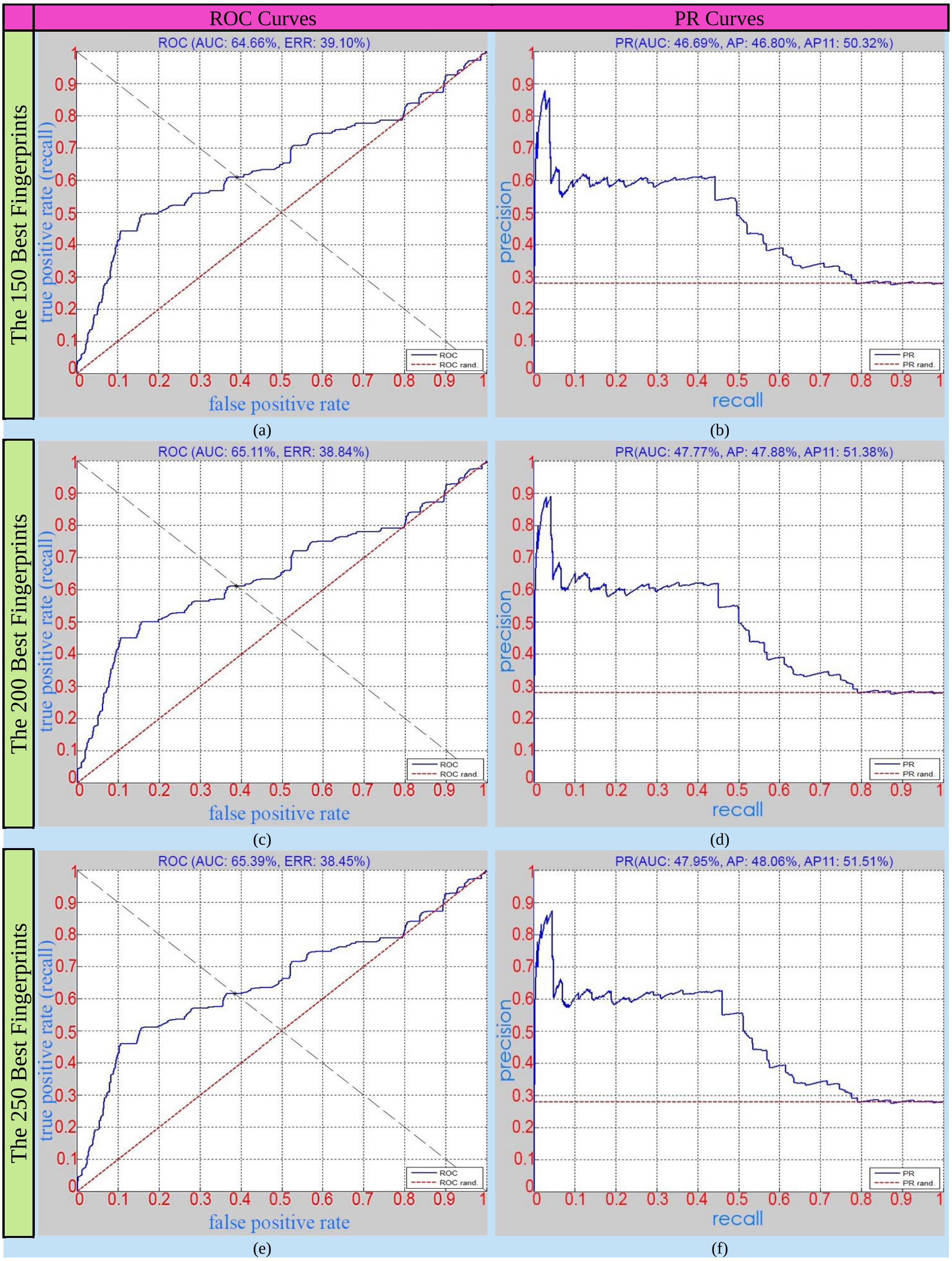


(a) (b)

(c) (d)

(e) (f)

Figure 5.27: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using chi square statistics and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each pixel from the mean shift segments.



ROC (AUC: 64.66%, ERR: 39.10%)

(a)

PR(AUC: 46.69%, AP: 46.80%, AP11: 50.32%)

(b)

ROC (AUC: 65.11%, ERR: 38.84%)

(c)

PR(AUC: 47.77%, AP: 47.88%, AP11: 51.38%)

(d)

ROC (AUC: 65.39%, ERR: 38.45%)

(e)

PR(AUC: 47.95%, AP: 48.06%, AP11: 51.51%)

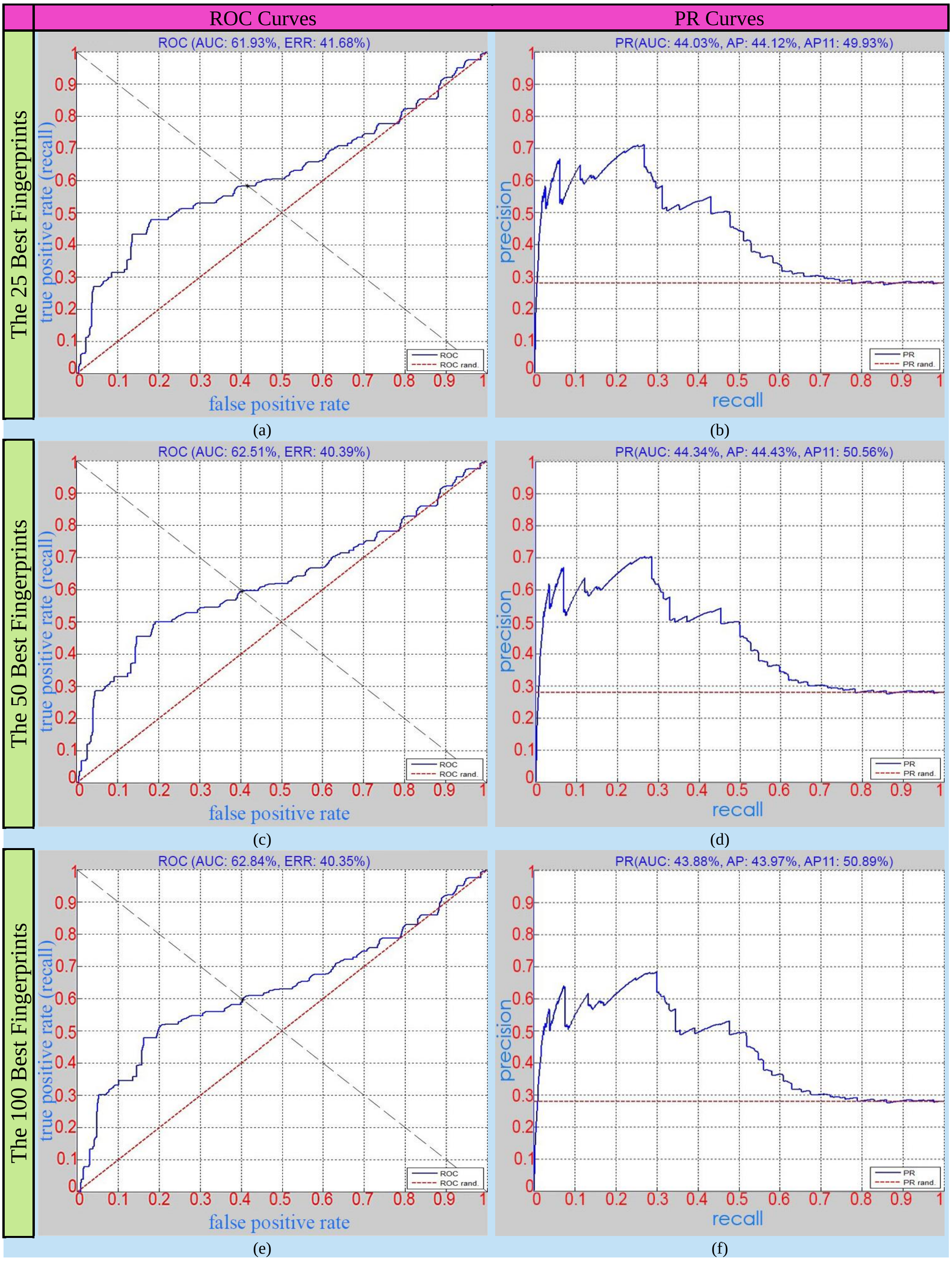
(f)

Figure 5.28: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using chi square statistics and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each pixel from the mean shift segments.

Both of the ROC and PR curves presented in Figure 5.27 and Figure 5.28 belong to Naïve Bayes classifiers trained with the set of object models shown in Figure 5.1 and recovered by using chi square statistics when they are performed on the development dataset. Same as Figure 5.25 and Figure 5.26, these figures likewise present the curves obtained based on the development dataset by using each object model of the mentioned set in a determined single row. Hence, the object models are sorted into the rows of figures in ascending order according to their sizes.

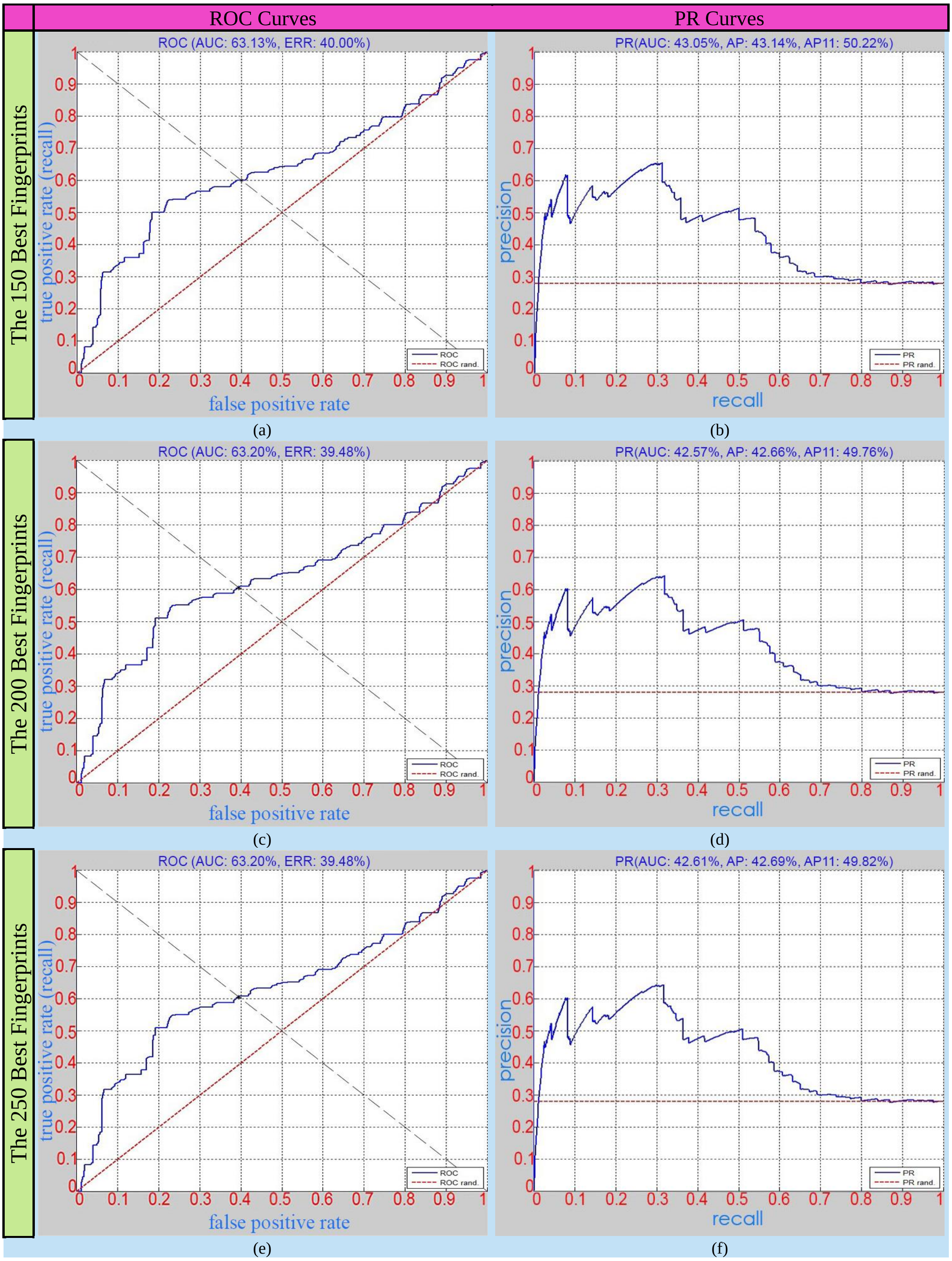
As illustrated in Figure 5.27 and Figure 5.28, when the size of the object model utilized for the training of the classifier increases, the areas under both ROC and PR curves becomes larger and the equal error rates become smaller as the results of the classifier based on the development dataset. The areas under the curves respectively change based on the size of the utilized object model in the ranges of [55.23%, 65.39%] and [32.91%, 47.95%] for ROC and PR. Furthermore, the range of equal error rates received from the classifiers based on different sizes of considered object models is [38.45%, 45.94%]. The classifier which is trained by using the object model placing in this set and keeping the 250 best fingerprints has the highest accuracy from the ones trained by using all the set of these object models and it is 0.7730. Apart from that, the mean accuracy pertaining to this set is 0.7268. Since the mean accuracies of the classifiers presented in Figure 5.25, Figure 5.26, Figure 5.27 and Figure 5.28 are in the range of [0.7 and 08), the employment of the Naïve Bayes classifiers for these sets of object models are evaluated as satisfactory. In addition, as seen in Figure 5.27 and Figure 5.28, when a Naïve Bayes classifier is trained by using at least the 100 best fingerprints from the utilized object models, it receives an accuracy which is greater than 0.7 and less than 0.8. Hence, each Naïve Bayes classifier trained by using one of these object models is evaluated as satisfactory while others trained by using the either 25 or 50 best fingerprints are evaluated as closely satisfactory. Consequently, all these four major evaluations regarding ROC curves, PR curves, equal error rates and mean accuracies show that the previously presented set of object models generally conducts a better training session for the introduced Naïve Bayes classifier than the one presented in 5.32 and Figure 5.28.

Figure 5.29 and Figure 5.30 illustrate the classification results of the Naïve Bayes classifiers trained with a set of models whose precision scores are joint with a yellow curve in Figure 5.17 by using both ROC and PR curves. The subfigures presented in Figure 5.29 are generated by using the mentioned object models whose sizes are 25, 50 and 100. Moreover, the ROC and PR curves presented in Figure 5.30 are related to the object models from the same set keeping 150,200 and 250 fingerprints. As seen in these two figures, the range of areas under ROC curves is [61.93%, 63.20%] while the range of areas under PR curves is [42.57%, 44.34%]. Apart from that, the equal error rates of these foregoing classifiers place in the range of [39.48%, 41.68%].



(a) (b) (c) (d) (e) (f)

Figure 5.29: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using l_2 distance and the GB color histograms extracted from the mean shift segments obtained in CIELUV color space.



(a) (b)

(c) (d)

(e) (f)

Figure 5.30: The representation of both ROC and PR curves received from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using l_2 distance and the GB color histograms extracted from the mean shift segments obtained in CIELUV color space.

Since the accuracies received from the classifiers shown in Figure 5.29 and Figure 5.30 are less than 0.8 and greater than 0.7, the Naïve Bayes classifiers trained with this set of object models are evaluated as satisfactory in this thesis.

The left upper corners of the subfigures shown in (a), (c) and (e) of all the figures received from Naïve Bayes classifiers are closer to the majority of line segments of the ROC curves than the red dotted lines defining the respective experimental results of the random classifiers. In addition, the majority of the line segments of PR curves presented in these six figures is likewise higher than the red dotted lines received from the random classifiers. Table 5.3 represents the highest accuracies received from the Naïve Bayes classifiers as well as their mean accuracy.

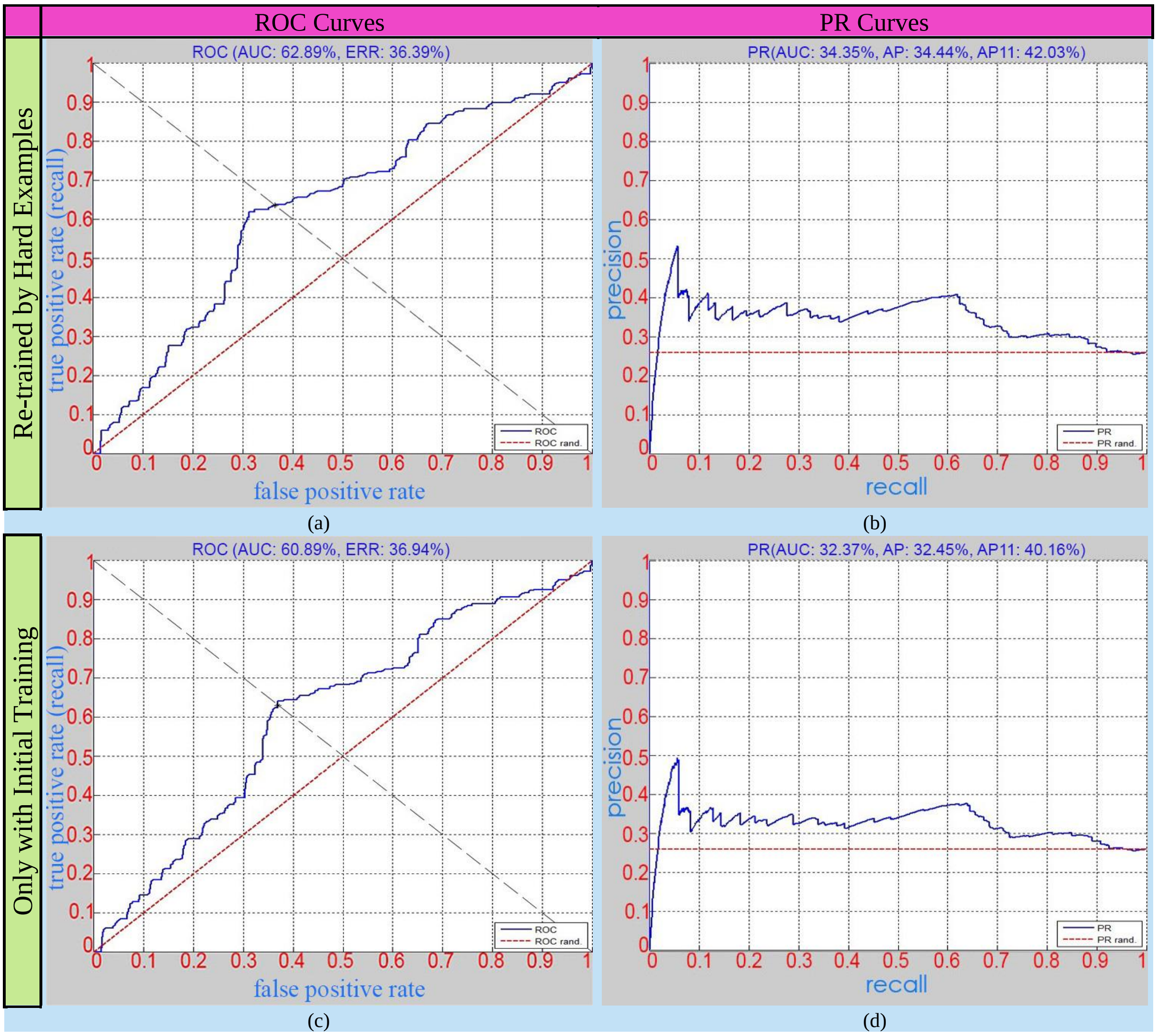
The Size of Object Model	150 (From Figure 5.26)	200 (From Figure 5.26)	250 (From Figure 5.26)	100 (From Figure 5.26)	250 (From Figure 5.28)	200 (From Figure 5.28)	**Mean Accuracy**
Accuracy	0.8230	0.8119	0.8047	0.7942	0.7730	0.7701	0.7961

Table 5.3: The highest mean accuracies obtained from Naïve Bayes-based Object Detection

5.2.2 SVM-BASED OBJECT DETECTION

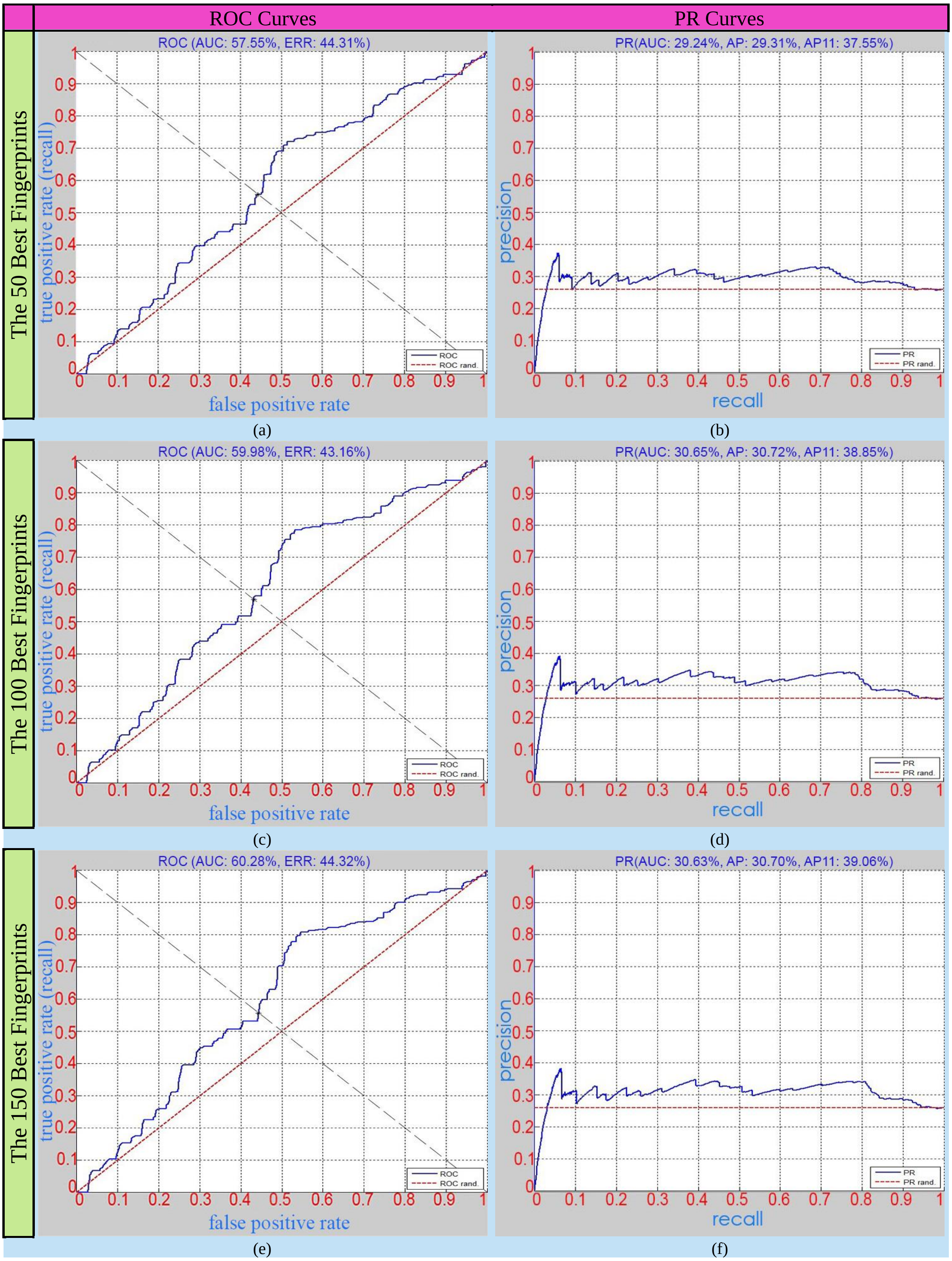
Figure 5.31 demonstrates both ROC and PR curves obtained from the linear SVM classifiers trained by using exactly the same object model used for the Naïve Bayes classifier whose results are shown in (a) and (b) of Figure 5.25. Therefore, this object model utilized for the presentation of Figure 5.31 also stores the 25 best fingerprints. Moreover, the results demonstrated in the first row of Figure 5.31 are obtained from the linear SVM classifier re-trained by using hard examples after its initial training with the foregoing object model and the randomly selected part of negative training dataset, but on the other hand, the last row of the same figure demonstrates the results related to the linear SVM classifier trained based on initial settings. As seen in these first and last rows of Figure 5.31, the area under the ROC curve enlarges 2% and the area under the PR curve enlarges 1.98% if the core (initial) training containing the foregoing object model keeping the 25 best fingerprints is completed for a linear SVM classifier and it is re-trained by using hard examples. Apart from that, this additional tuning of the classifier decreases the equal error rate as 0.55%. Nevertheless, the re-training with the hard examples improves the quality of the linear SVM classifier only if the foregoing object model from its complete set is used during its core training. Since the other object models store more fingerprints, this causes an overfitting during this re-training as an additional tuning process of the respective classifier.

Both ROC and PR curves received from the linear SVM classifiers trained by utilizing other object models storing more fingerprints from the foregoing set are presented in Figure 5.32 and Figure 5.33. One should note that the re-training is not performed on the linear SVM classifiers whose results related to the test images of development dataset are presented in these two figures. Each row of 5.37 and Figure 5.33 presents the results belonging to an SVM classifier trained with an object model storing a different number of fingerprints from the same set and the results presented in the rows are arranged in line with the numbers of the fingerprints belonging to the respective object models in ascending order. As shown in Figure 5.31, Figure 5.32 and Figure 5.33, the areas under the ROC and PR curves respectively change in the ranges of [51.32%, 60.89%] and [25.87%, 32.37%]. Apart from that, the equal error rates vary in the range of [36.94%, 48.54%]. According to all these results excluding the first row of Figure 5.31, the linear SVM classifier whose training session containing the object model storing only the 25 best fingerprints is evaluated as the best classifier from the ones trained with each object model belonging to the same foregoing set. On the other hand, the object model storing the 250 best fingerprints tunes each linear SVM classifier in the worst way from all the object models from this set. Some fragments of both PR curves shown in Figure 5.33 have even smaller precision values than the ones pertaining to the random classifier and their linear SVM classifiers trained by using the respective object models storing the either 200 or 250 best fingerprints have lower accuracies than 0.5; therefore, they are evaluated as unsatisfactory classifiers.



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

Figure 5.31: The representation of both ROC and PR curves received from the linear SVM classifier initially trained with the object model (keeping the 25 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments and the ones received from re-trained of the same classifier with the hard examples.



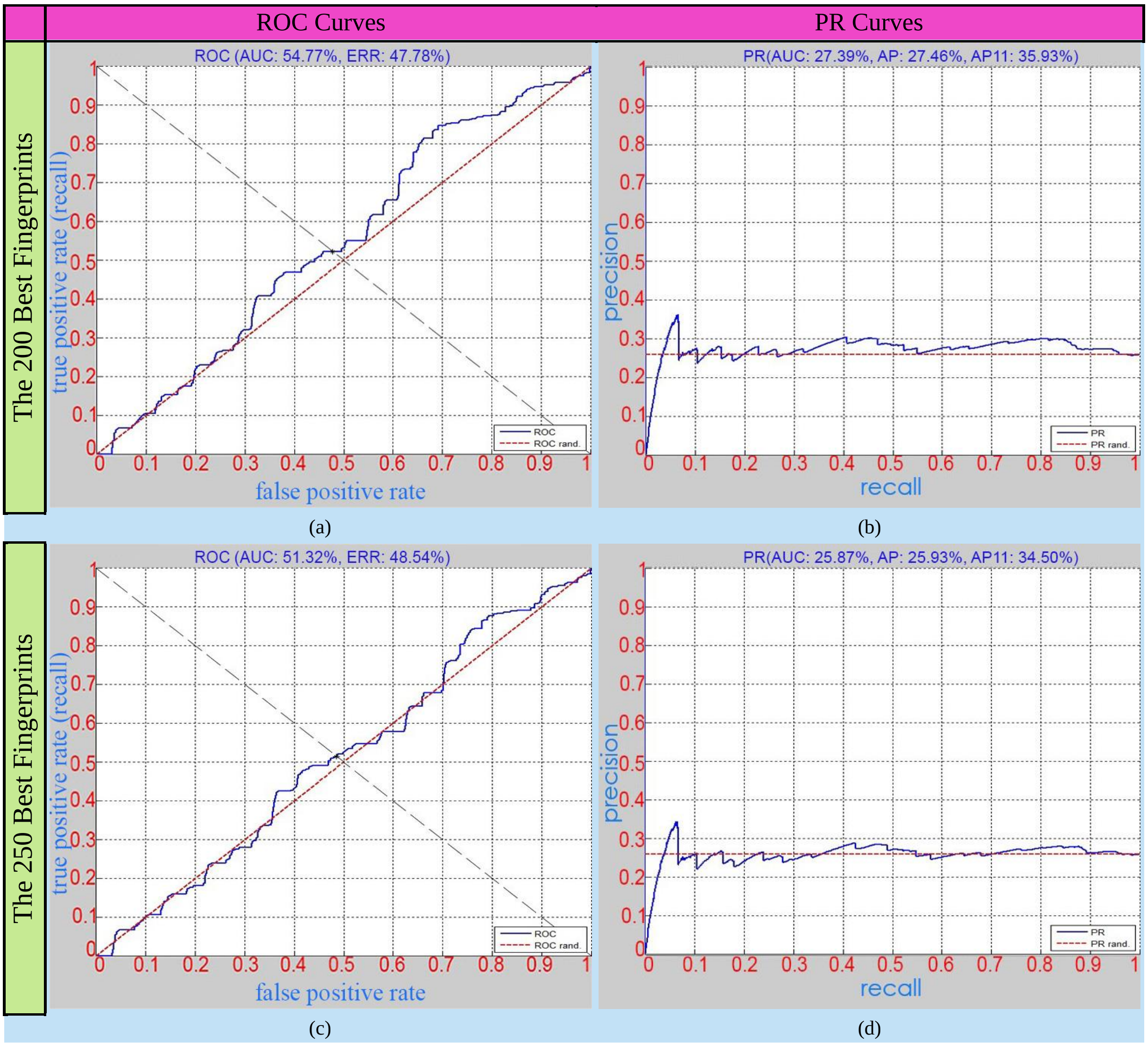
(a) (b)

(c) (d)

(e) (f)

Figure 5.32: The representation of both ROC and PR curves received from the linear SVM classifiers initially trained with the object models from the set (keeping the 50, 100 and 150 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

The highest accuracy of the linear SVM classifiers is even smaller than the lowest accuracy obtained from Naïve Bayes classifiers trained by VLAD descriptors. Nevertheless, the linear SVM classifier respective to this highest accuracy is closely satisfactory because its accuracy is greater than 0.5 and less than 0.7.



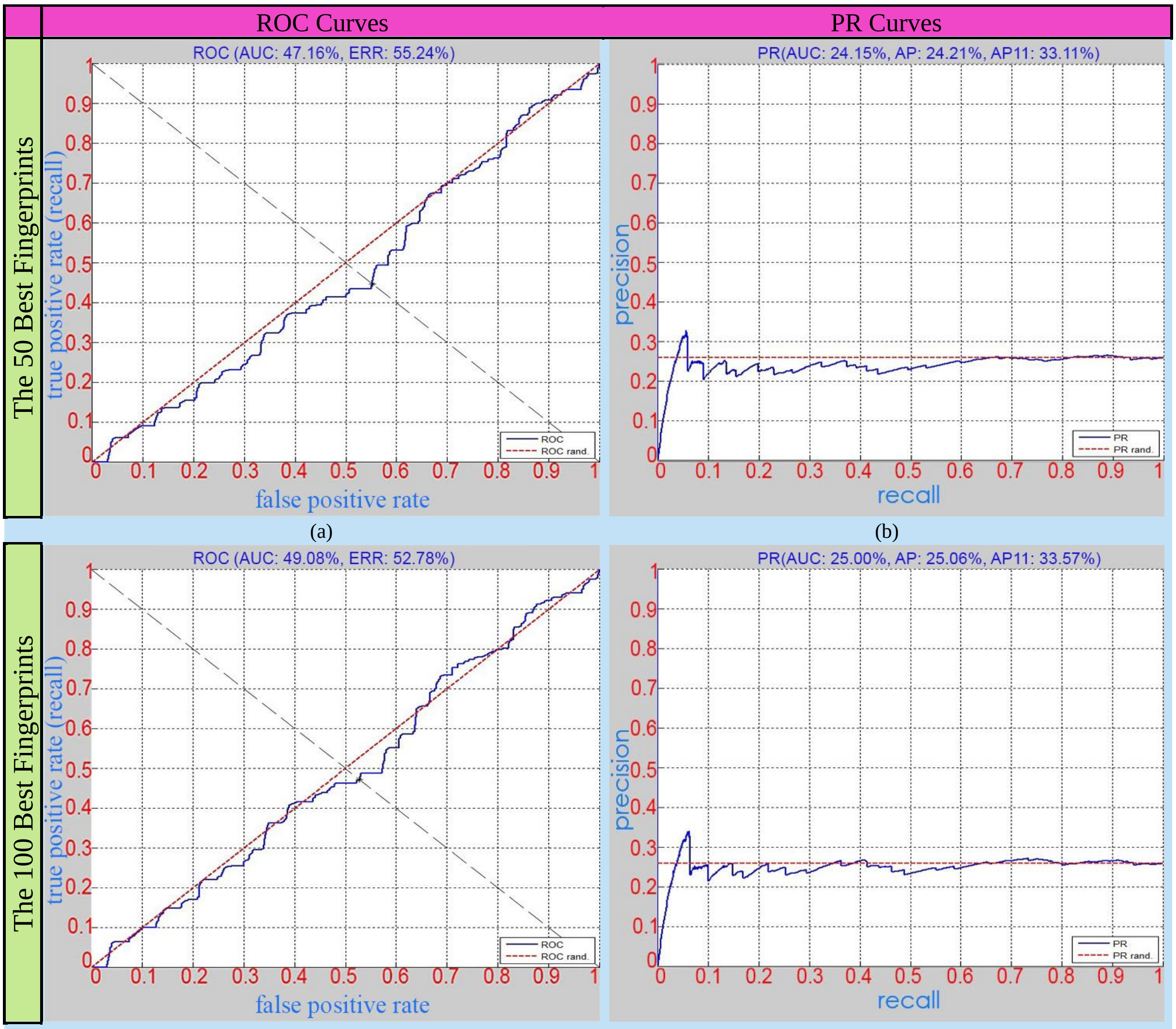
(a) (b)

(c) (d)

Figure 5.33: The representation of both ROC and PR curves received from the linear SVM classifiers initially trained with the object models from the set (keeping the 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

The mean accuracy decreases about 0.2783 when the linear SVM classifiers are employed for the same set of object models instead of the Naïve Bayes classifiers. Since the mean accuracy of the classifiers presented in Figure 5.32 and Figure 5.33 is slightly bigger than 0.5, the usage of this set of object models for the linear SVM classifiers is evaluated as closely satisfactory. All the respective results show that the previously introduced Naïve Bayes classifiers have broadly better qualifications than these linear SVM classifiers.

Figure 5.34 presents both ROC and PR curves in order to see the effect of already mentioned overfitting, which occurs due to the re-training of a specified linear SVM classifier with hard examples. These curves are obtained from the re-trained linear SVM classifiers whose core trainings are performed by using the either 50 or 100 best fingerprints likewise utilized for the first two rows of Figure 5.32. As seen in Figure 5.32 and Figure 5.33, the areas not only under ROC curves but also PR curves presented in the first two rows of Figure 5.32 are significantly greater than the ones shown in Figure 5.34.

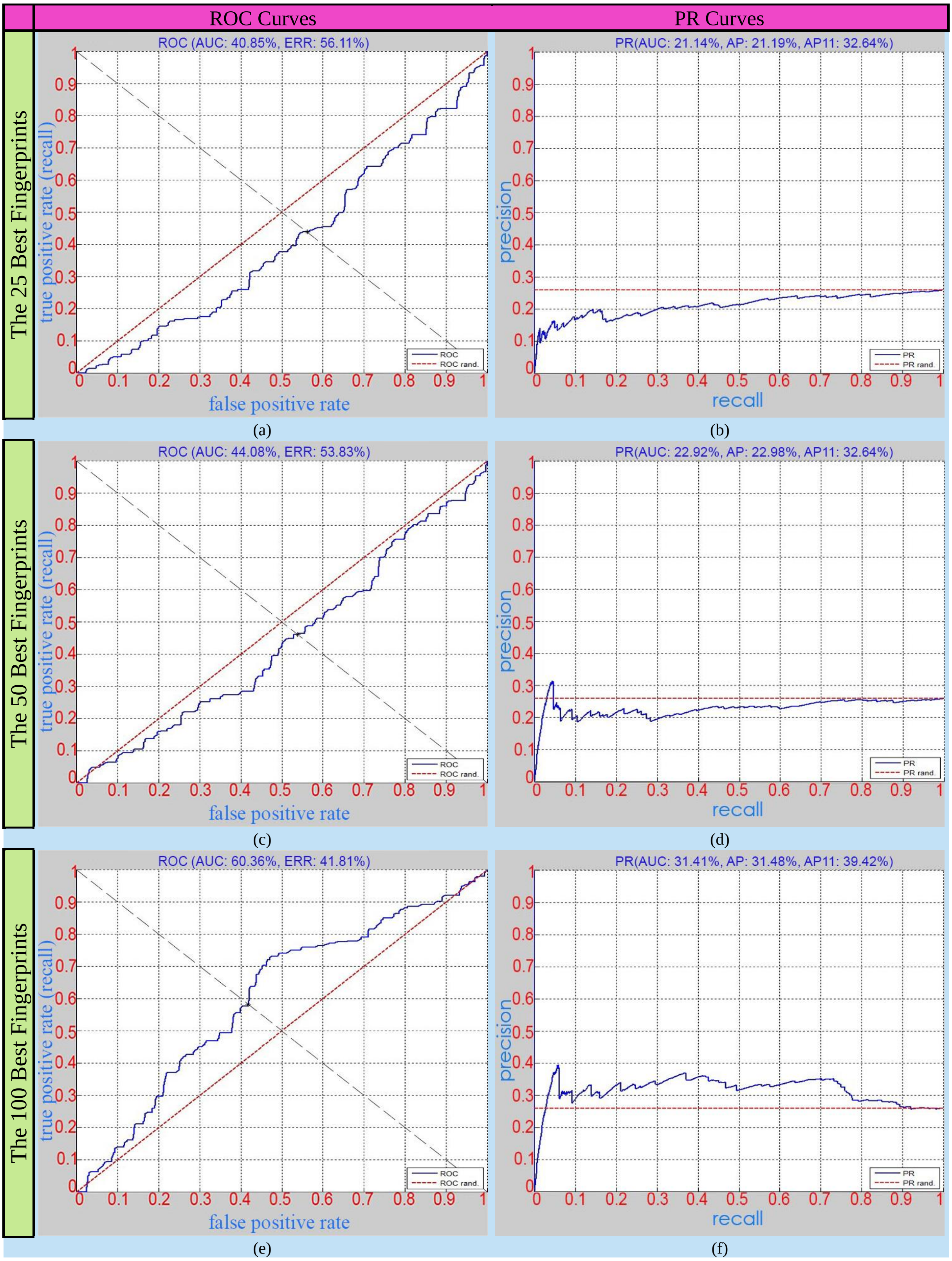


(a) (b)

(c) (d)

Figure 5.34: The representation of both ROC and PR curves received from the linear SVM classifiers re-trained with the hard examples after their initial training with the object models from the set (keeping the 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

The evaluations presented in Figure 5.35 and Figure 5.36 belong to the linear SVM classifiers which are tuned by performing only core training process containing the set of object models utilized in Figure 5.2 and built by l_2 distance. When the object model storing the either 25 or 50 best fingerprints is selected from the mentioned set for the training of the linear SVM classifier, many fragments of both ROC and PR curves obtained from this classifier place lower than the random classifier.



(a) (b)

(c) (d)

(e) (f)

Figure 5.35: The representation of both ROC and PR curves received from the linear SVM classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using l_2 distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

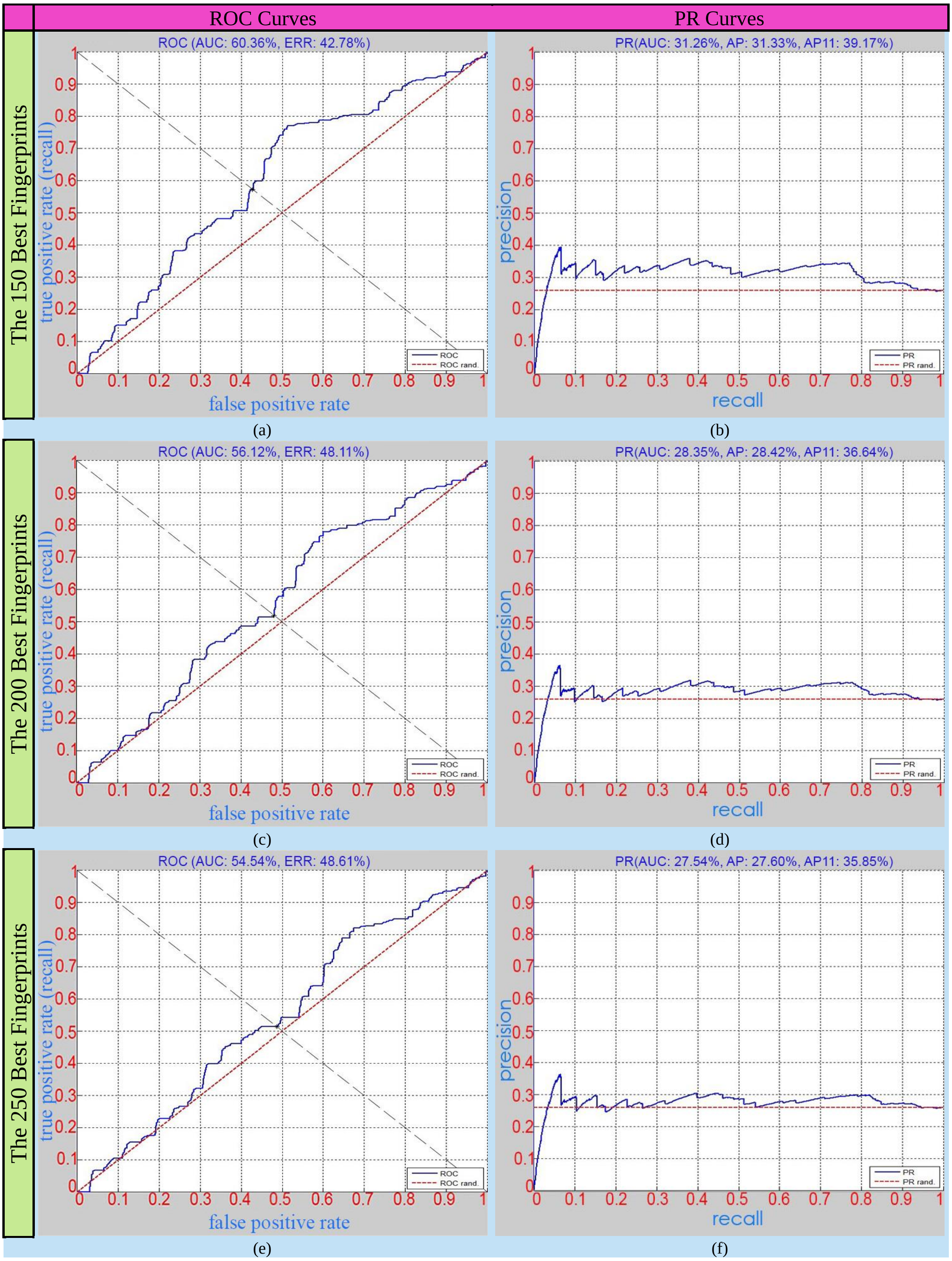


Figure 5.36: The representation of both ROC and PR curves received from the linear SVM classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using l_2 distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

The mean accuracy of the classifier presented in Figure 5.35 and Figure 5.36 is slightly smaller than the one received by the classifier shown in Figure 5.32 and Figure 5.33. Therefore, utilizing l_2 distance instead of diffusion distance for the similarity measurements of the VLAD descriptors extracted by sampling each two pixels does not supply a significantly better feature space in order to train a linear SVM classifier. The ranges of the areas under ROC and PR curves are respectively [40.85%, 60.36%] and [21.14%, 31.41%]. The largest areas under these ROC and PR curves are very similar to the ones having the largest areas from all PR and ROC curves of the previously introduced linear SVM classifiers tuned by having only core training session based on the set of object model recovered by using diffusion distance.

The areas under the PR curves presented in Figure 5.35 and Figure 5.36 are also significantly smaller than the ones obtained based on the firstly introduced two experimental setups containing the employment of Naïve Bayes classifiers and presented in Figure 5.25, Figure 5.26, Figure 5.27 and Figure 5.28.

The SVM classifiers trained (without any additional re-training session) by using the respective object models storing 25 or 100 or 150 fingerprints are evaluated as closely satisfactory since their accuracies are bigger than 0.5 and less than 0.7.

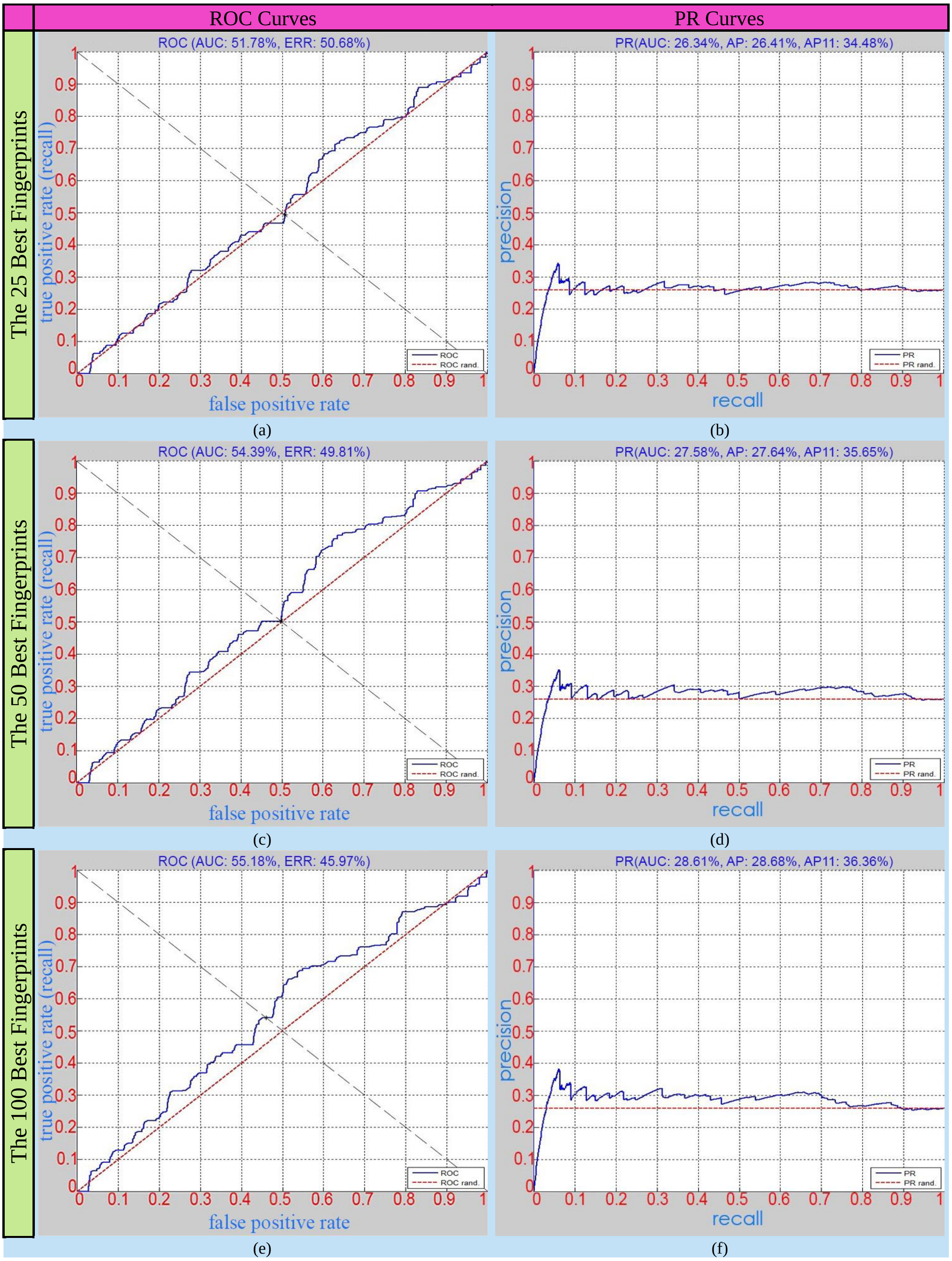
Table 5.4 represents the highest accuracies received from the linear SVM classifiers as well as their mean accuracy.

The Size of Object Model	25 (From Figure 5.31)	100 (From Figure 5.35)	150 (From Figure 5.36)	100 (From Figure 5.32)	50 (From Figure 5.32)	150 (From Figure 5.32)	**Mean Accuracy**
Accuracy	0.6248	0.5795	0.5594	0.5489	0.5446	0.5446	0.5669

Table 5.4: The highest mean accuracies obtained from SVM-based Object Detection

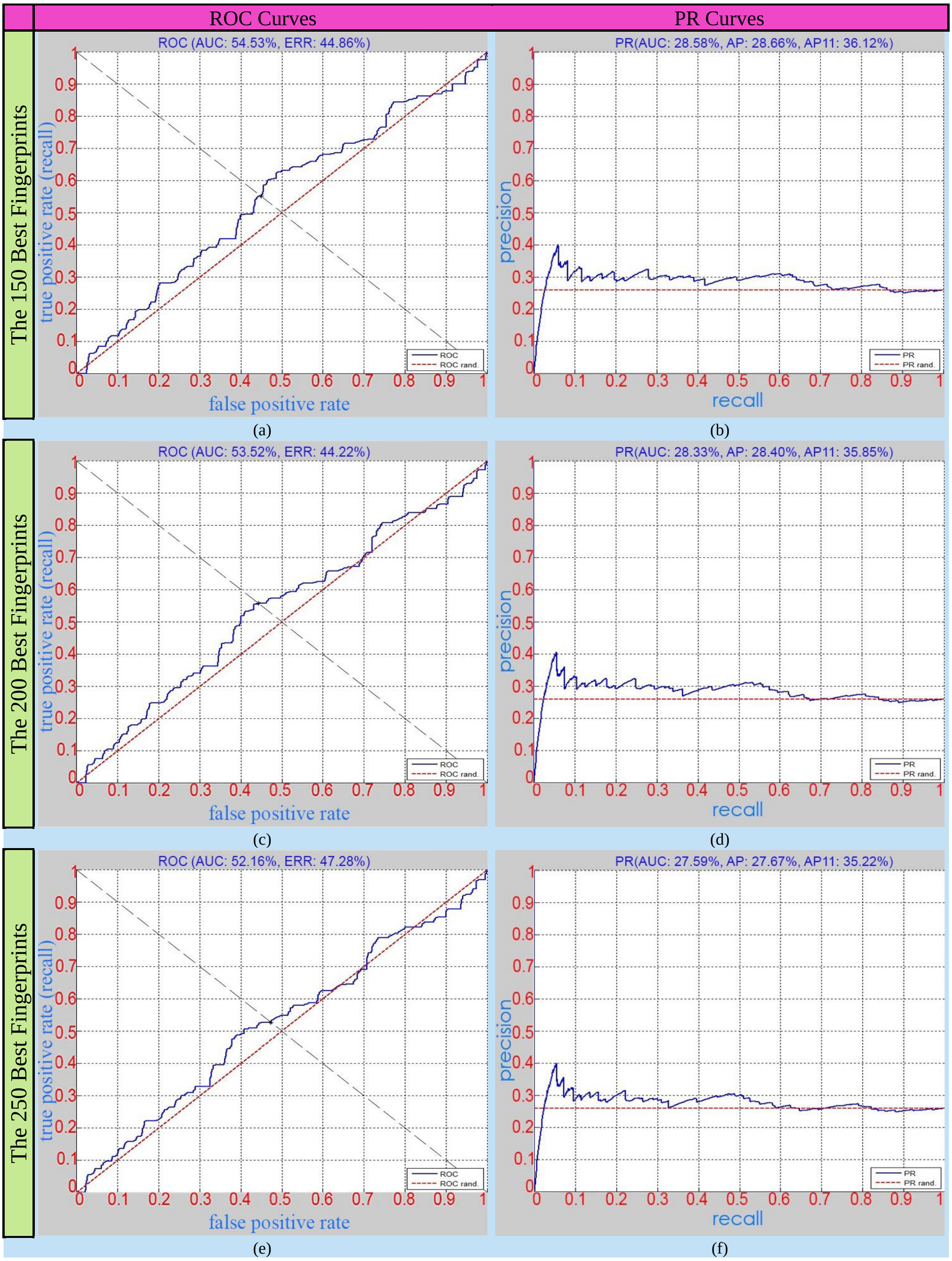
5.2.3 THRESHOLDS-BASED OBJECT DETECTION

The same set of object models utilized for the generation of Figure 5.31, Figure 5.32 and Figure 5.33 is also employed for the construction of a set of target visual object detectors by using the algorithm described as the third detector approach in Subsection 3.4. Their evaluations based on the test images of development dataset are illustrated in Figure 5.37 and Figure 5.38. Each object model selected from this set is employed to provide a distance space based on the thresholds values of its fingerprints. Since this set of object models is recovered by using diffusion distance, the same distance function is used for the similarity measurements between the test and training segments. The details of the criteria for the classification of the segments as foreground or background is provided in Subsection 3.4. As seen in Figure 5.37 and Figure 5.38, the majority of the fragments of both ROC and PR curves locates either above or on the red lines defined by random classifiers. Moreover, if the target visual object detector represented in Figure 5.35 and Figure 5.36 is constructed based on an object model storing at least 100 fingerprints, its accuracy becomes in the range [0.5129, 0.5622]. Therefore, these target visual object detectors are evaluated as closely satisfactory while the rest constructed based on the object models storing either 25 or 50 is evaluated as unsatisfactory.



(a) (b) (c) (d) (e) (f)

Figure 5.37: The representation of both ROC and PR curves received from the classifiers constructed based on the thresholds of the fingerprints belonging to the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.



(a) (b)

(c) (d)

(e) (f)

Figure 5.38: The representation of both ROC and PR curves received from the classifiers constructed based on the thresholds of the fingerprints belonging to the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

As presented in Figure 5.37 and Figure 5.38, the ROC curve placing in (e) of Figure 5.37 owns the largest area as 55.18% under itself from all the ROC curves presented in these two figures. The same classifier whose ROC curve having the largest area likewise receives the PR curve with the largest area as 28.61% shown in (f) of Figure 5.37. Apart from that, 44.22% is the smallest equal error rate obtained from all of these ROC curves and belongs to one presented in (c) of Figure 5.38. The largest areas under the curves of ROC and PR pertaining to each previously presented set of evaluations in this subsection are respectively larger than 55.18% and 28.61%.

Since the range of accuracies of the classifiers shown in Figure 5.37 and Figure 5.38 is tight, it shows that the qualifications of classifiers constructed by using the threshold values of the selected object models do not significantly change depending on the sizes of the selected object models from the same set. The PR ad ROC curves collected by changing the sampling step from 2 to 5 pixels in the configuration settings of the previous classifiers of this subsection are presented in Appendix A.4. Table 5.5 represents the highest accuracies received from the classifiers which are constructed based on the thresholds. Moreover, their mean accuracy is also represented in Table 5.5.

The Size of Object Model	150 (From Figure 5.38)	200 (From Figure 5.38)	250 (From Figure 5.38)	100 (From Figure 5.37)	50 (From Figure 5.37)	25 (From Figure 5.37)	**Mean Accuracy**
Accuracy	0.5594	0.5342	0.5622	0.5129	0.4687	0.4529	0.5147

Table 5.5: The highest mean accuracies obtained from Thresholds-based Object Detection

5.3 RESULTS OBTAINED FROM TIGER DATASET

In this section, two experimental setups selected from the previous sections of this chapter are employed for the species of tiger as a different target visual object than the one used in these previous sections. For this purpose, the experimental runs are launched based on the completely different sets of training and test images which forms the tiger dataset described in detailed in Subsection 4.1.2. The one of these experimental setups is selected based on the highest mean accuracy represented in the previous subsection. Since Table 5.3 presents the highest mean accuracy from all the tables placing in Section 5.2, the Naïve Bayes classifiers trained with a set of models whose recovery is based on the same experimental setup with the one defined with a violet curve in Figure 5.2 are employed for the detection of tigers from the tiger dataset. Consequently, the mentioned experimental setup regarding the object model recovery is the second selected experimental setup for this section and the respective set of object models is recovered based on it from the training images of the tiger dataset. Therefore, this section provides a verification of generalizability of the proposed approaches against each selection of target visual objects and datasets to the readers by showing their evaluations.

Other sets of object models are likewise recovered from the test images of the tiger dataset by changing only distance function from diffusion distance to the one of other seven options and keeping the rest of the mentioned second experimental setup. The precision scores received from all these sets of object models are presented in Figure 5.39. Hence, in the feature extraction processing step of their recovery, a dense extraction of VLAD descriptors at each two pixels is performed on the segments which are obtained by applying mean shift segmentation in CIELUV color space to the test images of the tiger dataset. After the VLAD descriptors are extracted, their twelve principal components are gathered by applying PCA to them. As seen in Figure 5.39, the violet curve joins the precision scores of the object models recovered by using diffusion distance. When the sizes of the object models related to this violet curve become smaller, their precision scores become greater. Therefore, the tilts directions of the line segments of this violet curve are desirable. Moreover, the mean precision scores of the sets defined with the curves in Figure 5.39 are shown in Table 5.6. Since the majority of the precision scores placing in the violet curve is greater than all the rest of the precision scores and their mean accuracy is shown as the highest one in Table 5.6, the set recovered by using diffusion distance is evaluated as the best one from all the sets for this experimental setup same as the one illustrated in Figure 5.2. Apart from that, it is evaluated as satisfactory because its mean accuracy is greater than 0.75 and less than 0.85. All the mean accuracies received from the rest of the sets are smaller than 0.65; therefore, they are evaluated as unsatisfactory.

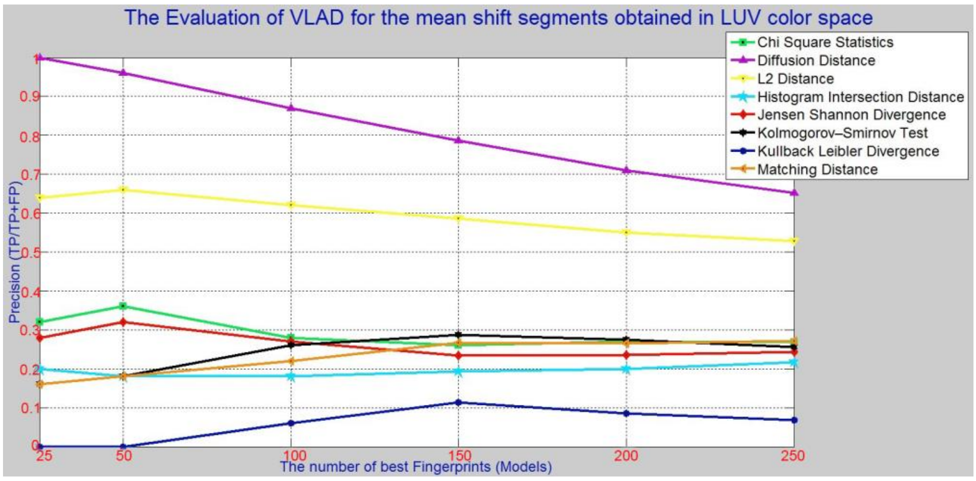
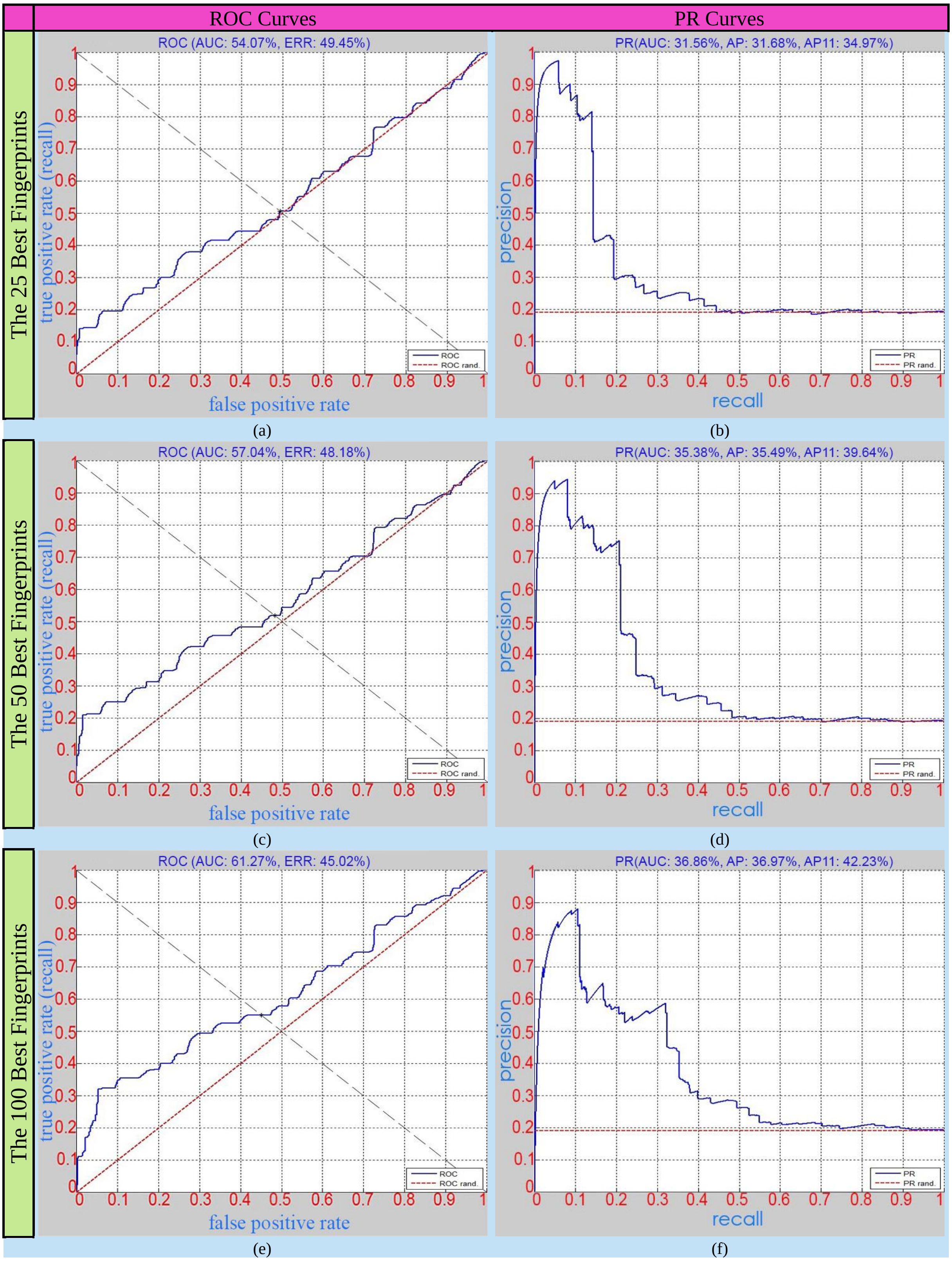


Figure 5.39: The representation of the precision scores collected based on the tiger dataset from the object models built by using VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

Distance function	Diffusion distance	L2 distance	Jensen Shannon divergence	Kolmogorov-Smirnov test	Kullback Leiber divergence	Chi square statistics	Histogram intersection distance	Matching distance
Mean Precision	0.8298	0.5974	0.2637	0.2363	0.0544	0.2930	0.1949	0.2273

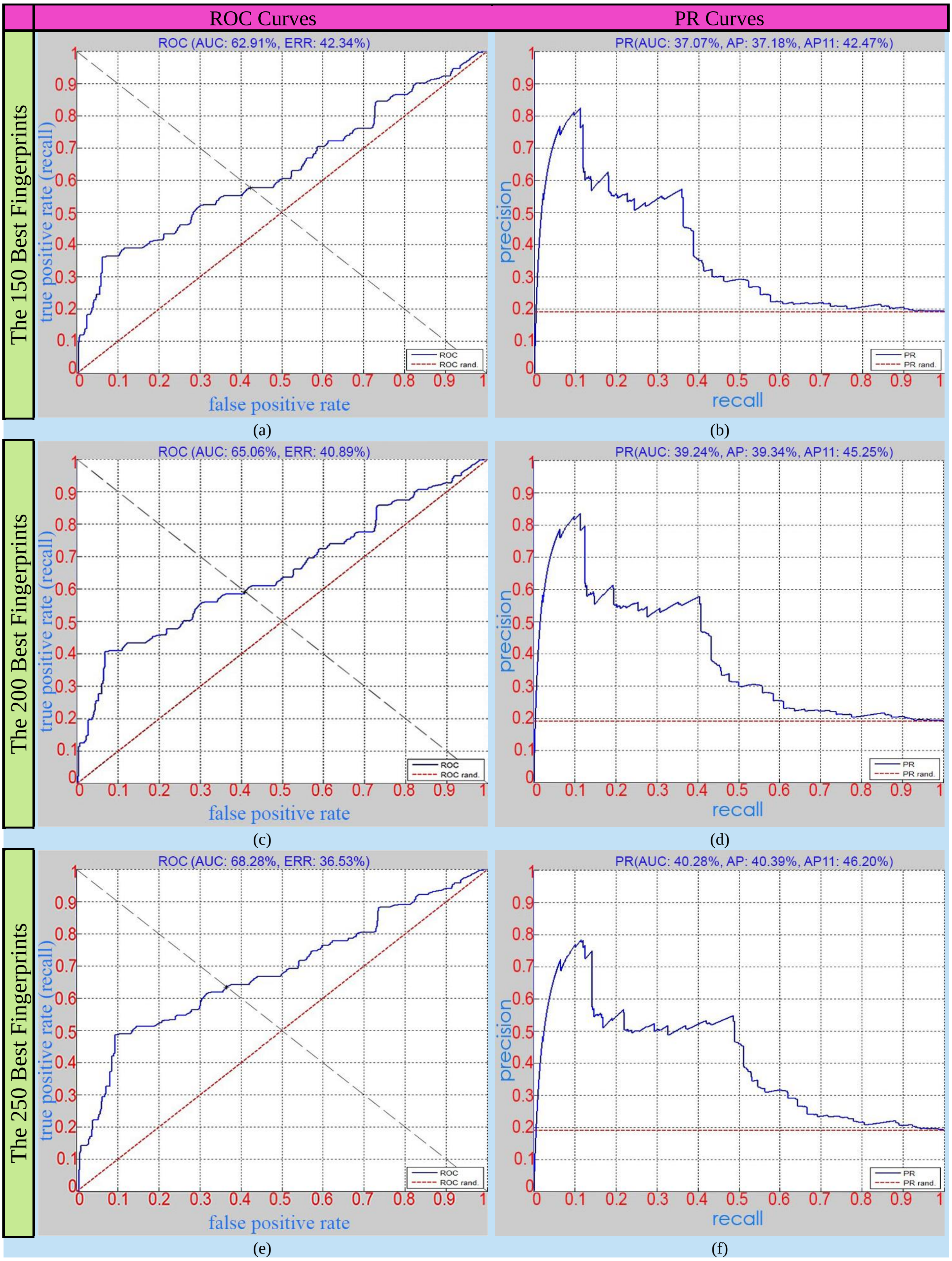
Table 5.6: The mean precisions of the set of models built based on the same experimental setup and the tiger dataset of Figure 5.39 by using the same distance function for each set are presented.

The both curves of ROC and PR received from each Naïve Bayes classifier trained with one of the object models recovered based on the experimental setup utilized for the illustration of Figure 5.39 by using diffusion distance are presented in Figure 5.40 and Figure 5.41. Same as the previous figures presenting ROC and PR curves, these two figures also present the curves related to each object model from the foregoing set in a separate row in ascending order according to the sizes of the object models. As seen in both figures, the majority of the line segments belonging to the ROC and PR curves places in more desirable locations than the lines defining the experimental results of the random classifiers.



(a) (b) (c) (d) (e) (f)

Figure 5.40: The representation of both ROC and PR curves collected based on the tiger dataset from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.



(a) (b) (c) (d) (e) (f)

Figure 5.41: The representation of both ROC and PR curves collected based on the tiger dataset from the Naïve Bayes classifiers trained with the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 2 pixels from the mean shift segments.

In Figure 5.40 and Figure 5.41, the ranges of the areas under ROC and PR curves and the equal error rates are respectively [54.07%, 68.28%], [31.56%, 40.28%] and [36.53%, 49.45%]. Apart from the areas under PR curves, other experimental results look similar to the ones presented in Figure 5.25 and Figure 5.26. The areas under PR curves presented in Figure 5.25 and Figure 5.26 are larger than the ones presented in Figure 5.40 and Figure 5.41.

The Size of Object Model	25	50	100	150	200	250	**Mean Accuracy**
Accuracy	0.8302	0.8362	0.8278	0.8269	0.8305	0.8257	0.8295

Table 5.7: The accuracies received from the Naïve Bayes classifiers whose ROC and PR curves are demonstrated in Figure 5.40 and Figure 5.41 are presented with their mean accuracy.

Furthermore, the accuracies of the mentioned Naïve Bayes classifiers based on the tiger dataset and their mean accuracy are presented in Table 5.7. Since all these accuracies are greater than 0.8, they are evaluated as highly satisfactory. Apart from that, they are slightly bigger than the ones presented in Table 5.3. Hence, using foregoing experimental setups for two different datasets and target visual object models receive similar evaluations and all of them can be considered as highly satisfactory. While the accuracies of the Naïve Bayes classifiers employed for the detection of the species of elephant are slightly smaller than the ones utilized for the detection of the species of tiger, the former Naïve Bayes classifiers receive slightly better ROC curves, PR curves and equal error rates than the latter ones.

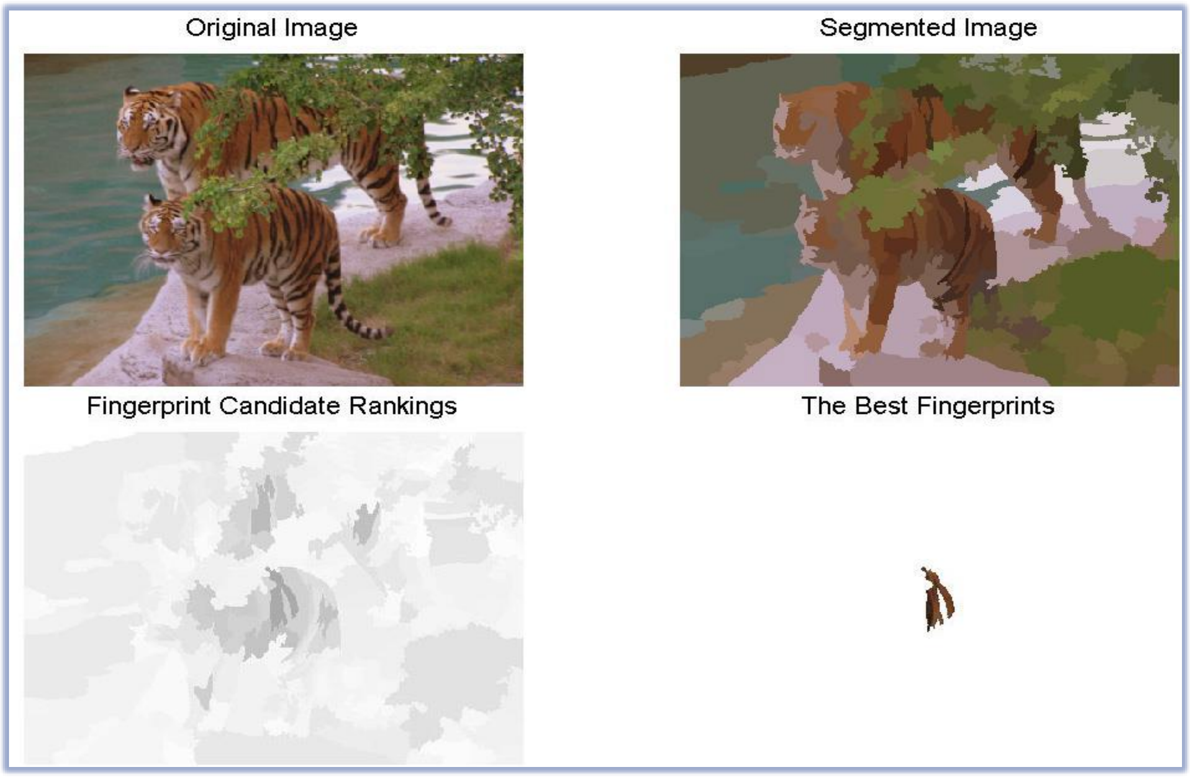


Figure 5.42: The recovery process of the object model (storing the 25 best fingerprints and having 100% precision score shown in Figure 5.39) based on the first sample training image of the tiger dataset is visualized.

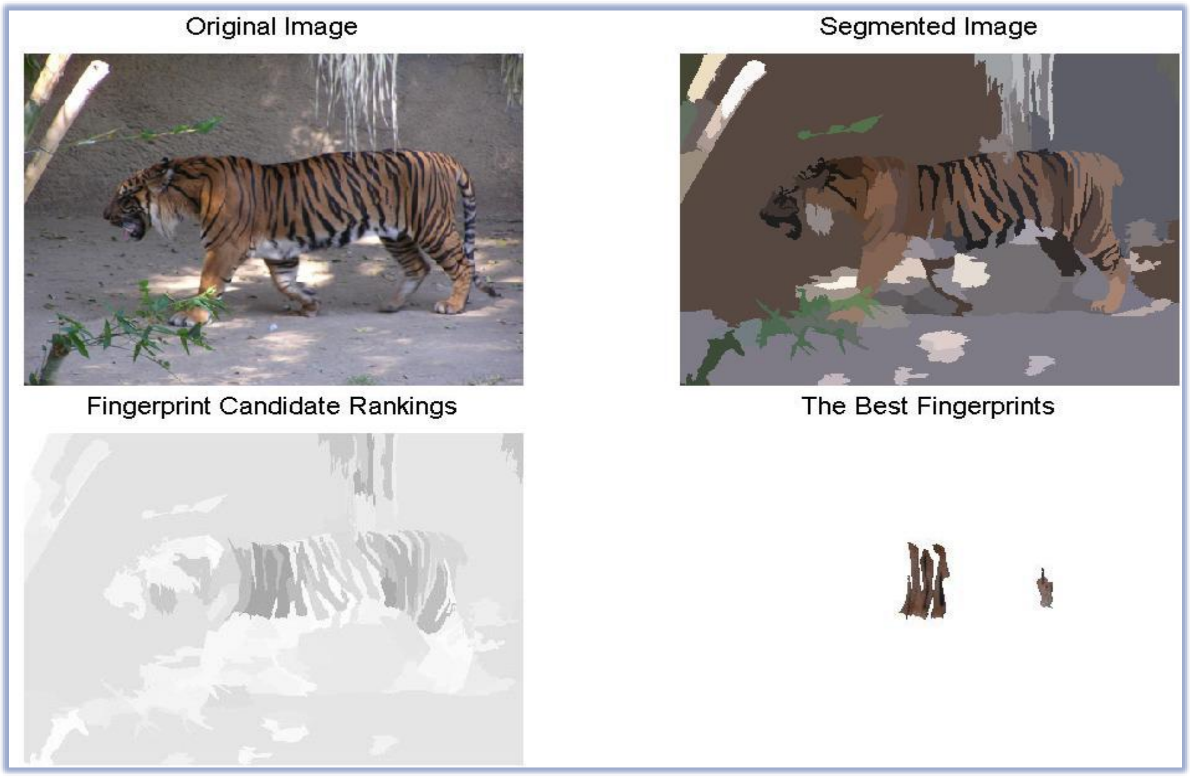


Figure 5.43: The recovery process of the object model (storing the 25 best fingerprints and having 100% precision score shown in Figure 5.39) based on the second sample training image of the tiger dataset is visualized.

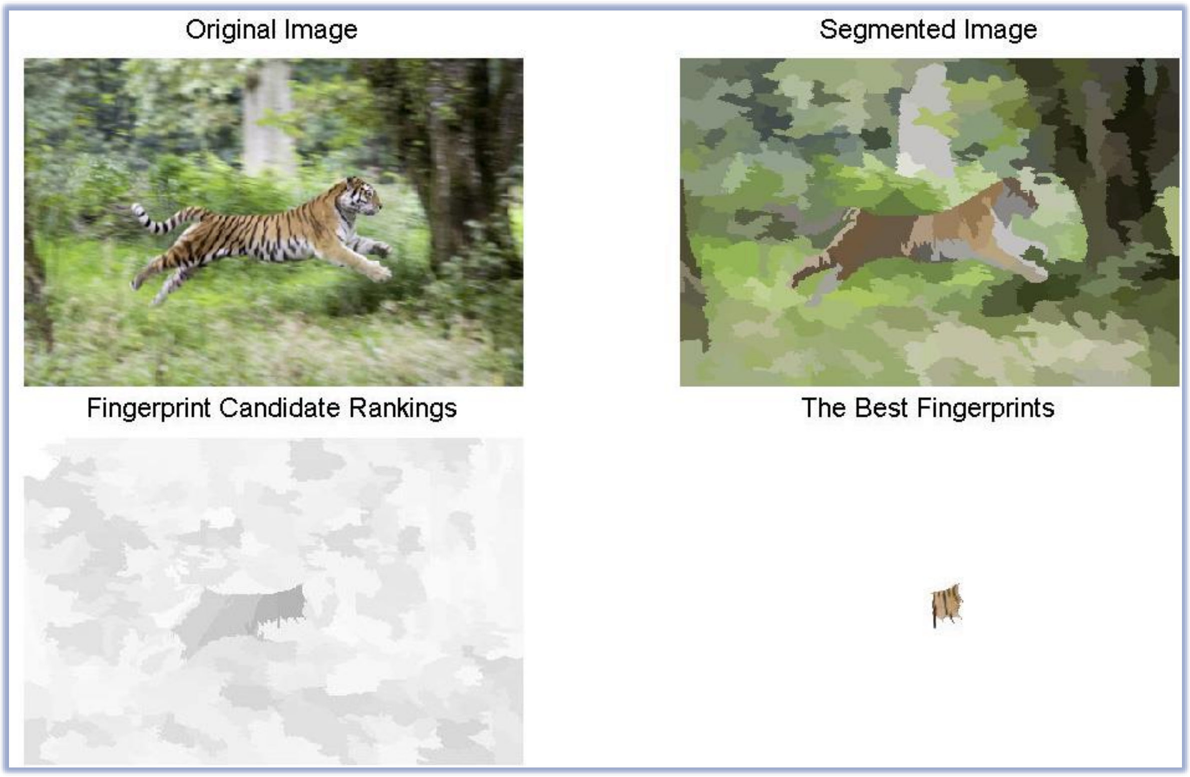


Figure 5.44: The recovery process of the object model (storing the 25 best fingerprints and having 100% precision score shown in Figure 5.39) based on the third sample training image of the tiger dataset is visualized.

Figure 5.42, Figure 5.43 and Figure 5.44 visualize the segmented images, the fingerprint candidate rankings and the selected fingerprints based on the selected object model from three samples of the training images of the tiger dataset. The selected object model keeps the 25 best fingerprints and places in the same set whose mean accuracy is presented in Table 5.7. As seen in each visualized recovery process of this object model from all these training images of the tiger dataset, their common quality for the discrimination of the foreground and background segments is similar to the ones presented in the subfigures placing in the second rows of Figure 5.6 and Figure 5.7 as well as the one presented in Figure 5.8. Since the stripes of the tigers are mostly colored darker than other segments in the fingerprint candidates shown in Figure 5.42, Figure 5.43 and Figure 5.44, they generally possess high information gain values and, therefore, the ones having the highest information gain values are selected as fingerprints for the mentioned object model.

5.4 CONCLUSION OF THE EXPERIMENTS

When diffusion distance is employed to compute the similarities between the descriptors which are obtained after performing l_2 and power normalization into 12 principal components of VLAD descriptors extracted from the mean shift segments, the captured similarity measurements enable a non-parametric discriminative weakly-supervised learning scheme based on information gain to recover a generalized object model for a target visual object. Moreover, the mean shift segments are collected from in CIELUV color space. The generalized object model collects the 25 best fingerprints from the selected entire training dataset and its experimental results obtained from two different training datasets utilized in this thesis show 100% precision scores. Since each of these training datasets is based on a different target visual object, it proofs the generalizability of the approach against the usage of different datasets and its independence against the selection of a target visual object. When the size of the generalized object model based on one of these two target visual objects is extended from 25 to the larger ones existing in the scope of the thesis, its precision score is decreased after each extension. This provides an evidence that if the information gain value assigned to a segment is higher than the one belonging to another segment, the former segment has more probability for being foreground segment than the latter. Nevertheless, if there is a strict bottleneck regarding execution timings in an implementation plan, the employment of diffusion distance can be replaced with l_2 distance by venturing 3.33% loss of the mean precision score (based on the evaluation result obtained from the development dataset of this thesis) since l_2 distance runs about 5 times faster than diffusion distance. However, one should note that diffusion distance provides better stabilization against the changes on the datasets and the target visual object since the loss in the mean precision score is much bigger for l_2 distance than diffusion distance when the development dataset is switched to the tiger dataset. Apart from that, another modeling approach containing the extraction of GB color histograms instead of VLAD descriptors from the same segments and the similarity measurements with the employment of l_2 distance receives a mean precision score which is only 3.83% smaller than the one pertaining to the modeling approach mentioned with VLAD descriptors in the beginning of this chapter. According to the experimental results obtained from the development dataset, the mean accuracy of the Naïve

Bayes classifiers is only 5.99% smaller when they are trained by using GB color histograms rather than VLAD descriptors. These two approaches possess higher and more standardized precision and accuracy scores than all other investigated and related state-of-the-art methods while the hard disk occupation is minimized.

CHAPTER 6

6 SUMMARY

Weakly-supervised framework of visual object detection is investigated in this thesis. For this purpose, an object model keeping the most discriminative segments of the target visual object is built from a set of image-level labeled training images. The recovered object model trains a classifier, which is used to detect the target visual object from a set of test images.

The research of object model recovery in this thesis starts with extending the weakly-supervised object modeling approach of Hao et al. [23] originated from texture classification to color image segments since the gathered accuracy results of the classification of insect sounds by using this proposed object model recovery from a training set of greyscale spectrogram segments are presented as 77% and 93% for 10 and 20 different insect categories, respectively. To this end, this thesis employs a tool of MPEG-1 video encoding by using CK-1 distance function presented by Campana et al. [28] and likewise used by [23] for their non-parametric similarity measurement framework. In fact, the usage of CK-1 distance function entails a segmentation method partitioning the segments with rectangular boundaries. Furthermore, each pair of segments of which the distance is computed by CK-1 distance function has to have exactly the same shape and area. Therefore, two different segmentation methods such as sliding windows and mean shift segmentation are investigated as the degrees of freedom in the processing step of segmentation. Since CK-1 distance function is applicable only to the segment pairs with rectangular boundaries having exactly the same shapes and areas, mean shift segments are extracted with their closest outer rectangular boundaries and then each pair of segments is resized based on their sizes by maintaining their aspect ratios. The background pixels are set with 0s (black color) in order to extract the segments with their closest outer rectangular boundaries.

A discriminative learning scheme based on information gain introduced by Hao et al. [23] is operated in this thesis by using the similarity measurements of a set of training image segments, which are computed by the CK-1 distance function. The image segments receiving the highest information gain values are collected to build a model for the target visual object.

The research way direction of the object model recovery is changed by replacing CK-1 distance function with a feature extraction technique and a similarity metric. Therefore, a wide range of different techniques for feature extraction and similarity measurement is investigated by keeping the same learning strategy, which is based on information gain. For instance, different types of color histograms are constructed as visual feature descriptors of image segments by using some or all three-color components of different color models such as CIE L*a*b, CIE L*u*v* and RGB. Apart from color histograms, the success of a texture based feature extraction technique [90] is investigated by employing dense SIFT descriptors. In addition to exploiting texture as the only visual attribute, other feature extraction techniques constructing histograms with bag of visual words model based on both visual features of color and texture placing in the image segments are employed as additional degrees of freedom in the processing step of feature extraction. These are PHOW, VLAD, CEDD and MPEG-7 color descriptors.

Finally, in order to detect the target visual object from the test images, a classifier is trained or constructed with the object models which are selected based on the best evaluations obtained from all the experimental results with respect to the object model recovery. Consequently, three different classification approaches are added in the scope of this thesis. A linear SVM classifier is employed as one of these different classification approaches while a Naïve Bayes classifier is the another one. Apart from these two, an alternative classifier working based on the spherical threshold spaces constructed by using the thresholds of the fingerprints belonging to the selected object model is employed as the third classification approach. Furthermore, the experimental results are likewise collected for the case that the linear SVM classifier is re-trained with the hard examples.

The precision scores of object models recovered by using mean shift segmentation are mostly higher than the ones recovered by using sliding windows when the rest of the experimental setup is the same. This happens due to the well-known drawback of the sliding windows method which is the generation of the segments with the rectangular shapes as the only one possibility. Since the most of the target visual objects do not have rectangular shapes, the segments extracted by sliding windows mostly contains background pixels. Apart from that, since the fixed sized segmentation is used for the sliding windows method introduced by this thesis, it conflicts with the property of the scale invariance. This is another root cause of the lower precision scores of the object models built by using sliding windows. Setting black color for the background pixels causes a noise in each segment for the mean shift segmentation. However, the effect of this noise on the related experimental results is smaller than the drawback of sliding windows. The only way to eliminate these additional black pixels is to extract the closest inner rectangular boundaries of each segment; however, this type of extraction causes the loss of the shape information of each segment.

The reason of replacing the CK-1 distance function with other approaches is its execution time frame since the original idea is extended to color images. The execution time of a single optimal object model recovery from 30 positive and 30 negative training color images by using CK-1 distance function takes approximately from 2 and half days to 6 days depending on the selected segmentation method although each training image is downscaled to its quarter size before the training images are segmented. Another drawback of employment of CK-1 distance function is that it does not supply indices representing the corresponding image segments, contrary to the global visual feature descriptors of the image segments. Therefore, an entire set of training images has to be kept in the hard disk drives due to the possible future repetitions of learning process during the detection of target visual objects from the test images although the size of each image is larger than each type of its global feature descriptor.

Since VLAD descriptors keep the distances between the local descriptors and their assigned visual words, this provides a persistence regarding the discrimination of the descriptors against dimension reduction performed by PCA. This additional info kept by VLAD descriptors provides better experimental results for the recovery of the object models than the ones built by either dense SIFT or PHOW descriptors. The highest mean precision scores from all the experimental results regarding object model recovery belongs to the ones using VLAD descriptors in their processing step of feature extraction. The reason why the precision scores of the object models recovered by using VLAD descriptors than the ones recovered by color histograms could be the loss of spatial information of image segments in the color histograms. Although CEDD and MPEG-7 color descriptors also keep both color and texture information from the image segments, the precision scores received from the modeling approaches containing one of these visual feature descriptors are evaluated as unsatisfactory contrary to the ones utilizing VLAD descriptors.

The complete set of the experimental results collected in the part of the thesis regarding the detection of the target visual objects show that the Naïve Bayes classifiers trained with the same set of object models having highest mean precision scores from the recovery part receive not only the highest mean accuracies but also the largest areas under ROC and PR curves. One should note that the experimental setups producing the best evaluations from the development dataset of the thesis are also employed on a different dataset containing a different target visual object in order to verify their stability against the changes of these conditions.

6.1 FUTURE WORK

As the main focus of the thesis is weakly-supervised object modeling, the research regarding the part of target visual object detection can be extended further by finding a junction point accumulating the advantages of the already introduced classification approaches to receive higher accuracies and larger areas under ROC and PR curves. Moreover, the generalizability of the proposed approaches is verified in this thesis. Therefore, the experimental setup of object model recovery evaluated as the best one in this thesis will be employed to detect target visual objects or abnormal behaviors from images sequences.

APPENDIX A

A.1 FORMAL PROOF OF A PROBLEM REDUCTION

The reduction from the problem of universal string compression to the problem of Kolmogorov complexity can be defined in polynomial time which is demonstrated in the following formal proof of the reduction. In this reduction, the problem of universal string compression makes use of the problem of Kolmogorov complexity as a subroutine. Moreover, this reduction is a special case of Turing reduction because the computation of the problem of universal string compression finishes immediately after the completion of the execution of problem of Kolmogorov complexity, which is called for the first time as a subroutine in the problem of universal string compression. Such a special case of Turing reductions is called many-one reductions, and if they can be defined in polynomial time, then those many-one reductions are called Karp reductions. Since it is specified above that, there is a reduction from the problem of universal string compression to the problem of Kolmogorov complexity in polynomial time, this particular kind of many-one reduction is a Karp reduction. In order to show the formal definition of this Karp reduction, the problems have to be presented in the form of decision problems. Hence, the following decision problems can be defined.

Definition of the decision problem of Kolmogorov complexity
INSTANCE: $\mathrm{L} = \{\mathrm{w} \in \{0,1\}^* \mid |\mathrm{w}| \text{ is the length of the program which prints out string x}\}$ and a string $y \in L$, i.e. L contains all strings w such that each string w, which is a string of the binary digits and is constructed by symbols 0 and 1. Moreover, w is the string of the program which prints out the string x.
QUESTION: $|y|$ is the shortest length of all lengths $|w|$ such that $w \in L$?

Definition of the decision problem of universal string compression
INSTANCE: $\mathrm{H} = \{\mathrm{c} \in \{0,1\}^* \mid |\mathrm{c}| \text{ is the length of the compression of the string z}\}$ and a string $q \in L$, i.e. H contains all strings c such that each string c, which is a string of the binary digits and is constructed by symbols 0 and 1. Moreover, c is the compressed string of the string z.
QUESTION: $|q|$ is the shortest length of all lengths $|c|$ such that $c \in H$?

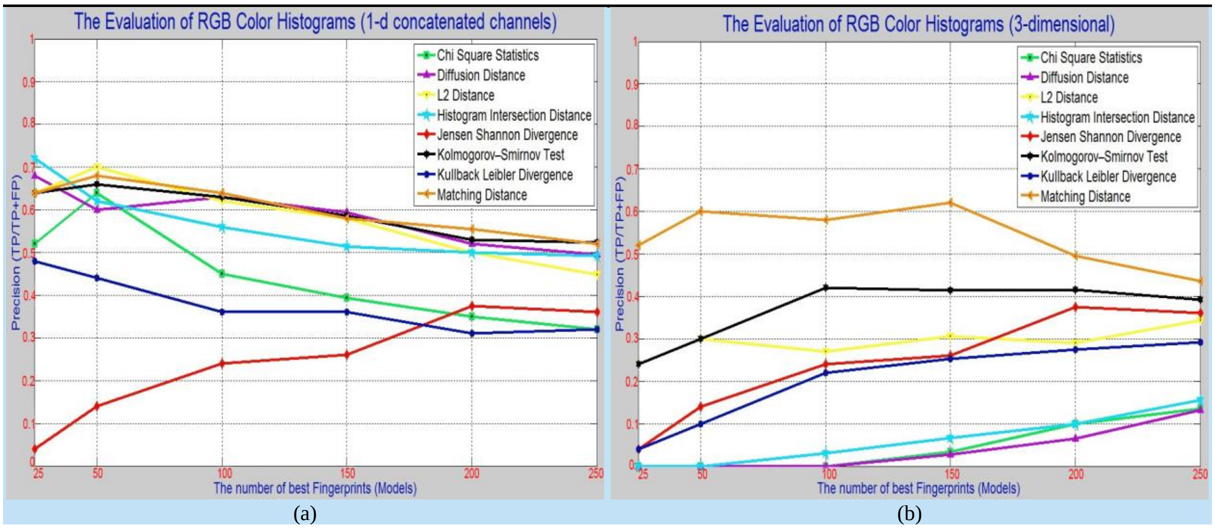
The reduction from the decision problem of universal string compression to the decision problem of Kolmogorov complexity is defined as follows. Let (H, q, z) be an arbitrary instance of the decision problem of universal string compression. An instance (L, y, x) is built by setting $H = L$, $x = z$, and $q = y$ since the randomness of a sample string is the core part of both problems. In fact, when the mapping reduction function R is applied to an instance $m = (H, q, z)$ as $R(m)$, then $R(m)$ maps m to an instance of (L, y, x). In order to prove the correctness of this reduction, the following has to be pointed out: (H, q, z) is a positive instance of **the decision problem of universal string compression** $\Leftrightarrow$ (L, y, x) is a positive instance of **the decision problem of Kolmogorov complexity**. The individual proof of each direction of this equivalence is presented below.

”⇒” Suppose that (H, q, z) is a positive instance of **the decision problem of universal string compression**, i.e. $|q|$ is the shortest length of all the compressed strings which can be obtained by compressing the string z in aid of the generalized version of the different string compression techniques utilized on a worldwide scale. Afterwards, $|y|$ becomes the shortest length of all the programs L printing the string z due to the settings of $H = L$, $x = z$, and $q = y$ by the problem reduction. It follows that (L, y, x) is a positive instance of **the decision problem of Kolmogorov complexity**.

”⇐” Suppose that (L, y, x) is a positive instance of **the decision problem of Kolmogorov complexity**, i.e. $|y|$ is the shortest length of all the programs which are implemented with a primitive programming language print out the string x. Afterwards, $|q|$ becomes the shortest length of all the compressed strings H for the given string z due to the settings of $H = L$, $x = z$, and $q = y$ by the problem reduction. It follows that (H, q, z) is a positive instance of **the decision problem of universal string compression**.

These settings are carried out in this problem reduction because the more repeated structure of the string x makes the length of the program printing x shorter while the more repeated structure of the string z makes the length of the compressed string shorter due to the execution of the lossless compressor on the string z. Therefore, the best string compressor can be approximated to the Kolmogorov complexity for a given string which is proven above. It is shown above that the setting of the reduction is applicable in polynomial time.

A.2 RESULTS OF RGB COLOR HISTOGRAMS-BASED MODEL RECOVERY



(a) (b)

Figure A.1: The representation of the precision scores received from the object models built by utilizing either 1d RGB color histograms constructed based on the concatenation of the color channels into a single dimensional vector or 3d RGB color histograms.

The precision scores presented in Figure A.1 belong to the object models built based on one of two different approaches which construct RGB color histograms from a given set of image segments. These two approaches are different than the one employed for the illustration of

Figure 5.20 and their contents are also explained in Subsection 3.2.1.4. The left subfigure of A.1 presents the precision scores obtained depending on an extraction process which transforms each channel of a given training image segment into a column vector and ultimately concatenate the columns vectors of all its three channels. On the other hand, the precision scores presented in the right subfigure of A.1 are collected when a three-dimensional RGB color histogram is extracted from each training image segment during their object recovery. The mean precision scores of all the sets presented in A.1 are less than 0.65; therefore, all these sets are unsatisfactory same as Figure 5.22 and Figure 5.24.

A.3 RESULTS OF DENSE SIFT AND PHOW DESCRIPTORS-BASED MODEL RECOVERY

The precision scores collected from the object models recovered based on all the introduced different experimental setups consisting of the employment of either PHOW descriptors or dense SIFT descriptors are presented in Figure A.2, Figure A.3, Figure A.4 and Figure A.5. The subfigures (a), (c) and (e) of Figure A.1 show the precision scores of object models concerning PHOW descriptors built by setting sampling step size to single pixel. In addition, switching only the sampling step size from single pixel to 2 pixels and aligning the rest of the experimental setup same as the left side of Figure A.2 generates the results shown in subfigures (b), (d) and (f) of Figure A.2. While the precision scores presented in Figure A.2 belonging to the object models are recovered based on three different vocabulary sizes such as 50, 100 and 150, the recovery of object models receiving the precision scores shown in Figure A.3 is accomplished by setting the vocabulary size to either 200 or 250. Therefore, the recovery processes utilized while obtaining the precision scores shown in Figure A.3 keeps the related subfigure alignments and their experimental setups same as Figure A.2 except vocabulary size.

The object models built by using histogram intersection shown in all the subfigures of both Figure A.2 and Figure A.3 generate continuously increasing precision scores while shifting from bigger object model to smaller object model apart from a single shift from the object model keeping 50 fingerprints to the one keeping 25 fingerprints shown in (c) of Figure A.2. These shapes of the curves with respect to histogram intersection are evaluated as a desired situation; however, all of their mean precision scores are unsatisfactory. The only sets of object models which obtain either satisfactory or closely satisfactory mean precision scores from all the precision scores shown in the subfigures of both Figure A.2 and Figure A.3 are the ones built by using Kullback Leibler divergence. Nevertheless, all the curves regarding Kullback Leibler divergence presented in these subfigures are undesirable because none of these curves has continuous increases in the precision scores while moving nodes from left to right direction. Furthermore, the sets presented in (b) of Figure A.2 and in (d) of Figure A.3 by the dark blue curves own unsatisfactory precision scores. The sets of object models which are built by utilizing one of the remaining distance functions collect unsatisfactory mean precision scores.

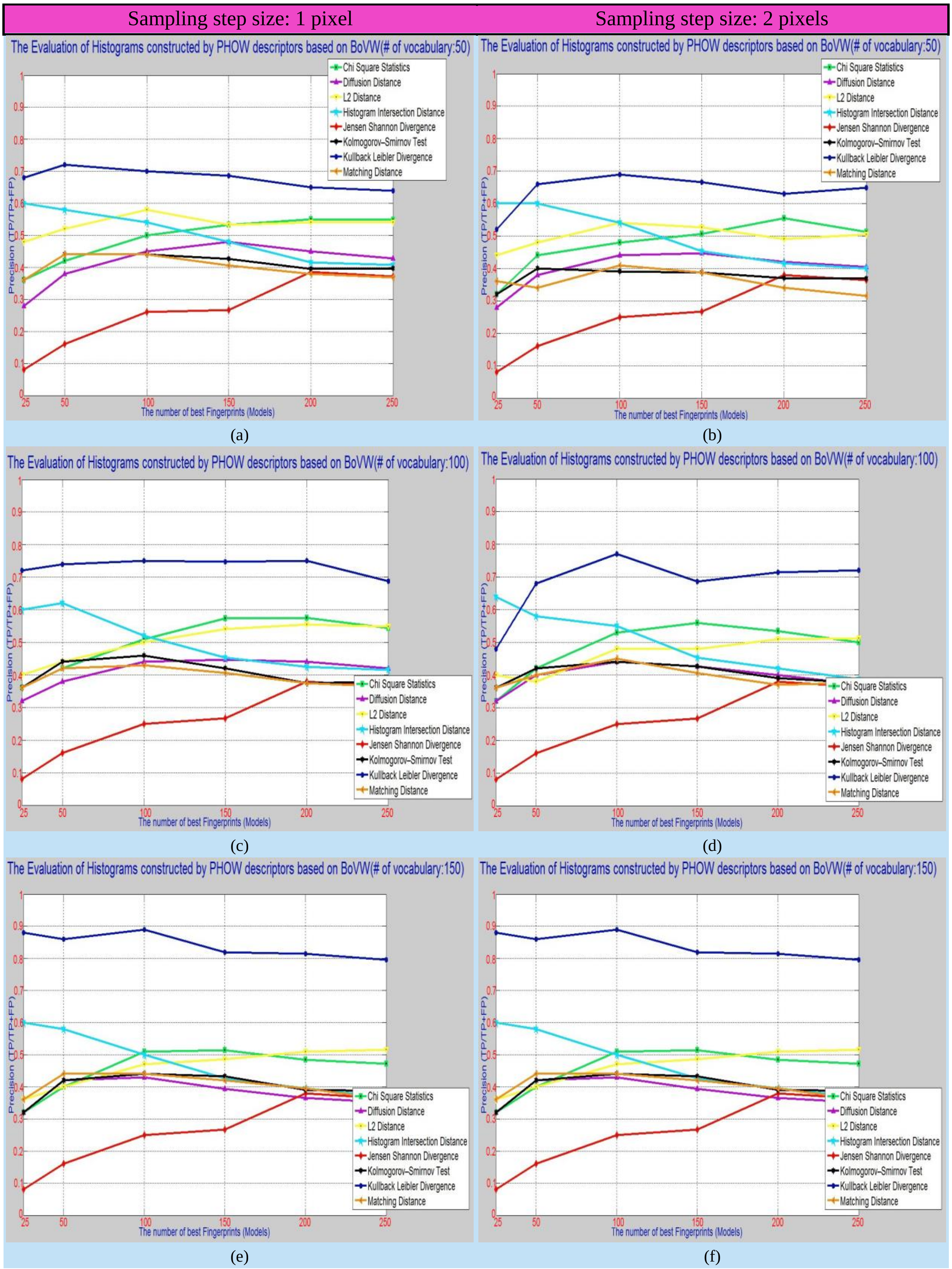
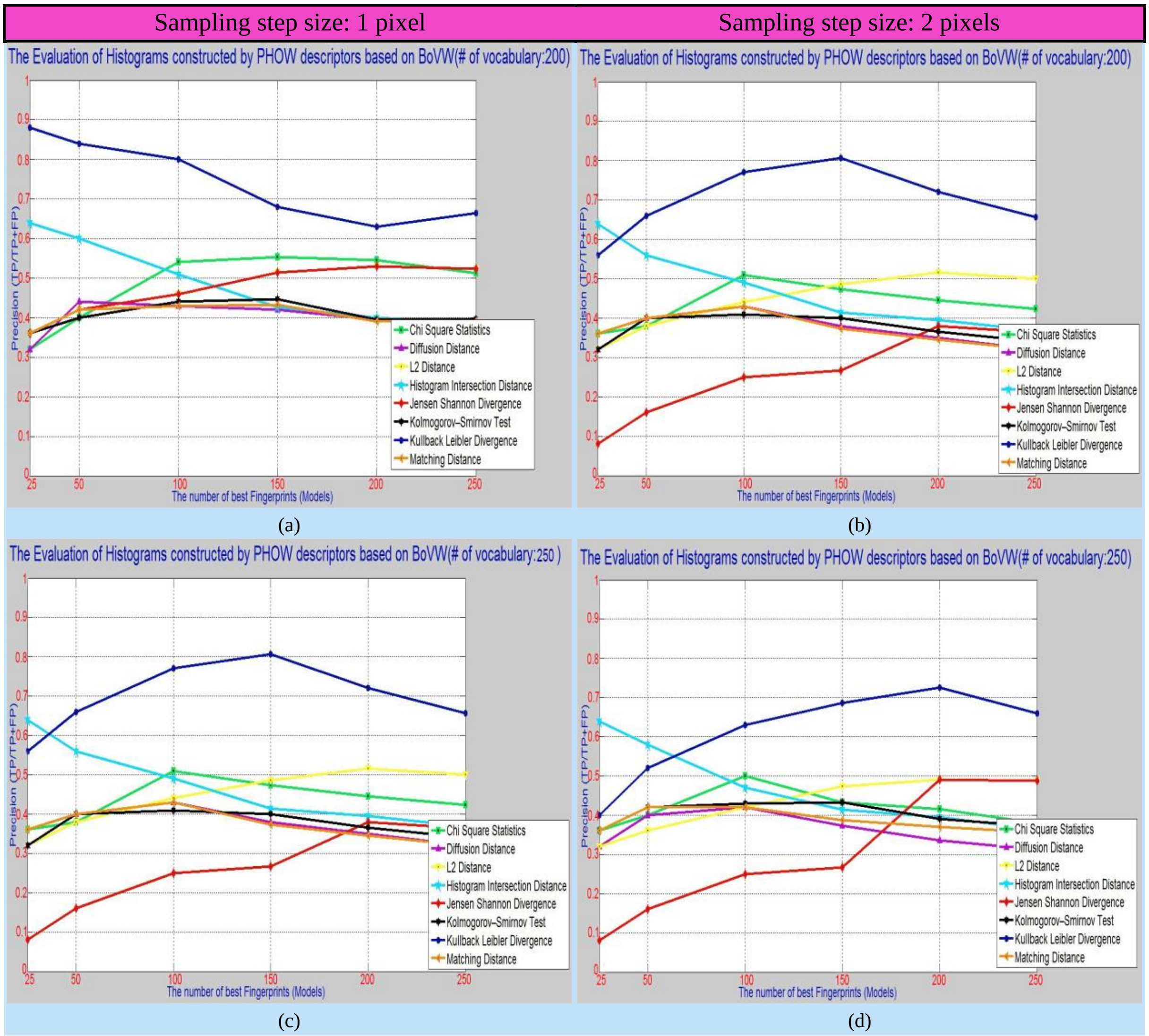


Figure A.2: The representation of the precision scores received from the object models recovered by constructing histograms based on the BoVW utilizing PHOW descriptors for different vocabulary sizes such as 50, 100 and 150.



(a)

(b)

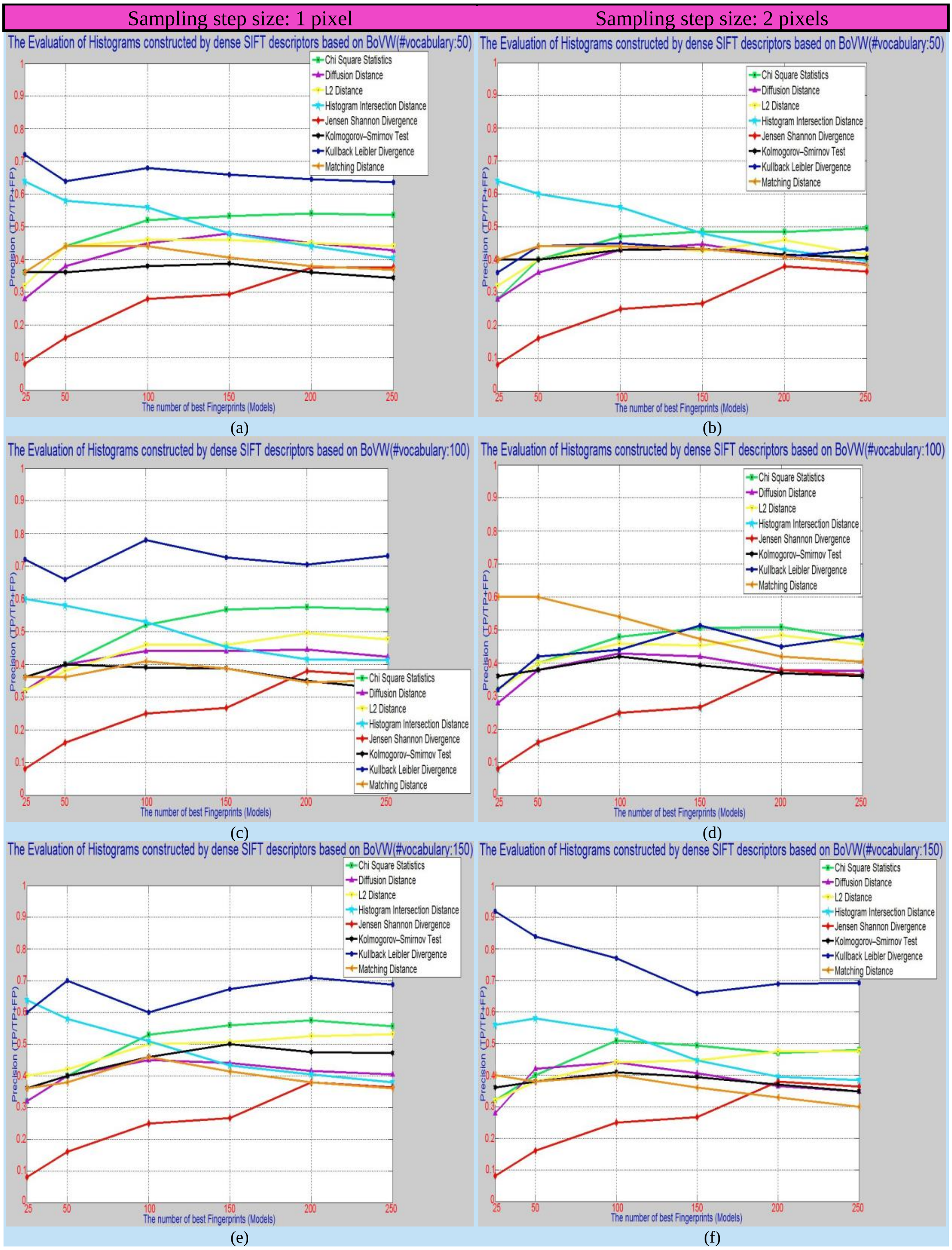
(c)

(d)

Figure A.3: The representation of the precision scores received from the object models recovered by constructing histograms based on the BoVW utilizing PHOW descriptors for different vocabulary sizes such as 200 and 250.

The subfigures (a), (c) and (e) of Figure A.4 illustrate the precision scores obtained by comparing the ground truth data and the object models recovered using the non-normalized histograms which are constructed based on the vocabulary sizes of 50,100 and 150 by extracting dense SIFT descriptors sampled at each pixel location. The results of subfigures (b), (d) and (f) of Figure A.4 are gathered with respect to the object models built with the same experimental setup of the left subfigures apart from the only change in the sampling step size by replacing the single pixel with 2 pixels.

Sampling step size: 1 pixel | Sampling step size: 2 pixels

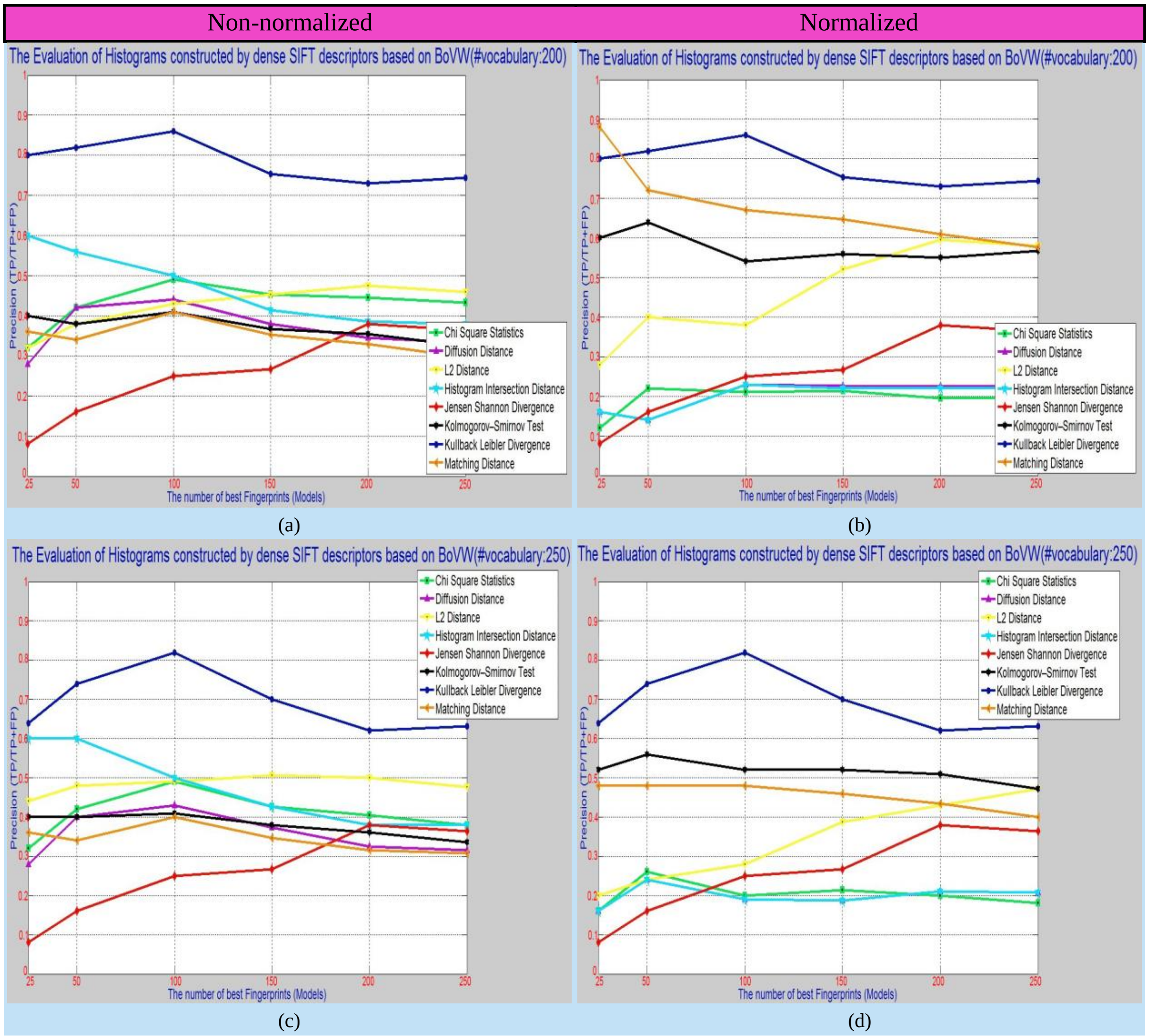


(a) (b)

(c) (d)

(e) (f)

Figure A.4: The representation of the precision scores received from the object models recovered by constructing non-normalized histograms based on the BoVW utilizing dense SIFT descriptors for different vocabulary sizes such as 50, 100 and 150.



(a) (b)

(c) (d)

Figure A.5: The representation of the precision scores received from the object models recovered by constructing either normalized or non-normalized histograms based on the BoVW utilizing dense SIFT descriptors for different vocabulary sizes such as 200 and 250.

The precision scores presented in the left subfigures of Figure A.5 are obtained from the object models built by launching experimental runs depending on exactly the same setup of the right subfigures of Figure A.4 including the alignment of the sampling step size as 2 pixels. The only difference between the object models shown in the left and right subfigures of Figure A.5 is that the extracted histograms are normalized in the right subfigures while they are not normalized in the left subfigures. As seen in Figure A.2, Figure A.3, Figure A.4 and Figure A.5, changing the sampling step size does not lead any significant improvement in most of the results regarding dense SIFT and PHOW descriptors. According to the left and right subfigures of Figure A.5, performing l_1 normalization on the dense SIFT descriptors increases the precision scores of all the object models built by using either matching distance or Kolmogorov-Smirnov test. On the other hand, the object models built by extracting dense SIFT descriptors and employing one of the three different distance functions such as diffusion distance, chi square statistics and histogram intersection distance gather bigger precision scores if l_1 normalization is not performed on the extracted descriptors during their recoveries. Two sets shown in both subfigures placing in the bottom of Figure A.2 by defining with dark blue curves have the highest mean precision scores from all the sets built by using either dense SIFT and PHOW descriptors. These two sifting lines traverse through the object models recovered by employing Kullback Leiber divergence and the mean precision scores of these two sets are equivalent. Since both of these scores are about 0.84, these two sets are evaluated as satisfactory. The set shown in (b) of Figure A.5 and built by using matching distance receive a closely satisfactory mean precision score. In addition, the sets with respect to Kullback Leiber divergence presented in Figure A.5 have either closely satisfactory or satisfactory mean precision scores. Likewise, the mean precision scores collected from the sets recovered by using the same distance function and demonstrated in some subsets of Figure A.4 are evaluated as either closely satisfactory or satisfactory. As seen in both Figure A.4 and Figure A.5, the rest of the sets obtain unsatisfactory mean precision scores.

A.4 SUPPLEMENTARY RESULTS OF THRESHOLDS-BASED OBJECT DETECTION

A shown in Figure A.6 and Figure A.7, a majority of both ROC and PR curve fragments places below the red lines generated by the random classifiers as an undesirable situation. Moreover, there is no classifier receiving any accuracy which is equal or greater than 0.5; therefore, all of these classifiers are evaluated as unsatisfactory. The accuracies of the classifiers represented in Figure A.6 and Figure A.7 are very similar to each other and even some of them are equal to each other; therefore, the difference between the maximum and the minimum accuracies is only 0,0137.

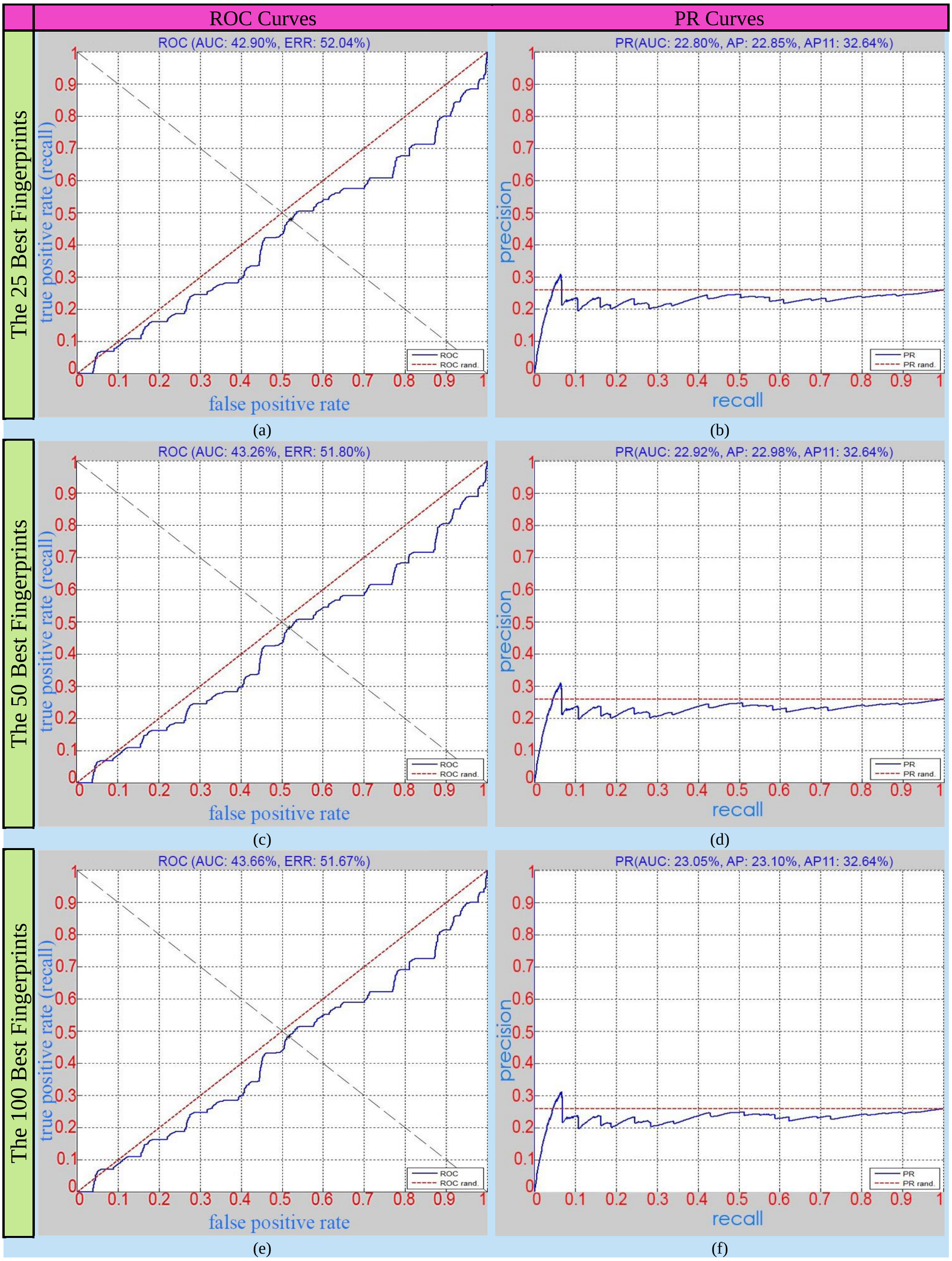
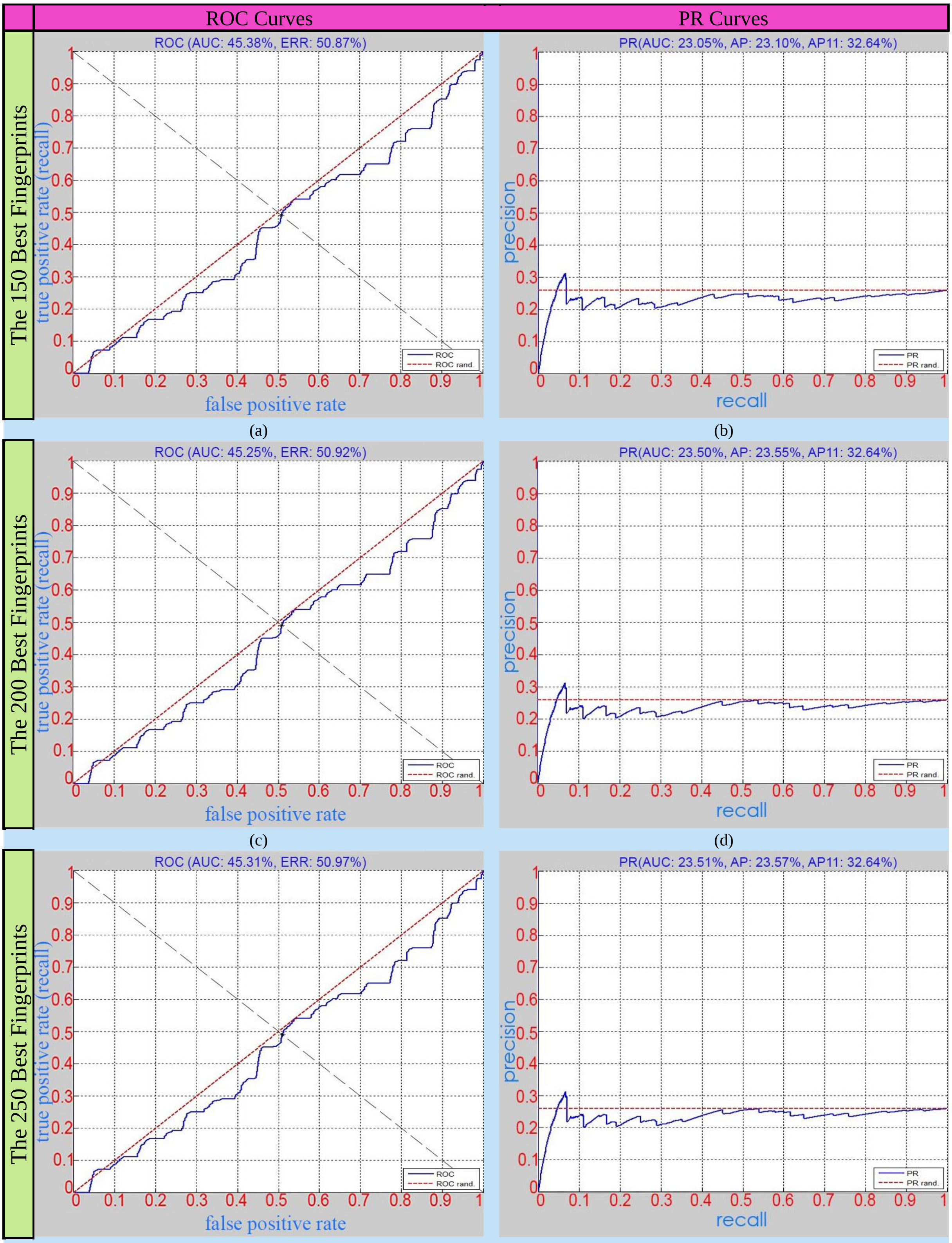


Figure A.6: The representation of both ROC and PR curves received from the classifiers constructed based on the thresholds of the fingerprints belonging to the object models from the set (only the ones keeping the 25, 50 and 100 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 5 pixels from the mean shift segments.



(a) (b) (c) (d) (e) (f)

Figure A.7: The representation of both ROC and PR curves received from the classifiers constructed based on the thresholds of the fingerprints belonging to the object models from the set (only the ones keeping the 150, 200 and 250 best fingerprints) recovered by using diffusion distance and the VLAD descriptors constructed based on the SIFT descriptors densely extracted at each 5 pixels from the mean shift segments.

REFERENCES

[1] Amnon Shashua, Yoram Gdalyahu, Gaby Hayun, "Pedestrian detection for driving assistance systems: single-frame classification and system level performance," in *IEEE Intelligent Vehicles Symposium*, 14-17 June 2004.

[2] Mingchen Gao, Junzhou Huang, Xiaolei Huang, Shaoting Zhang, Dimitris N. Metaxas, "Simplified Labeling Process for Medical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 7511 of the series Lecture Notes in Computer Science, Nice, France, Springer Berlin Heidelberg, 2012, pp. 387-394.

[3] Yu Zhang , Xiu-Shen Wei , Jianxin Wu , Jianfei Cai , Jiangbo Lu , Viet-Anh Nguyen , Minh N. Do, "Weakly Supervised Fine-Grained Categorization With Part-Based Image Representation," in *IEEE Transactions on Image Processing*, 18 February 2016.

[4] Shweta Singh, D Vijay Rao, "Recognition and Identification of Target Images using Feature Based Retrieval in UAV Missions," in *Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, Jodhpur, 18-21 Dec. 2013.

[5] Luis Patino, Hamid Benhadda, Nedra Nefzi, Bernard Boulay, Francois Bremond, Monique Thonnat, "Abnormal behavior detection in video protection systems," in *International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011)*, Sophia Antipolis, France, Sep 2011.

[6] Thirimachos Bourlai, Arun Ross, Anil Jain, "On Matching Digital Face Images Against Scanned Passport Photos," in *International Conference on Biometrics, Identity and Security (BIdS)*, Tampa, FL, USA, 22-23 Sept. 2009.

[7] Dingwen Zhang, Jianfeng Han, Dahai Yu, and Junwei Han, "Weakly Supervised Learning for Airplane Detection in Remote Sensing Images," in *The Proceedings of the Second International Conference on Communications, Signal Processing, and Systems*, 2014, Switzerland.

[8] Shaogang Gong, Tao Xiang, Visual Analysis of Behaviour: From Pixels to Semantics, Springer London, 2011, pp. 53-56.

[9] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, Second ed., New York, Chichester, Winheim, Brisbane, Singapore, Toronto: John Wiley & Sons, Inc., 2001, pp. 11, 16-17, 32-34, 605, 655.

[10] S. Abney, Semisupervised Learning for Computational Linguistics, 6000 Broken Sound Parkway NW, Suite 300: Chapman & Hall/CRC, 2008, pp. 4-8.

[11] "Flickr," Yahoo!, February 10, 2004. [Online]. Available: https://www.flickr.com/.

[12] D. D. Feng, Biomedical Information Technology, California, USA: Academic Press, Elsevier, 2008, pp. 83-107.

[13] Sara A. Solla, Todd K. Leen and Klaus-Robert Müller, "Advances in Neural Information Processing Systems 12," in *Proceedings of the 1999 Conference*, London.

[14] S. Niku, Introduction to Robotics, Second ed., John Wiley & Sons, Inc., October 2010, p. 412.

[15] Sean Ryan Fanello, Carlo Ciliberto, Lorenzo Natale, Giorgio Metta, "Weakly supervised strategies for natural object recognition in robotics," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 6-10 May 2013.

[16] R. Sukumar, The Living Elephants: Evolutionary Ecology, Behavior, and Conservation, 198 Madison Avenue, New York, New York 10016: Oxford University Press, 2003, pp. 383-396.

[17] M. Distefano, Homework Helpers: Biology, New Jersey, USA: The Career Press, 2004, p. 160.

[18] Alejandro Vaisman and Esteban Zimányi, "Data Warehouses: Next Challenges," in *Business Intelligence*, vol. 96, Springer Berlin Heidelberg, 2011, pp. 1-26.

[19] B. Thuraisingham, Managing and Mining Multimedia Databases, New York, USA: CRC Press LLC, 2001, pp. 72-75.

[20] Jiawei Han, Jian Pei, Micheline Kamber, Data Mining Concepts and Techniques, Third ed., Waltham, Massachusetts, USA: Elsevier Inc., 2012, p. 132.

[21] C. Schmid, "Constructing models for content-based image retrieval," in *IEEE International Conference on Computer Vision and Pattern Recognition*, Kauai, United States, Dec 2001.

[22] R. Szeliski, Computer Vision: Algorithms and Applications, London, Dordrecht, Heidelberg, New York: Springer, October 19, 2010, pp. 3, 78, 115, 222-223, 289-295, 662.

[23] Yuan Hao, Bilson Campana and Eamonn Keogh, "Monitoring and Mining Insect Sounds in Visual Space," in *the 12th SIAM International Conference on Data Mining (SDM 2012)*, Anaheim, California, USA, April 26-28, 2012.

[24] Li Deng, Douglas O’Shaughnessy, Speech Processing: A Dynamic and Optimization-Oriented Approach, New York, Basel: Marcel Dekker, Inc., 2003, pp. 23-28.

[25] David K. Mellinger, Christopher W. Clark, "Recognizing transient low-frequency whale sounds by spectrogram correlation," *Journal of Acoustical Society of America*, vol. 107, no. 6, pp. 3518-3529, 26 February 2000.

[26] Leonard A. Asimow, Mark M. Maxwell, Probability & Statistics with Applications: A Problem Solving Text, Winsted, CT: ACTEX Publications, Inc., 2010, pp. 310-318.

[27] P.R. Cohen, D. Jensen, "Overfitting explained," in *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics*, January 1997.

[28] Bilson J. L. Campana, Eamonn J. Keogh, "A Compression Based Distance Measure for Texture," *Journal Statistical Analysis and Data Mining*, vol. 3, no. 6, pp. 381-398, December 2010.

[29] O. Watanabe, Kolmogorov Complexity and Computational Complexity, Berlin, Heidelberg, New York, London, Paris, Tokyo, Honk Kong, Barcelona, Budapest: Springer-Verlag, 1992.

[30] J. L. Hein, Discrete Structures, Logic, and Computability, Third ed., Boston, Toronto, London, Singapore: Jones and Bartlett Publishers, 2010, pp. 828-836.

[31] O. S. Pianykh, Digital Imaging and Communications in Medicine (DICOM), Second ed., Heidelberg, Dordrecht, London, New York: Springer-Verlag, 2012, pp. 86-93.

[32] Lajos L. Hanzo, Peter Cherriman, Jurgen Streit, Video Compression and Communications: From Basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers, Second ed., Chichester, West Sussex, England: Wiley-IEEE Press, September 2007, pp. 12, 175, 379-385.

[33] Rafael Silva Pereira, Karin Breitman, Video Processing in the Cloud, First ed., London: Springer-Verlag, 2011, pp. 17-18.

[34] M. K. Mandal, Multimedia Signals and Systems, First ed., New York, NY, USA: Springer Science+Business Media,LCC, 2003, pp. 257-264.

[35] A. McAndrew, A Computational Introduction to Digital Image Processing, Second ed., New York, NY, USA: Chapman and Hall/CRC, November 5, 2015, pp. 125-133.

[36] John C. Russ, J. Christian Russ, Introduction to Image Processing and Analysis, Boca Ration, London, New York: CRC Press, October 31, 2007, p. 63.

[37] A. Bovik, The Essential Guide to Image Processing, Second ed., Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego San Francisco, Singapore, Sydney, Tokyo: Elsevier and Academic Press, 2009, pp. 110, 196-199.

[38] M. Bramer, Principles of Data Mining, London: Springer-Verlag, 2007, pp. 41-77, 135-154.

[39] Sakis Drosopoulos, Michael F. Claridge, Insect Sounds and Communication: Physiology, Behaviour, Ecology, and Evolution, 6000 Broken Sound Parkway NW, Suite 300: CRC Press, 2006, p. 8.

[40] Ramazan Gokberk Cinbis, Jakob Verbeek, Cordelia Schmid, "Multi-fold MIL Training for Weakly Supervised Object," in *CVPR 2014 - IEEE Conference on Computer Vision*, Columbus, United States, Jun 2014.

[41] Mark Everingham , Luc Van Gool, Christopher K. I. Williams, John Winn, Andrew Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision,* vol. 88, no. 2, pp. 303-338, June 2010.

[42] Yann Chevaleyre, Jean-Daniel Zucker, "A Framework for Learning Rules from Multiple Instance Data," in *Proceedings of the 12th European Conference on Machine Learning (ECML 2001)*, Freibug, Germany, Springer-Verlag, Sep 5-7 2001, pp. 49-60.

[43] Schlegl T., Waldstein S.M., Vogl WD., Schmidt-Erfurth U., Langs G., "Predicting Semantic Descriptions from Medical Images with Convolutional Neural Networks," in *Information Processing in Medical Imaging*, 2015.

[44] Jorge Sanchez, Florent Perronnin, Thomas Mensink, Jakob Verbeek, "Image Classification with the Fisher Vector: Theory and Practice," *International Journal of Computer Vision,* vol. 105, no. 3, pp. 222-245, December 2013.

[45] J. R. R. Uijlings , K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision,* vol. 104, no. 2, pp. 154-171, September 2013.

[46] C. Lawrence Zitnick , Sing Bing Kang, "Stereo for Image-Based Rendering using Image Over-Segmentation," *International Journal of Computer Vision,* vol. 75, no. 1, pp. 49-65, October 2007.

[47] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, November 2004.

[48] Koen E. A. van de Sande, Theo Gevers, Cees G. M. Snoek, "Evaluation of Color Descriptors for Object and Scene Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008),* Anchorage, AK, 23-28 June 2008.

[49] Annalisa Barla, Francesca Odone, Alessandro Verri, "Histogram Intersection Kernel for Image Classification," in *Proceedings ICIP 2003 International Conference on Image Processing,* Barcelona, Catalonia, Spain, 14-17 Sept. 2003.

[50] Navneet Dalal, Bill Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005),* San Diego, CA, USA, 25-25 June 2005.

[51] Dorin Comaniciu, Peter Meer, "Mean shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, 2002.

[52] S. Krig, Computer Vision Metrics: Survey, Taxonomy, and Analysis, Apress, June 2, 2014, p. 133.

[53] K. Fukunaga, L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory,* vol. 21, no. 1, pp. 32-40, Jan 1975.

[54] Markus Stricker, Michael Swain, "The Capacity of Color Histogram Indexing," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94.),* Seattle, WA, 21-23 Jun 1994.

[55] Greg Pass, Ramin Zabih, "Histogram Refinement for Content-Based Image Retrieval," in *Proceedings 3rd IEEE Workshop on Applications of Computer Vision (WACV '96.),* Sarasota, FL, 2-4 Dec 1996.

[56] S. K. Shevell, The Science of Color, Second ed., OSA and Elsevier, 2003, pp. 135-204.

[57] S. R. Buss, 3D Computer Graphics: A Mathematical Introduction with OpenGL, Cambridge, United Kingdom: CAMBRIDGE UNIVERSITY PRESS, 2003, pp. 146-154.

[58] J. Schewe, Digital Print: Preparing Images in Lightroom and Photoshop for Printing, Peachpit Press, 2014, p. 27.

[59] G. A. Agoston, Color Theory and Its Application in Art and Design, Second Completely Revised and Updated ed., Berlin, Heidelberg, New York: Springer-Verlag, 1987, p. 107.

[60] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Upper Saddle River, New Jersey: Pearson Prentice Hall, 2008, p. 402.

[61] B. Thaller, "Visualization of Complex Functions," Institute of Mathematics, University of Graz, Austria, [Online]. Available: http://www.mathematica-journal.com/issue/v7i2/articles/contents/thaller/html/.

[62] R.J. Qian, P.J.L. Van Beek and M.I. Sezan, "Image retrieval using blob histograms," in *2000 IEEE International Conference on Multimedia and Expo (ICME 2000),* 30 July-2 August, 2000.

[63] D.S. Messing , P. van Beek and J.H. Errico, "The MPEG-7 colour structure descriptor: image description using colour and local spatial information," in *2001 International Conference on Image Processing*, 7-10 October 2001.

[64] B. S. Manjunath,Philippe Salembier and Thomas Sikora, Introduction to MPEG-7: Multimedia Content Description Interface, John Wiley & Sons Ltd., 2002.

[65] Savvas A. Chatzichristofis and Yiannis S. Boutalis, "CEDD: Color and Edge Directivity Descriptor: A Compact Descriptor for Image Indexing and Retrieval," in *Computer Vision Systems*, vol. 5008, Springer Berlin Heidelberg, 2008, pp. 312-322.

[66] Savvas Chatzichristofis and Yiannis Boutalis, "A hybrid scheme for fast and accurate image retrieval based on color descriptors," in *ASC '07 Proceedings of The Eleventh IASTED International Conference on Artificial Intelligence and Soft Computing*, 2007.

[67] Dong Kwon Park, Yoon Seok Jeon and Chee Sun Won, "Efficient use of local edge histogram descriptor," in *Proceedings of the 2000 ACM workshops on Multimedia*, New York, USA.

[68] T. Joachims, Learning to Classify Text Using Support Vector Machines, New York: Springer Science+Business Media, 2002, pp. 9-15.

[69] Anna Bosch, Andrew Zisserman, Xavier Muñoz, "Scene Classification Via pLSA," in *Computer Vision – ECCV*, vol. 3954, Graz, Austria, Springer Berlin Heidelberg, May 7-13, 2006, pp. 517-530.

[70] Gabriella Csurka ,Christopher R. Dance, Lixin Fan, Jutta Willamowski and Cédric Bray, "Visual Categorization with Bags of Keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1-22.

[71] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi and Andrew Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2011.

[72] Anna Bosch, Andrew Zisserman, Xavier Munoz, "Image Classification using Random Forests and Ferns," in *IEEE 11th International Conference on Computer Vision (ICCV),* Rio de Janeiro, 14-21 Oct. 2007.

[73] Martin Loesdau, Sébastien Chabrier, Alban Gabillon, "Hue and Saturation in the RGB Color Space," in *Image and Signal Processing*, vol. 8509, Cherbourg, France, Springer International Publishing, June 30-July 2, 2014, pp. 203-212.

[74] Herve Jegou, Matthijs Douze, Cordelia Schmid, Patrick Perez, "Aggregating local descriptors into a compact image," in *CVPR 2010 - 23rd IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, United States, 13-18 June 2010.

[75] Relja Arandjelovic, Andrew Zisserman, "All About VLAD," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Portland, OR, 23-28 June 2013.

[76] Tewodros M. Dagnew and Umberto Castellani, "Supervised Learning of Diffusion Distance to Improve Histogram Matching," in *Similarity-Based Pattern Recognition*, vol. 9370, Springer International Publishing, 2015, pp. 28-37.

[77] Javed M. Aman, Ronald M. Summers and Jianhua Yao, "Characterizing Colonic Detections in CT Colonography Using Curvature-Based Feature Descriptor and Bag-of-Words Model," in *Virtual Colonoscopy and Abdominal Imaging. Computational Challenges and Clinical Opportunities*, vol. 6668, Springer Berlin Heidelberg, 2011, pp. 15-23.

[78] Taylor B. Arnold and John W. Emerson, "Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions," *The R Journal,* vol. 3/2, December 2011.

[79] Tien-Vu Nguyen, Dinh Phung and Svetha Venkatesh, "Topic Model Kernel: An Empirical Study towards Probabilistically Reduced Features for Classification," in *Neural Information Processing*, vol. 8227, Springer Berlin Heidelberg, pp. 124-131.

[80] Lu Bai, Edwin R. Hancock and Lin Han, "A Graph Embedding Method Using the Jensen-Shannon Divergence," in *Computer Analysis of Images and Patterns,* vol. 8047, Springer Berlin Heidelberg, 2013, pp. 102-109.

[81] Ann E. Watkins, Richard L. Scheaffer, George W. Cobb, Statistics: From Data to Decision, Second ed., John Wiley & Sons, 2011, p. 596.

[82] S. G. Tzafestas, Advances in Intelligent Systems: Concepts, Tools and Applications, Springer Science+Business Media Dortrecht, 1999, p. 234.

[83] Michele d'Amico, Patrizio Frosini, Claudia Landi, "Using matching distance in Size Theory: a survey," *International Journal of Imaging Systems and Technology,* vol. 16, no. 5, p. 154–161, 9 March 2007.

[84] C. M. Bishop, Pattern Recognition and Machine Learning, New York, NY, USA: Springer-Verlag, 2006, pp. 380-381.

[85] "Code for the Edge Detection and Image SegmentatiON system," 14 April 2003. [Online]. Available: http://coewww.rutgers.edu/riul/research/code/EDISON/index.html. [Accessed 26 09 2016].

[86] Radu Timofte, Karel Zimmermann and Luc Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Machine Vision and Applications,* vol. 25, no. 3, pp. 633-647, April 2014.

[87] Q. Wang and Z. Wang, "A Subjective Method for Image Segmentation Evaluation," in *Asian Conference on Computer Vision (Springer)*, 2009.

[88] C.-W. Wang, "A Bayesian Learning Application to Automated Tumour Segmentation for Tissue Microarray Analysis," in *Machine Learning in Medical Imaging*, Beijing, China, Springer Berlin Heidelberg, 2010, pp. 100-107.

[89] I. Menken, Cloud Testing Complete Certification Kit - Core Series for IT, Emereo Publishing, 20 March 2013.

[90] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, Andrea Vedaldi, "Describing Textures in the Wild," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 23-28 June 2014.