CLASSIFIER FUSION METHODS ADAPTED TO TEMPLATE MATCHING ON SATELLITE IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET ERSÖZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2013

Approval of the thesis:

CLASSIFIER FUSION METHODS ADAPTED TO TEMPLATE MATCHING ON SATELLITE IMAGES

Submitted by **AHMET ERSÖZ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering, Middle East Technical University,** by,

Prof. Dr. Canan Özgen ______________
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan ______________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Dr. İlkay Ulusoy ______________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members

Prof. Dr. Gözde Bozdağı Akar ______________
Electrical and Electronics Engineering Dept., METU

Assoc. Dr. İlkay Ulusoy ______________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu ______________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Aydın Alatan ______________
Electrical and Electronics Engineering Dept., METU

Dr. Erdem Akagündüz ______________
Senior Design Engineer, ASELSAN

Date: 28.08.2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Lastname : Ahmet Ersöz

Signature :

ABSTRACT

CLASSIFIER FUSION METHODS ADAPTED TO TEMPLATE MATCHING ON SATELLITE IMAGES

Ersöz, Ahmet

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Dr. İlkay Ulusoy

September 2013, 122 pages

Classifier combination and selection methods are becoming popular in vision research. Classifier fusion studies started to take the place of continuous development of new algorithms. In this study, template matching methods are used as classifiers and the results of the template matching methods from satellite images are taken as input to the fusion center. Template matching methods are adapted to different classifier fusion methods. In literature, there is not any performance measurement standard for the binary template matching output images. In order to analyze and compare the template matching methods and the classifier fusion methods, two performance measurement methods are proposed. In one of them, pixel-by-pixel intersection of the output image and the ground truth image are considered. In the other method, the output image is considered as a set of objects and the intersection of the object in the output image and the ground truth image is analyzed. The individual performance of the template matching methods is highly dependent on the threshold values used in the methods. When combining the template matching results, choosing optimal threshold is important to analyze the performance of the classifier fusion method. There have been very few studies for optimizing the fusion system performance and the studies have only been presented on decision level fusion. As a final task, a method for optimizing the performance in score (raw output) level fusion system is proposed. The results are quite promising such that the outputs of the proposed method outperformed to most of the score level fusion methods in the literature.

Keywords: Template Matching, Classifier Fusion, Classifier Selection, Performance Measure, Optimizing System

ÖZ

UYDU GÖRÜNTÜLERİ İÇİN SINIFLANDIRICI KAYNAŞTIRMA YÖNTEMLERİNİN ŞABLON EŞLEME METODLARINA ADAPTASYONU

Ersöz, Ahmet

Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Assoc. Dr. İlkay Ulusoy

Eylül 2013, 122 sayfa

Sınıflandırıcı birleştirme ve seçme üzerine olan çalışmalar son yıllarda çok popüler olmuştur. Yeni algoritma geliştirme yaklaşımı yerini olan algoritmaları birleştirip daha başarılı bir sistem oluşturma yaklaşımına bırakmaktadır. Bu çalışmada şablon eşleme metodları sınıflandırı olarak ele alınmış ve uydu görüntüsüne uygulanan şablon eşleme metodlarının sonuçları sınıflandırıcı kaynaştırma yöntemlerine girdi olarak verilmiştir. Şablon eşleme yönteminin kullanılan sınıflandırıcı kaynaştıcı yöntemine göre adaptasyonu sağlanmıştır. Literatürde uydu görüntüsüne uygulanan şablon eşleme yöntem sonuçları için ortak bir performans ölçüm metodu bulunmadığı için iki performans ölçüm yöntemi önerilmiştir. Bu metodlardan ilkinde çıktı görüntüsü ile kesin sonuç görüntüsündeki pikseller arası örtüşme göz önünde bulundurulmuştur. Diğer yöntem ise çıktı görüntüsünün nesnelerden oluştuğunu varsayıp çıktı görüntüsü ile kesin sonuç görüntüsündeki nesnelerin örtüşmesini ölçer. Şablon eşleme yöntemlerinin başarısı algoritma içinde seçilen eşik değerine bağımlıdır. Şablon eşleme yöntemlerini birleştirirken algoritmalar için bireysel olarak optümum eşik değerleri seçilmesi kullanılan sınıflandırıcı kaynaştırma metodunun başarısını analiz etmek açısından çok önemlidir. Literaturde kaynaştırıcı sistemin başarısını optimumlaştırmak adına çok fazla çalışma yer almamaktadır ve olan çalışmalar da genellikle ikili karar seviyesindeki çıktılar için yapılmıştır. Bu çalışmada algoritmaların skor (ham) seviyesindeki çıktıları için optimumlaştırma yöntemi önerilmiştir. Yapılan testlerde umut verici sonuçlar elde edilmiştir. Önerilen metodda, literaturdeki skor seviyesindeki sınıflandırıcı kaynaştırıcı metodlarının çoğuna göre daha başarılı sonuçlar vermiştir.

Anahtar Kelimeler: Şablon eşleme, sınıflandırıcı kaynaştırıcı, sınıflandırıcı seçme, performans ölçümü, sistem optimumlaştırma

To Mom, Dad and Sister

ACKNOWLEDGEMENTS

I would like to express my gratitude and deep appreciation to my supervisor Associate Professor İlkay Ulusoy for her guidance, positive suggestions and support in this study.

I would like to also acknowledge The Scientific and Technological Research Council of Turkey (TUBITAK) for its funds.

Moreover, I would like to thank to Aselsan R&D for the great research environment it had provided.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES:

LIST OF TABLES

TABLES:

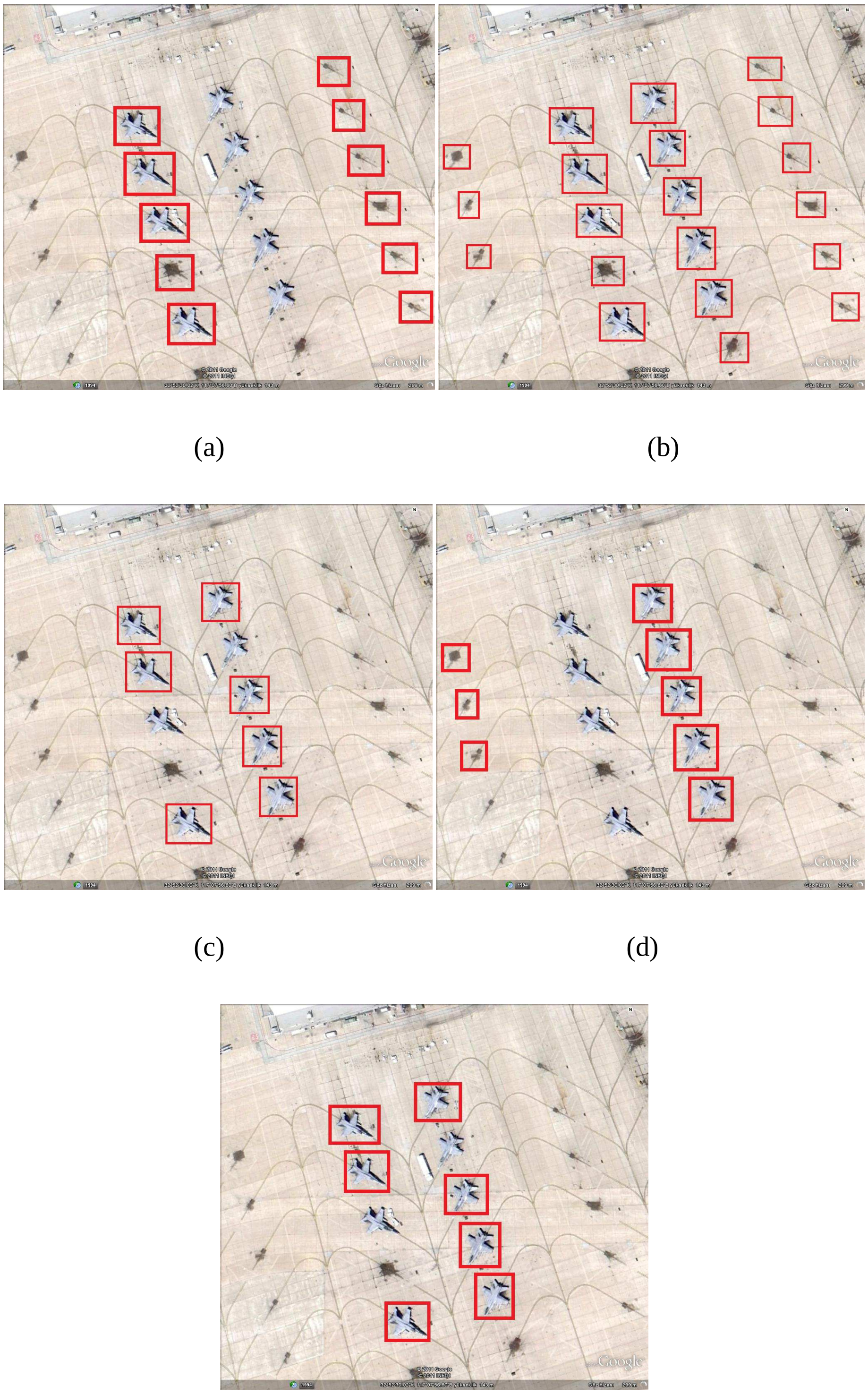
CHAPTER 1

INTRODUCTION

1.1. MOTIVATION

Decision making can be defined as a process of selecting a logical choice from the available options. Identifying the values, uncertainties and other issues relevant in a given decision is important to concern decision's rationality, and examining its optimality. Generally, searching the optimal decision is an everyday problem of all people in their daily lives (Figure 1) and this problem is addressed in many disciplines including economics, psychology, philosophy, mathematics, statistics, engineering etc. In decision theory, it is assumed that an ideal decision maker is fully informed and able to compute with perfect accuracy and fully rational. In engineering applications, providing fully informed system is generally impossible to satisfy. Every decision support system has some limitations and all decisions include some degree of uncertainties and error rates. The common approach to obtain more accurate decisions is to progressively improve the decision maker systems or provide a creative model to increase the performance. But, it is very hard to design a decision maker optimal in every condition. Progressive improvement sometimes reaches the limits. In order to solve this problem, decision fusion approach is developed. In this approach, decisions of different decision makers are integrated to obtain improved performance.

Imaging technologies have been developed extremely during the last decades. Object recognition, identification, classification, tracking and template matching became the hot topics in computer vision area. In a digital image processing applications, the method developed for the specific task can be interpreted as a decision maker. Different methods developed for the same purpose provide the set of decision makers and the optimal decision may be obtained from the fusion of these decision makers. For example, detection of war crafts from a satellite image (Figure 1). There may be more than one detection algorithms, each one of which produces a different detection result and these results should be fused to obtain detection result, i.e., the result which is closest to the ground truth, if there is any.



(a) (b)

(c) (d)

(e)

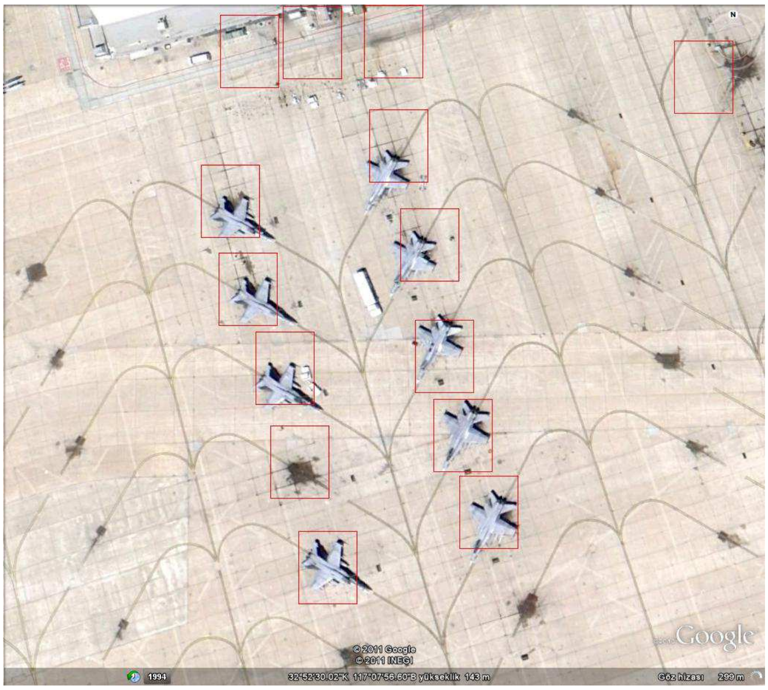
Figure 1: Example war craft detection from satellite images by four different decision maker (a), (b), (c), (d) and the fused output of them (e)

1.2. SCOPE OF THE THESIS

In this study, we aim to perform decision fusion to template matching methods. Template matching is a digital image processing technique and widely used in target detection and many other areas. A template is given and the regions in the image, which are similar to this template, are searched throughout the image (Figure 2). Many methods, which utilize different features and different similarity measurement, have been developed in the literature for template matching applications. As in the other image processing applications, developing a best method for every input type is a very difficult task to achieve. In this study, in order to improve the success rate of the template matching methods, classifier fusion methods are adapted for template matching methods. Besides, a new classifier fusion method, which depends on optimization, is developed for the same purpose. Outputs of template matching methods and classifier fusion methods for an example test image and template are given in Appendix C.

(a) (b)

Figure 2: Example template matching image with input image (a), template (b) and the resulting image (c)



(c)

Figure 2: Example template matching image with input image (a), template (b) and the resulting image (c) (continued)

It is also an important problem to evaluate the performances of template matching methods. Determining the ground truths as well as the hit and miss conditions need to be handled carefully. If the results of template matching, fusion and the ground truth are given as binary images, by checking the intersection of these gives some opinion about the performance of the algorithm. However, by directly comparing the result and the ground truth pixel by pixel, the objects in the image are considered as the combinations of pixels and thus every pixel becomes very important and directly affects the success rate. Ground truth boundaries are also important in this kind of pixel based approaches. If the object's representation in the ground truth has loose boundaries, this may cause false increases in the hit and miss rate (Figure 3). If the boundaries are so tight, this may cause false increases in the false alarm rate, although the algorithm detects the object to some level (Figure 4).

(a) (b)

Figure 3: Example for loose boundaries in the ground truth image (a). Ground truth boundaries are illustrated with red colored window and detection methods output is shown with blue color. In the performance measure image (b), hit pixels are shown with green and miss are shown with red color**.**

(a) (b)

Figure 4: Example for tight boundaries in the ground truth image (a). Ground truth boundaries are illustrated with red colored window and detection methods output is shown with blue color. In the performance measure image (b), hit, miss and false alarms are shown with green, red and blue color respectively.

Instead of pixel based approaches, objects in the image may be considered as a whole, for example as connected components, and the hit and miss rates may be measured by the intersection of the connected components in the ground truth and algorithm’s output image. If the intersection rate is above a threshold value, then this target may be interpreted as detected. Otherwise, it is interpreted as missed. But, the drawback of this method is that, threshold value directly affects the performance measure. If the center point of the detected object in the algorithm’s output image is varied from the center location of that object in the ground truth image, but still intersection rate is above the threshold, this variation is not punished and cannot be recognized from the performance measure outputs (Figure 5). Thus, in order to analyze the performances of the template matching methods and also the fusion results in detail, both of the performance measurement approaches are used. These methods are named as pixel-based approach and object-based approach and explained in the following chapters.

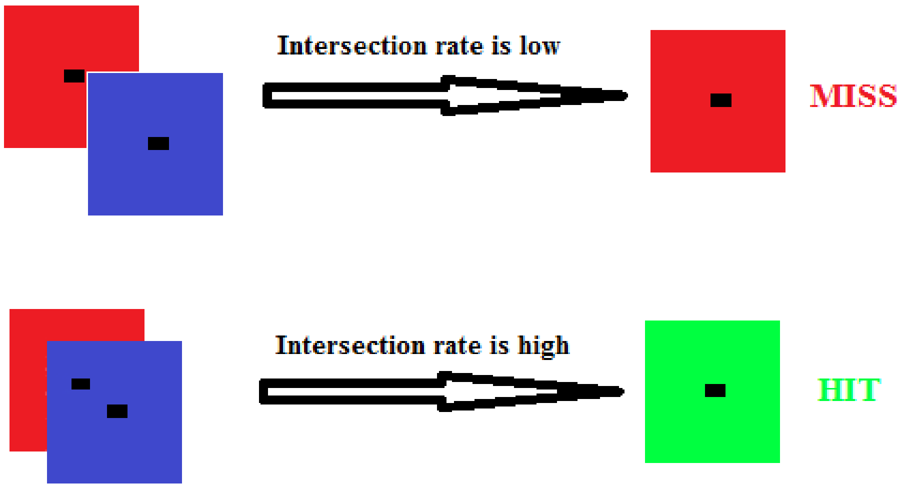


Figure 5: Miss and hit conditions in performance measure by considering the objects in the image as a whole is illustrated. Red and blue objects denote the objects in the ground truth image and output image respectively. Although the center points are varied from each other, the intersection rate determines hit or miss.

1.3. OUTLINE OF THE THESIS

In Chapter 2 of the thesis, performance measurement methods are covered. In all of the chapters following Chapter 2, the experimental results are presented based on the precision, recall and f-measure values discussed in Chapter 2.
In Chapter 3, template matching methods are discussed. First, a literature review and the related works are summarized. Then, four major template matching methods which are correlation based template matching, edge based template matching, histogram based template matching and angular radial transform template matching are explained. After covering each method, experimental results of that method are presented by the two performance measurement methods.

Chapter 4 includes classifier fusion methods. As in Chapter 3, first the classifier fusion methods developed so far are summarized. Then, the methods, which are most appropriate to be used for template matching, are discussed. The way in which they are adapted to the template matching applications is emphasized. At the end of each classifier fusion method, experimental results related to that method are presented.

Finally, comparison, conclusion, discussion and suggested future works are in Chapter 5.

CHAPTER 2

PERFORMANCE MEASUREMENT APPROACHES

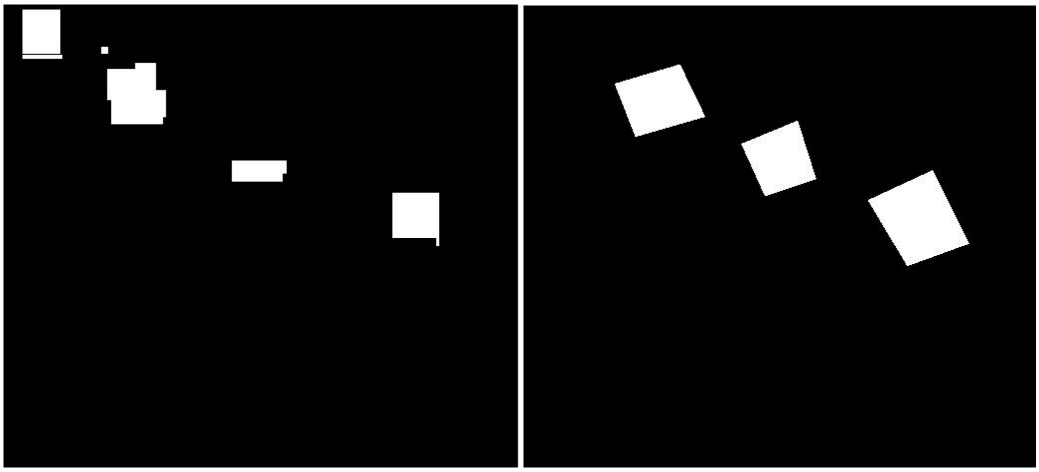
2.1. INTRODUCTION

Performance measurement is an important task to analyze and compare the decision making systems. In this study, focus is on the measurement of the hit and miss rates of the template matching and fusion algorithm outputs. Both the ground truth images and the algorithms' outputs are given as binary images.

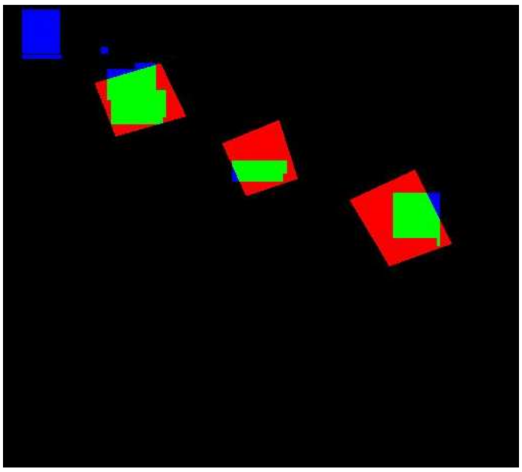
Template matching methods draw a rectangle around a center point where they detect the template object. Ground truth images are formed in a similar way, i.e., a rectangle is drawn at the object position on the binary ground truth image. Two performance measurement methods are developed. One of them is pixel based, and the other one is object based.

2.2. PIXEL BASED PERFORMANCE MEASUREMENT

In pixel based performance measurement method, the intersection of the output image and the ground truth image is considered in a pixel wise way. When looking the intersection of the output image and the ground truth image, pixel values in the same location of both images are compared. But, this approach does not give enough information about the performance of the fusion results. Since, when fusing the outputs of the template matching methods, rectangular outputs are disrupted. Although, the resulting image gives hit pixels very near to the hit pixels in the ground truth image, the fusion result is highly punished due to small location variance. As shown in Figure 6, even one pixel distance variations are punished as miss or false alarm. In order to handle the neighborhood relations, nearest neighbor values are also taken into consideration. Let (u,v) be the location of the pixel in consideration, the nearest neighbors of that pixel are analyzed and the number of pixels having positive value are calculated. Similarly, the pixel in (u,v) location and nearest neighbors of that pixel in the ground truth image are analyzed and again the number of pixels having positive value is calculated. If the numbers calculated from both the resulting image and the ground truth image are above a threshold value, hit condition occurs in that neighborhood (Figure 7). If the number of positives in the resulting image is above the threshold but the number of positives in the ground truth image is below the threshold, false alarm condition occurs (Figure 8). If the number of positives in the resulting image is below the threshold, but the number of positives in the ground truth image is above the threshold, a miss condition occurs in that neighborhood (Figure 9).



(a) (b)



(c)

Figure 6: Disrupted fused output image (a), ground truth image (b) and performance measure output (c) by illustrating the hit, miss and false alarm with green, red and blue color respectively

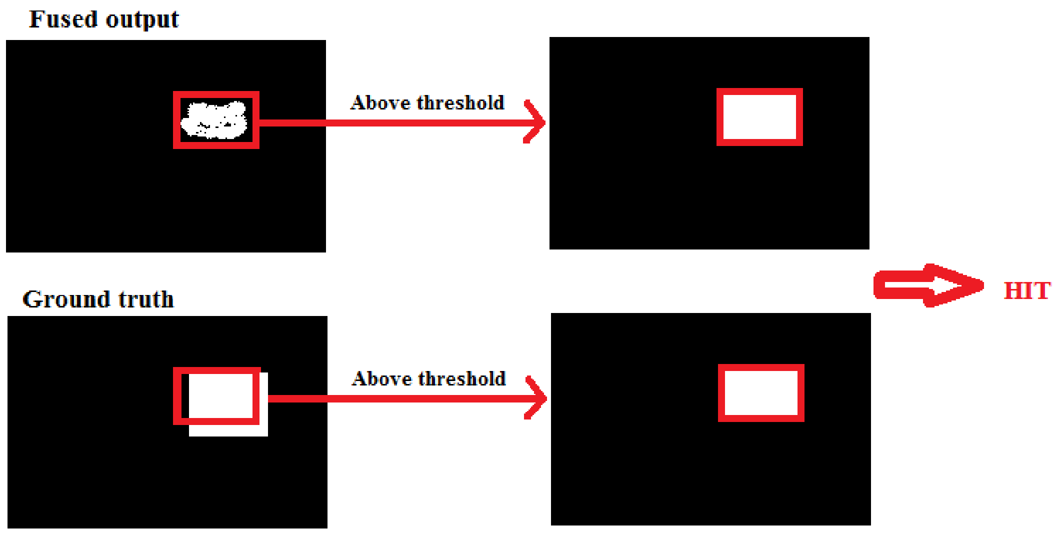


Figure 7: Hit condition in pixel based performance measurement is illustrated. The center point of the red window is the pixel in consideration and all the pixels enclosed by that window are the nearest neighbors of it.

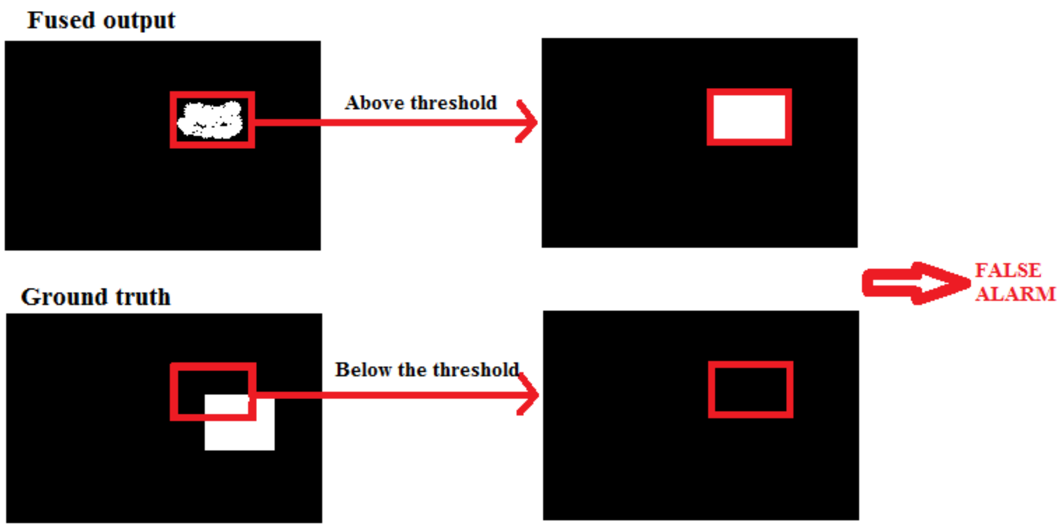


Figure 8: False alarm condition in pixel based performance measurement is illustrated.

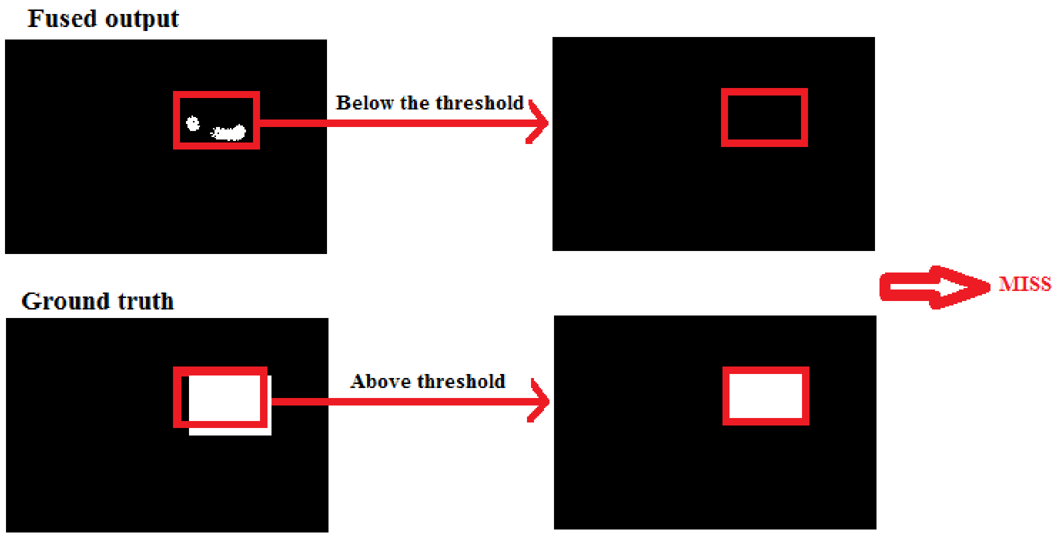
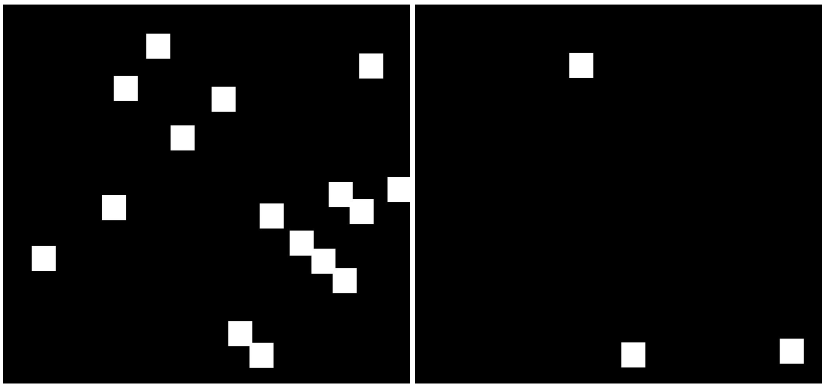


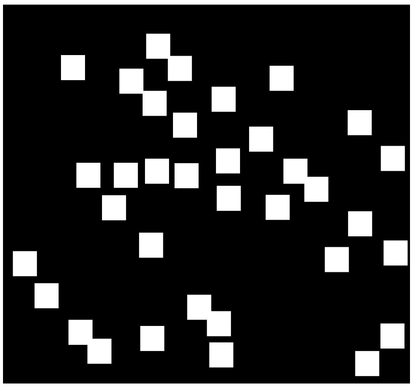
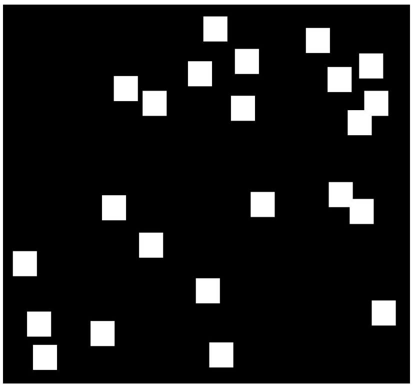
Figure 9: False alarm condition in pixel based performance measurement is illustrated.

Experiments are performed for different nearest neighbor sizes. According to empirical results, choosing neighbor size as 20 seems appropriate for the train and test images used in this study. An example binary fusion result image, ground truth image and the pixel based performance measure illustration are given in Figure 10. Hit, miss and false alarm conditions are shown with green, red and blue colors respectively. As shown in Figure 10(e), the organized rectangular outputs are disrupted in the fusion result of the four template matching methods.

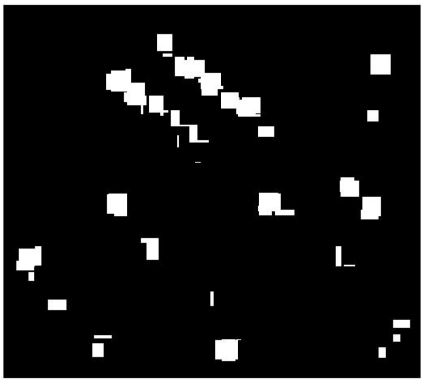
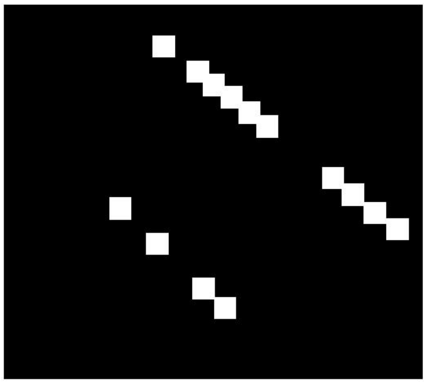


(a) (b)

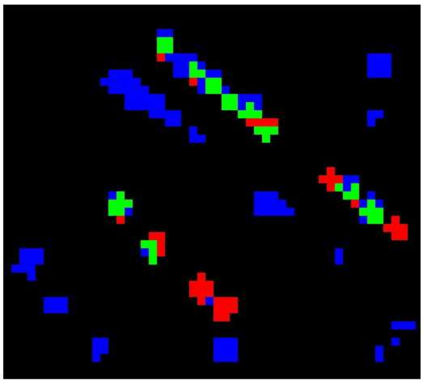
Figure 10: Example outputs of four methods (a), (b), (c), (d), resulting image of them after fusion(e), ground truth image (f), and the illustration of the performance measure (g) by representing the hit conditions by green, miss conditions by red and false alarms by blue color are given

(c) (d)

(e) (f)



(g)

Figure 10: Example outputs of four methods (a), (b), (c), (d), resulting image of them after fusion(e), ground truth image (f), and the illustration of the performance measure (g) by representing the hit conditions by green, miss conditions by red and false alarms by blue color are given (continued)

2.3. OBJECT BASED PERFORMANCE MEASUREMENT

In this approach, each object is considered as a whole instead of as a set of pixels. Instead of measuring the intersection of the object in the ground truth and the resulting image by looking the hit pixels in both images, decision analysis is made on whether the object which is represented as a rectangular connected component in the ground truth image can be taken as detected or not according to the binary resulting image. As mentioned before, template matching methods find a center point and draw a rectangle around that center in their binary output image. Fusion operations are performed in pixel level and the resulting images may disrupt the organized rectangular output with specific center points. But, in the fusion result,, at least one of the center points given as detection by the template matching methods are preserved. The preserved center points are found at the resulting image and rectangular windows are drawn around these center points. Finally, both the resulting image and the ground truth image have rectangular connected components. Then, the performance measure is performed by the intersection rate of these connected components. Each connected component is considered as a single object. By masking all other connected components in the resulting image, the intersection with the ground truth image is measured. If the intersection rate is above a threshold value, then this connected component is considered as a hit. Otherwise, this connected component is considered as false alarm. In order to detect the miss conditions, each connected component in the ground truth image is considered one by one. When one of them is considered, the other connected components are masked and the intersection rate of that image with the resulting image is calculated. If the intersection rate is below the threshold value, it is considered as that object in the ground truth is missed. An example representation of converting disrupted fusion result to an output image with the rectangular representation of the object is shown in Figure 11. Example performance measure is shown in Figure 12. Again the hits are shown with green color, misses with red color and false alarms with blue color.

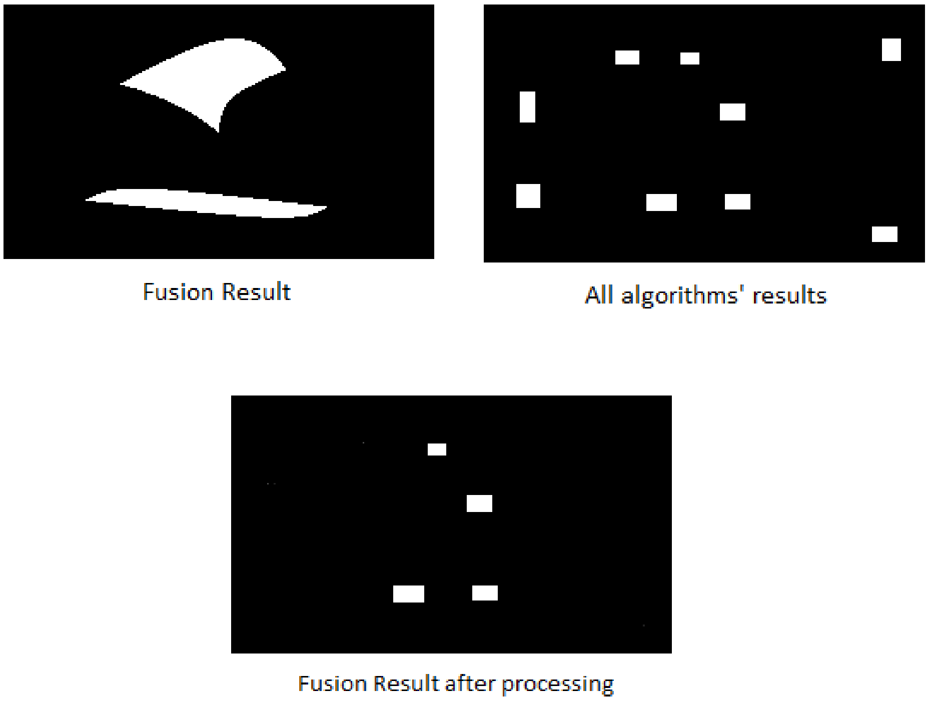
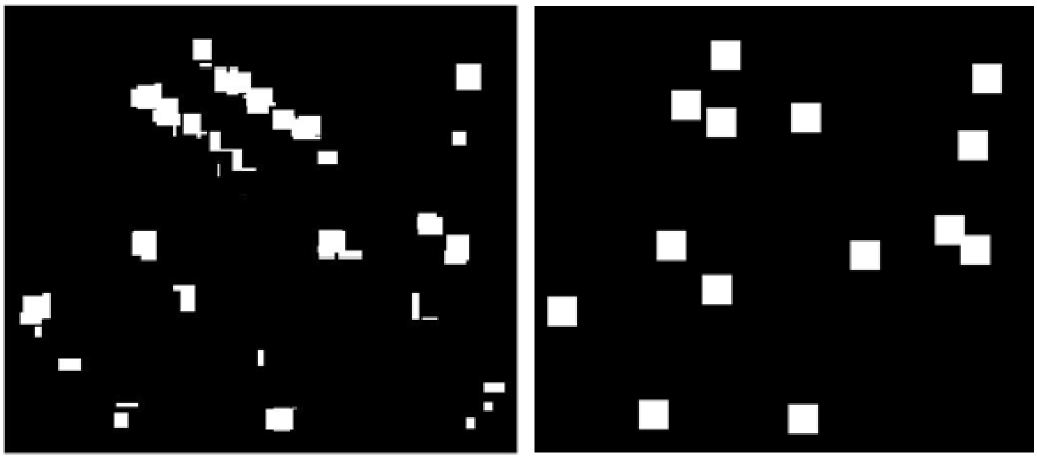
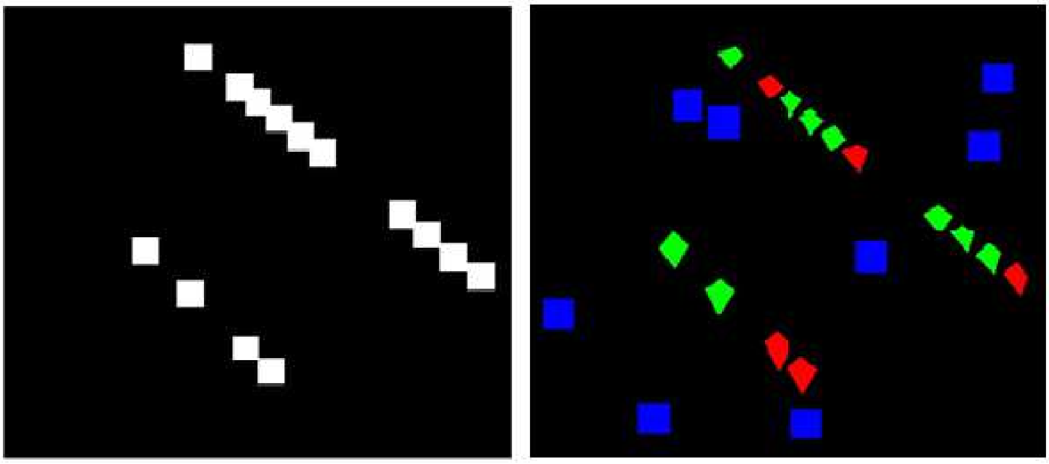


Figure 11: Converting fusion result to oriented rectangular representation for each object



(a) (b)

Figure 12: Example pixel based performance measure by binary fusion resulting image (a), image converted object based representation, (b), ground truth image (c), and the illustration of the performance measure (d) by representing the hit conditions by green, miss conditions by red and false alarms by blue color



(c) (d)

Figure 12: Example pixel based performance measure by binary fusion resulting image (a), image converted object based representation, (b), ground truth image (c), and the illustration of the performance measure (d) by representing the hit conditions by green, miss conditions by red and false alarms by blue color (continued)

When analyzing the performances of the template matching methods and the fusion methods, precision, recall and f-measure values are taken into consideration. Let N_A is the total detection instances by the algorithm, N_{GT} is the total object instances in the ground truth and $N_{\cap}$ is the total instances where the algorithm's output and the ground truth intersect. Then, precision is the ratio of the intersected result to the total detection instances by the algorithm. Recall is the ratio of the intersected result to the total instances in the ground truth. In other words, by precision, we measure the probability of an instance given as detected by the algorithm is actually an object. By recall, we measure the probability of detection of an object which actually exists in the ground truth. Combining the precision and recall values for measuring the overall performance of the system is very important. For that purpose, a measure called f-measure is used. F-measure combines the precision and recall by calculating the harmonic mean of precision and recall. The formulation of precision, recall and f-measure are given in *(2.1)*, *(2.2)* and *(2.3)* respectively.

$$Precision = \frac{N_{\cap}}{N_A} \qquad \textbf{(2.1)}$$

$$Recall = \frac{N_{\cap}}{N_{GT}} \qquad \textbf{(2.2)}$$

$$FMeasure = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad \textbf{(2.3)}$$

CHAPTER 3

TEMPLATE MATCHING METHODS

3.1. INTRODUCTION AND RELATED WORK

Template matching is one of the revolutionary concepts in computer vision. It has wide-spread applications in robotics, automatic target recognition, image registration etc. In general, template matching is a classification method, which compares portions of image, called as the template, with another image [1]. Template matching methods may be divided into two categories according to the approaches used. One of them is area based approaches and the other one is feature-based approaches. Area-based approaches are generally correlation-like methods. The similarity values are calculated according to the intensity values of both image and template. Squared differences in fixed intensities, correction-based methods, optimization methods and mutual information are the mostly used techniques in area-based approach. Feature-based approaches are used when template and input image have more correspondence with respect to features and control points. Feature-based methods are utilized to locate the pair-wise connection between reference and template by using the spatial relations or descriptors of features. Both in the area-based and feature-based approaches, template matching problem is reduced to similarity measure in the end. A great number of techniques have been developed to measure the similarities between the input image and the template. One of the most known similarity measure method is sum of absolute intensity differences which is defined by Devijver and Kittler [2]. Geometric distance is a preferred technique for similarity measure, when both template and image has binary structures [1]. Average distance is also used by determining the closest structure points between the image and the template. Then, the average of the distances between the corresponding points is used in the similarity measure. In mutual information method, correlation in intensities of both image and the template is calculated by dealing out all voxels in template and the image. The similarity measure approaches discussed so far are ineffective in template matching when matching images have rotational differences. Invariant moment method is free from orientation. In this method, invariant of the position and orientation of a pattern is obtained by normalizing the moments [1]. Sum of squared distances is another popular similarity measure method. It is widely used in image matching applications such as tracking and stereo matching. This method is very sensitive to outliers and is not robust to template variations [3]. Hausdorff distance measure reduces the effect of outliers and is quite tolerant of small position errors [4]. Given two finite point sets A = $\{a_1, \ldots\ a_p\}$ and B = $\{b_1, \ldots\ b_p\}$, the mathematical definition of Hausdorff distance is defined as

$$H(A, B) = \max\left(h(A, B), h(B, A)\right) \tag{3.1}$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||\, a - b|| \tag{3.2}$$

Four most popular template matching approaches are normalized correlation-based template matching, histogram based template matching, edge based template matching and angular radial transform template matching.

Normalized correlation matching is the most popular method in target tracking algorithms. Direct image information collected from all pixels, such as image brightness, is used in normalized correlation method in order to minimize the error measure [5]. Although, the normalized cross correlation is a reasonable choice in template matching applications, it is computationally expensive. In order to reduce the computational complexity, fast normalized cross correlation is proposed [6]. The basic idea of the fast normalized cross correlation is to represent the normalized cross correlation as a sum of rectangular basis functions. Let the template image be represented by t and $\bar{t}$ is the mean value of the template. Then, the zero mean template function t' shifted by u steps in the x direction and by v steps in the y directions is defined as *(3.3)*.

$$t'(x - u, y - v) = t(x - u, y - v) - \bar{t} \tag{3.3}$$

The basic idea to simplify the calculation of the normalized cross correlation is to expand the zero mean template function t'(x,y) to the weighted sum of K rectangular basis functions t_i, yielding an approximation $\check{t}$(x,y) of the template function as given in *(3.4).*

$$\check{t}(x, y) = \sum_{i=1}^{K} k_i t_i(x, y) \tag{3.4}$$

For automatic determination of the basis functions, the quadratic criterion given in *(3.5)* is used to assess the quality of the approximation.

$$J = \sum_{x,y} (t'(x, y) - \check{t}(x, y))^2 \tag{3.5}$$

A recursive algorithm is proposed to determine the basis functions. Algorithm divides the template function t(x,y) into rectangular basis functions. It starts with a single basis function t_1(x,y) = $\bar{t} = constant$ and calculates J using *(3.4)* and *(3.5)*. If $J > J_{max}$, where

J_{max} is a predefined threshold, the basis function is divided into two basis functions and the coefficients k_i, i = 1,2 are recalculated under the condition that J is minimized. This process is continued recursively for each basis function t_i until $J < J_{max}$. By this approximation, the number of calculations required to evaluate the normalized cross correlation depends linearly on the number of basis functions used, but not on the size of the template t. For example, a video graphics array (VGA) camera image with the image function size 640x480 pixel and a template function 64x64 pixel, the number of multiplications for the numerator of correlation function is reduced 2048 times in fast normalized cross correlation compared to the direct calculation of the normalized cross correlation. Fourier-transform based implementations of the normalized cross correlation algorithm are also implemented to reduce the number of calculations. Fast normalized cross correlation can outperform the Fourier-transform based implementations. For the VGA image (640x480) and template image (64x64) mentioned before, the number of multiplications required for the numerator of the correlation is reduced 47 times in fast normalized cross correlation compared to the Fourier-transform based correlation calculation.

Edge-based template matching is one of the feature template matching methods. Feature template matching methods minimize the error measure based on geometrical constraints between corresponding features in input image and the template. Edges are one of the features. In edge based template matching methods, edge intensity value is used for each pixel [7]. Edge intensity value of the pixel can be obtained by edge detection algorithm. At the present, there are a lot of methods of edge detectors [8]. Kirsch edge detector is one of the important edge detectors. In Kirsch edge detector, eight directional derivatives of every pixel are obtained and the value of the maximum directional derivative is used as edge intensity [1].

In histogram based methods, the aim is obtaining a distinctive histogram to represent the object or template. Color histogram may be used as a distinctive histogram. Matching is performed by searching the similar region in the image, whose histogram best matches the object or template in the input image [9]. The general approach in histogram based methods is focusing on the distribution of features at each pixel. The information contained in each pixel may consist of the color intensity or certain other features like the gradient [9]. Although color provides high discriminative power, the most existing template matching methods were designed in gray scale. The reason of that is the problem in color constancy. Illumination changes also worsen the performance of pattern recognition algorithms [10]. Some studies have been made to overcome these difficulties and use color as a feature in template matching. C-color-SIFT is one of the important algorithm [11] which combines SIFT descriptor with a set of color invariants proposed by Geusebroak [12]. Color-Ciratefi is also a newly developed template matching method based on grayscale template matching Ciratefi technique (Circular, Radial, and Template Matching Filter) [13]. According to experiments of Araujo et al [10], Color-Ciratefi produces more accurate results compared to C-color-SIFT algorithm. Moreover, Color-Ciratefi is robust to minor viewpoint variations and blur.

The Angular Radial Transform (ART) was adopted in MPEG-7 as a regional based shape descriptor and widely used in human face detection applications. Histogram equalized intensity map and local intensity map of a potential object or face pattern are created and applied to the ART to derive the compact representation of the pattern. Then, ART coefficients are obtained and given to the support vector machine (SVM) to determine the existence or not of the pattern in the image [14].

Although many studies conducted on finding the best method in template matching, there is no winning method for every input type. The methods perform differently for different feature sets. Due to the nature of methods' dependency to the input set, some studies performed for choosing the best classifier for the given input set or combining the outputs of different methods and taking the decision of combination.

3.2. TEMPLATE MATCHING METHODS

Correlation based, edge based, histogram based and ART template matching methods were covered as template matching algorithms. Edge and histogram based methods are feature-based approaches and correlation based method is area based approach. Before calculating the similarity, a template image is determined according to the chosen object. This object is enclosed with a rectangular boundary. Then, in the test image, this rectangular template is moved all over the image and the similarity is measured in everywhere. The region where the similarity value is above a threshold value is detected. Threshold selection is also an important problem. High threshold value may cause high false negative results and low threshold value may cause high false positive results. The images and templates for each of them are given in Appendix A. The results of the template matching methods and classifier fusion methods by object based approach for an example image are given in Appendix C.

3.2.1. Correlation based template matching method

Normalized cross correlation is widely used method in template matching problems. Let the given image be f and the intensity value of the image of the size M_xxM_y at the point (x,y) $x \in \{0, \dots, M_x - 1\}$, $y \in \{0, \dots, M_y - 1\}$, be denoted by f(x,y). Let the template image is represented by t and the size of it is given as $N_x x N_y$. The problem in template matching is to determine the position of a given pattern in the input image f. In order to calculate the position of the pattern (u_{pos}, v_{pos}) in the input image f, the normalized cross correlation value γ at each point for f and the template image t is calculated. The normalized correlation method is defined as [6]:

$$\gamma = \frac{\sum_{x,y}\left(f(x,y)-\overline{f_{u,v}}\right)\left(t(x-u,y-v)-\bar{t}\right)}{\sqrt{\sum_{x,y}\left(f(x,y)-\overline{f_{u,v}}\right)^2 \sum_{x,y}\left(t(x-u,y-v)-\bar{t}\right)^2}} \qquad \textbf{\textit{(3.6)}}$$

$\overline{f_{u,v}}$ denotes the mean value of f(x,y) within the area of the template t shifted to (u,v) and is calculated as in *(3.7)*.

$$\overline{f_{u,v}} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x,y) \qquad \textbf{\textit{(3.7)}}$$

$\bar{t}$ is the mean value of template t similarly. By sliding the template image on the input image, the similarity degree at each location can be calculated. Then, the matching locations can be obtained by choosing the locations where the similarity value is above the threshold value.

In this study, the normalized cross correlation template matching is performed and the experiment results measured by the pixel and object based approaches are given in Table 1, Table 2, Table 3 and Table 4.

Table 1: Precision, Recall and F-Measure by the Pixel Based Approach

Image No	Precision	Recall	F-Measure
1	0.17	0.65	0,27
2	0.17	0.42	0,24
3	0.81	0.68	0,74
4	0.21	0.57	0,30
5	0.17	0.27	0,21
6	0.18	0.67	0,28
7	1	0.41	0,58
8	0.30	0.89	0,44
9	0.45	1	0,62
10	0.05	1	0,09
11	0.19	0.71	0,31
12	0.12	0.31	0,17
13	0.11	0.70	0,19
14	0.20	0.80	0,32
15	0.21	0.56	0,31
16	0.20	0.78	0,32
17	0.52	0.83	0,64
18	0.71	0.97	0,82

Table 2: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,32	0,68	0,38
Standard Deviation	0,27	0,22	0,21

Table 3: Precision, Recall and FMeasure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.19	0.75	0,30
2	0.19	0.60	0,29
3	1,00	0.71	0,83
4	0.25	1	0,40
5	0	0	0
6	0.20	1	0,33
7	1	0.50	0,67
8	0.50	0.83	0,63
9	1	1	1
10	0.08	1	0,14
11	0.20	1	0,33
12	0.17	0.50	0,25
13	0.17	1	0,29
14	0.25	1	0,40
15	0.50	0.57	0,53
16	0.52	0.86	0,65
17	0.67	1	0,80
18	1	1	1

Table 4: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,44	0,80	0,49
Standard Deviation	0,35	0,28	0,29

3.2.2. Edge based template matching

Edge detection is an important task in pattern recognition for discriminating the objects from the background [4]. The purpose of edge detection is to reduce the amount of data in the image significantly while preserving the structural properties to be used for further processing. As mentioned before, there are many edge detection methods. The edge detection method used in template matching in this paper is the Canny edge detector. Canny edge detector is one of the standard and successful edge detector methods. Canny's motivation for developing the method was to maximize the signal-to-noise ratio by increasing the probability of detecting real edges and minimizing the probability of falsely detected non-edge points, and locate the detected edges as close as possible to the real edge points. Canny edge detection algorithm runs in five separate steps: smoothing, finding gradients, non-maximum suppression, double thresholding and edge tracking by hysteresis. All images taken from a camera includes some amount of noise and smoothing is performed to reduce the noise by blurring the image. Smoothing is performed by a Gaussian filter. The kernel of the Gaussian filter with a standard deviation of σ = 1.4 is given in *(3.8)*.

$$B = \frac{1}{159} * \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \tag{3.8}$$

In finding gradients step, the aim is to find the points where the intensity change is high. In order to find the gradient points, two Sobel operators as shown in Figure 13 are used. By applying Sobel operators, the gradients in the x and y directions are found respectively. Gradient magnitudes can be calculated by the law of Pythagoras *(3.9)* or Manhattan distance measure *(3.10)*. The computational complexity of Manhattan distance is lower than Pythagoras calculation.

-1	0	1
-2	0	2
-1	0	1

S_x

1	2	1
0	0	0
-1	-2	-1

S_y

Figure 13: Convolution masks in Canny method [8]

$$|G| = \sqrt{G_X^2 + G_Y^2} \tag{3.9}$$

$$|G| = |G_X| + |G_Y| \tag{3.10}$$

In order to indicate exactly where the edges are, direction of the edges are determined and stored as given in *(3.11)*.

$$\theta = \arctan\left(\frac{|G_Y|}{|G_X|}\right) \tag{3.11}$$

After finding the gradient magnitudes and gradient directions, non-maximum suppression is applied. In non-maximum suppression, the blurred edges are converted to sharp edges by preserving all local maxima in the gradient image and deleting others. Eight directions are defined as edge directions as shown in Figure 14. The calculated edge direction is round to the nearest defined direction. Then, the edge strength of the current pixel is compared with the neighboring pixels in the positive and negative gradient directions. For example, if the gradient direction of the current pixel is north (θ = 90º), the edge strength of the current pixel is compared with the edge strength of the pixels at the north and south. If the edge strength of the current pixel is largest, then the edge strength is preserved, otherwise, the value of the edge strength is suppressed. The example application of non-maximum suppression is given in Figure 15. As shown in the figure, gradient directions are generally in north direction and the value of the edge strengths are compared with the pixels in the north and south and the pixels having the maximum value in this comparison are marked with white borders.

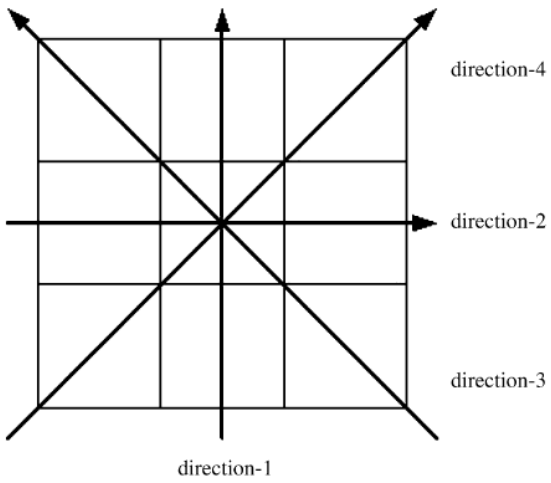


Figure 14: Four directions of edge [8]

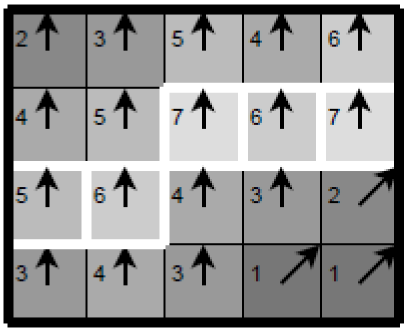


Figure 15: Illustration of non-maximum suppression [15]

Double thresholding is performed to the result of non-maximum suppression operation. Upper and lower threshold values are determined. The edges having higher values to the upper threshold value are marked as strong edges. The edges having strength values between upper and lower threshold values are marked as weak edge. The edges having strength values lower than lower threshold value are suppressed. The last step in the algorithm is edge tracking by hysteresis. Strong edges are interpreted as “certain edges” and directly included in the final edge image. Weak edges may come from true edges or noise/color variations. The probability of noise/color variants result in strong edges is low compared to the probability of real edges result in strong edges. Then, the ones in weak edges which result in strong edges are included in the final edge image. Edge tracking can be implemented by BLOB-analysis (Binary Large Object). The edge pixels are divided into connected BLOBs using 8-connected neighborhood. BLOBs containing at least one strong edge pixel are then preserved, while other BLOB’s are suppressed.

In the edge-based template matching algorithm used in this study, first, the input image and the template image are filtered by median filter. Then, the Canny edge detection is performed for both images. The example input image, the image after median filter is applied and the Canny edge detection resulting image are given in Figure 16, Figure 17 and Figure 18 respectively. Finally, the resulting image of the input image is filtered by the resulting image of the template image and the locations where the result of filtering is higher than a predetermined threshold are taken as matching points. Experimental result of the edge-based template matching method, when measured by the pixel-based and object based approaches are given in Table 5, Table 6, Table 7 and Table 8.

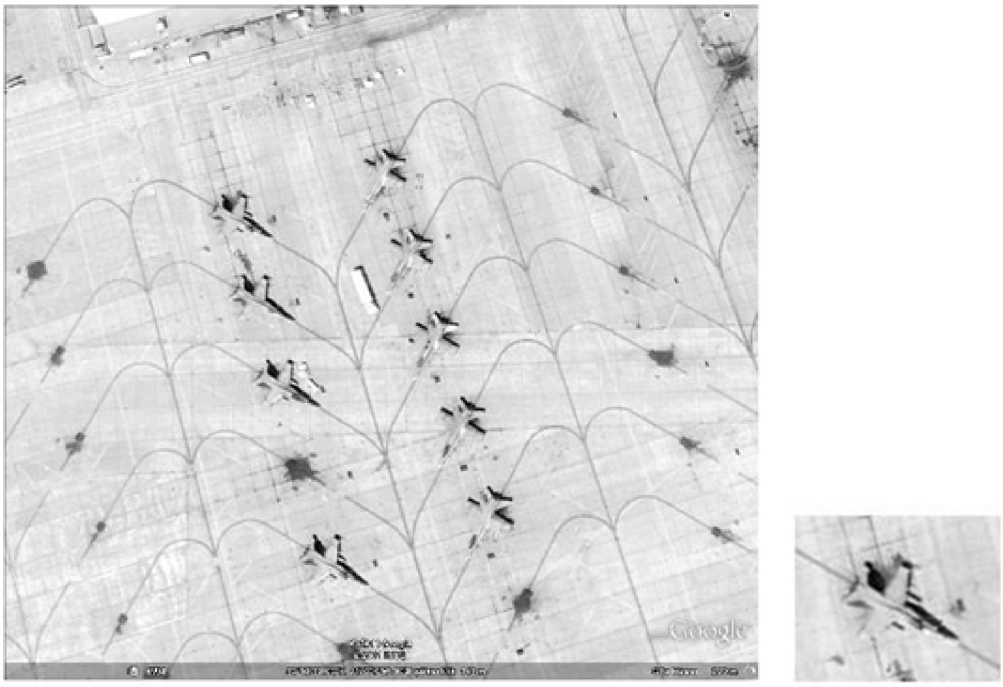


Figure 16: Example input image and template image before median filter applied

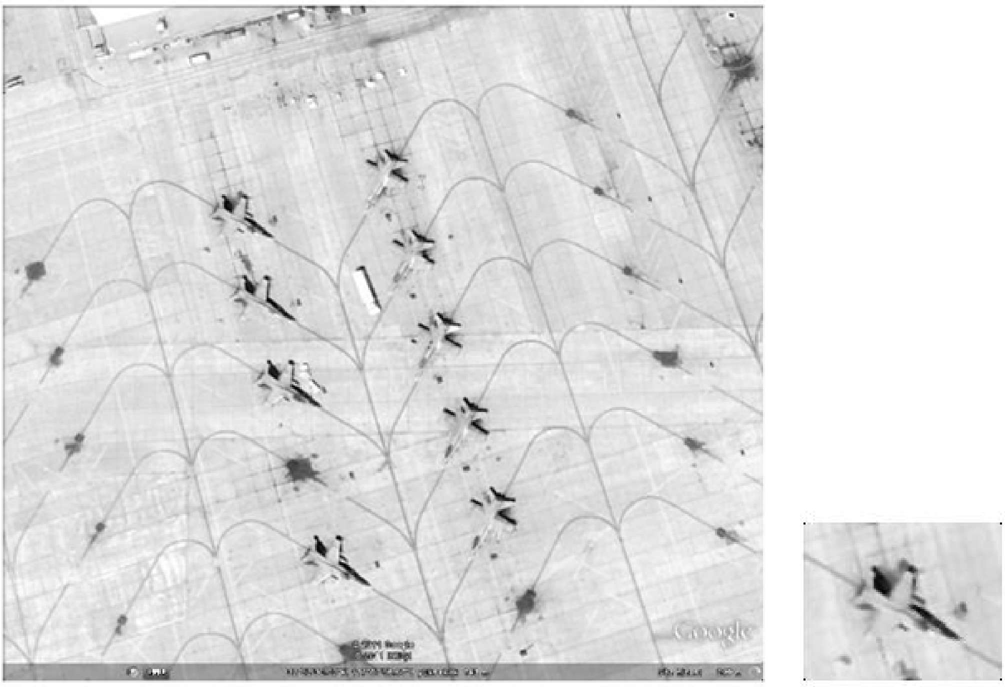


Figure 17: Example input image and template image after median filter applied

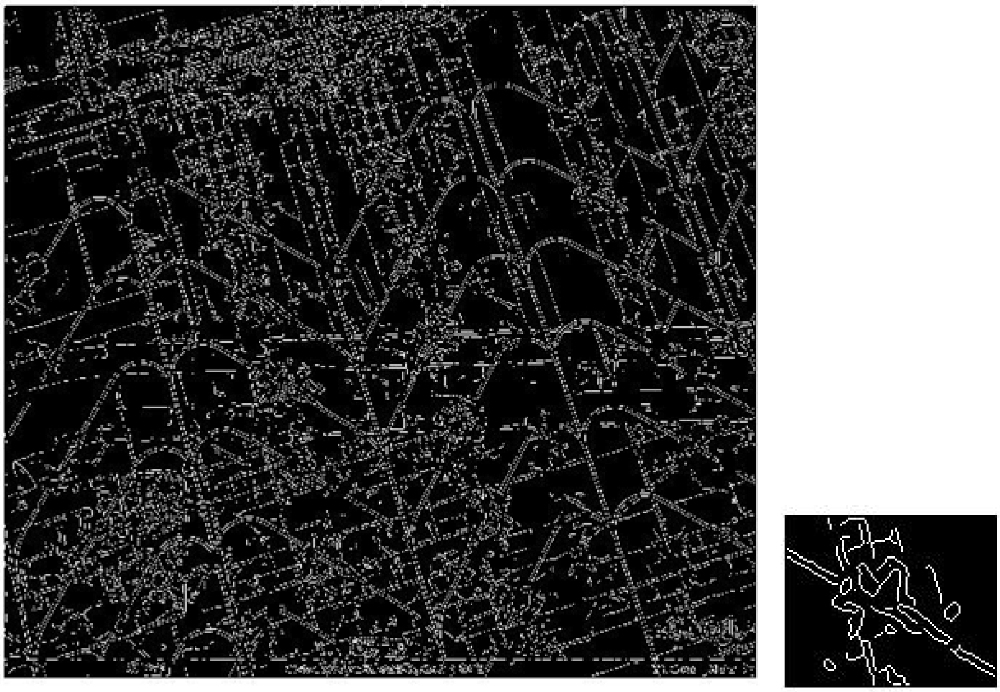


Figure 18: Example input image and template image after Canny edge detector applied

Table 5: Precision, Recall and F-Measure by the Pixel Based Approach

Image No	Precision	Recall	F-Measure
1	0.04	0.03	0,03
2	0	0	0
3	0.51	0.46	0,48
4	0	0	0
5	0	0	0
6	0	0	0
7	0.23	0.45	0,31
8	0.29	0.63	0,40
9	0.20	0.54	0,29
10	0.07	0.80	0,13
11	0.33	0.77	0,47
12	0	0	0
13	0	0	0
14	0	0	0
15	0.02	0.01	0,02
16	0	0	0
17	0	0	0
18	0.30	0.99	0,46

Table 6: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	FMeasure
Mean	0,11	0,26	0,14
Standard Deviation	0,16	0,35	0,19

Table 7: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.50	0.25	0,33
2	0	0	0
3	0.67	0.57	0,62
4	0	0	0
5	0	0	0
6	0	0	0
7	0.20	0.50	0,29
8	0.50	0.67	0,57
9	0.25	0.40	0,31
10	0.17	1	0,29
11	0.33	1	0,50
12	0	0	0
13	0	0	0
14	0	0	0
15	0.33	0.07	0,12
16	0	0	0
17	0	0	0
18	0.67	1	0,80

Table 8: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,20	0,30	0,21
Standard Deviation	0,24	0,39	0,26

3.2.3. Histogram based template matching

Histogram-Based template matching measures the similarity of the histograms of the template image and the input image. Both the input image and the template image are converted to gray scale. Let (u,v) is the pixel location in the input image and $t_r x t_c$ is the size of the template rectangle. Then, a rectangular window with the corner locations (u,v) and (u+t_r-1, v+t_c-1) is cropped from the input image. The histogram of both the template image and the cropped window are calculated. The intersection of the histograms are calculated by taking the minimum of the histogram values at each location. This operation guarantees to take the intersection of the histograms. An example demonstration of the histogram of the cropped window of the input image, template image and the histogram of the intersection is given in Figure 19. Then, all of the values in the intersection histogram are summed. This value gives the similarity of the cropped window and the template. This operation is performed for all locations in the input image by sliding the starting point (u,v). Then, for all pixels, a similarity value is assigned. The matching locations are found by comparing the similarity values with a threshold value. The experimental results of the histogram-based template matching method, when measured by the pixel and object based approaches, are given in Table 9, Table 10, Table 11 and Table 12.

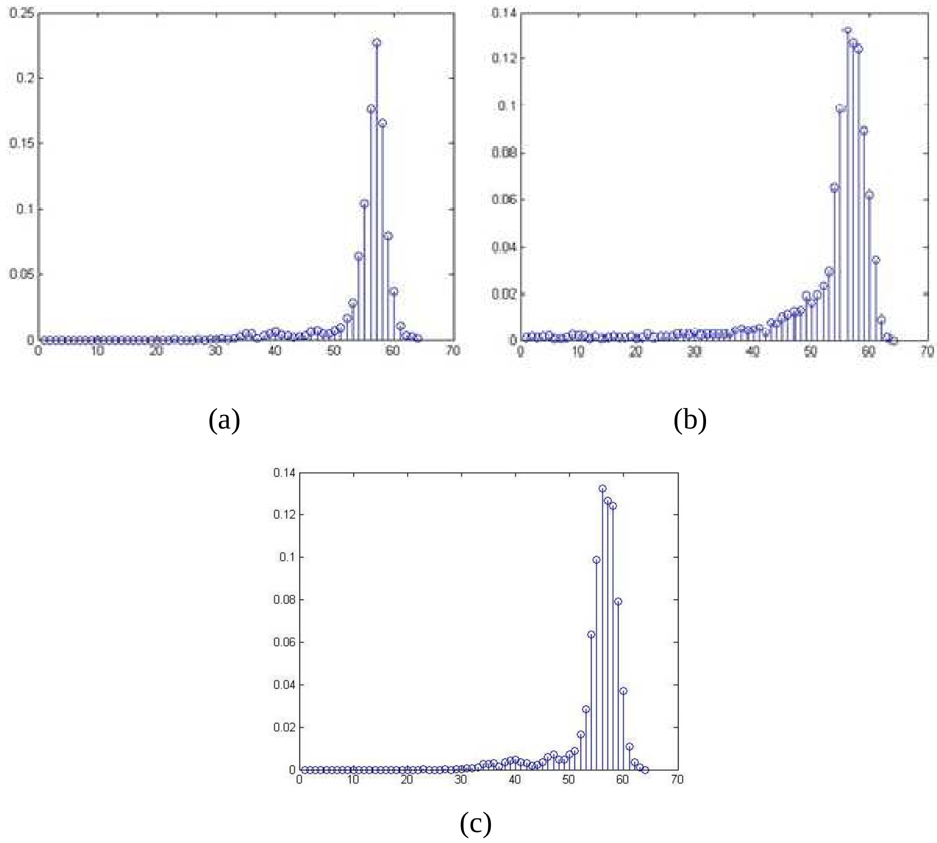


Figure 19: Example histogram of the cropped window of the input image (a), the template image (b) and, the intersection histogram (c)

Table 9: Precision, Recall and FMeasure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.09	0.59	0,15
2	0.21	0.70	0,32
3	0.39	0.72	0,50
4	0.31	0.47	0,37
5	0.46	0.49	0,47
6	0.22	0.77	0,34
7	0.11	0.61	0,19
8	0.12	0.89	0,21
9	0.08	0.54	0,14
10	0.02	1	0,04
11	0.12	0.90	0,21
12	0.04	0.28	0,07
13	0.17	0.83	0,29
14	0.04	0.87	0,08
15	0.14	0.63	0,22
16	0.27	0.59	0,37
17	0.20	0.68	0,31
18	0.14	0.97	0,24

Table 10: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	FMeasure
Mean	0,17	0,70	0,25
Standard Deviation	0,12	0,19	0,13

Table 11: Precision, Recall and FMeasure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.12	0.75	0,20
2	0.22	1	0,36
3	0.55	0.86	0,67
4	0.38	1	0,55
5	0.60	1	0,75
6	0.25	1	0,40
7	0.13	1	0,22
8	0.24	0.83	0,37
9	0.25	1	0,40
10	0.04	1	0,07
11	0.10	1	0,18
12	0.06	0.50	0,10
13	0.19	1	0,32
14	0.05	1	0,10
15	0.37	0.93	0,53
16	0.72	0.93	0,81
17	0.31	1	0,47
18	0.17	1	0,29

Table 12: Mean and Standard Deviation of Precision, Recall and FMeasure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,26	0,93	0,38
Standard Deviation	0,20	0,13	0,22

3.2.4. Angular Radial Transform Template Matching Method

Shape features of 2-dimensional and 3-dimensional objects are very important for image processing and image recognition applications. Object functionality and identity can be retrieved from the shape of the objects and object shape vectors are very powerful descriptors for similarity measure. This property distinguishes shape from some other elementary visual features such as color or texture. The Visual Part of the MPEG-7 standard defines three descriptors for shape features with different properties. These are the contour-based shape, the region-based shape and the 3D Shape spectrum descriptors. Two approaches were developed by MPEG to cover the 2D shape descriptor issue. These are contour-based shape tools and region-based shape tools. The region-based shape descriptor expresses pixel distribution within a 2D object region. The properties of multiple disjoint regions are described in a compact and efficient way in region-based descriptors. In contour-based shape descriptor, Curvature Scale Space representation of the contour is concerned. The contour based shape descriptors are very robust to non-rigid deformations such as outline of a running person. The powerful and weak sides of the region based shape descriptor and the contour based shape descriptor are illustrated in Figure 20. Although the objects in the first row are different from each other clearly, they have similar spatial distribution of pixels and are therefore similar according to the region based shape descriptor. On the other hand, contour based shape descriptor provides differential information among them and recognizes them as different. When contour-based similarity is concerned in the second column, objects in that column are similar according to the feature vectors produced by the contour-based shape descriptor. But, the spatial distribution of pixels in that column is very different and region based shape descriptor recognizes that the objects are different from each other.

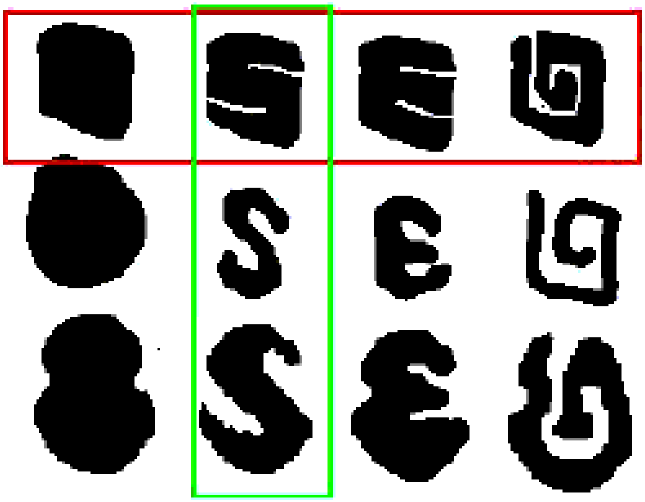


Figure 20: Example of region-based and contour based region similarity [16]

The region-based shape techniques are more powerful for complex shapes that consist of several disjoint regions such as trademarks or logos etc. Since the images used in this work are satellite/plane images, region based techniques are more suitable for these samples. Three region-based shape descriptors are proposed: Multi-Layer Eigen Vector Descriptor (MLEV), a descriptor based on Zernike moments and a descriptor based on Angular Radial Transform (ART). According to the experiments, it was concluded that ART descriptor offers the best overall performance for region-based similarity [16]. ART is the orthogonal unitary transform defined on a unit disk that consists of the complete orthonormal sinusoidal basis functions in polar coordinates. The ART coefficients are defined as in *(3.12)*.

$$F_{nm} = \langle V_{nm}(\rho,\theta), f(\rho,\theta)\rangle = \int_0^{2\pi}\int_0^1 V_{nm}^*(\rho,\theta), f(\rho,\theta)\rho d\rho d\theta \quad \textbf{\textit{(3.12)}}$$

Where F_{nm} is an ART coefficient of order n and m, $f(\rho,\theta)$ is an image function in polar coordinates and $V_{nn}(\rho,\theta)$ is the ART basis function that are separable along the angular and radial directions as in *(3.13)*.

$$V_{nm}(r,q) = A_m(q)R_n(r) \quad \textbf{\textit{(3.13)}}$$

In order to achieve rotation invariance, an exponential function is used for the angular basis function *(3.14)*

$$A_m(q) = \frac{1}{2p}\exp(jmq) \quad \textbf{\textit{(3.14)}}$$

The radial basis function is defined by a cosine function,

$$R_n(r) = \begin{matrix} 1 & n=0 \\ 2\cos(\rho nr) & n\neq 0 \end{matrix} \quad \textbf{\textit{(3.15)}}$$

Rotation invariance means that the magnitudes of the ART coefficient of the given image and the rotated version of the same image are the same *(3.16)*. Let the original image function be $f(r,q)$ and the rotated image function be $f^a(r,q)$ where the relation of the image functions are given in *(3.16)*.

$$f^a(r,q) = f(r,a+q) \quad \textbf{\textit{(3.16)}}$$

Then the ART of the rotated image are given as

$$F_{nm}^a = \frac{1}{2\rho}\int_0^{2\rho}\int_0^1 V_{nm}^*(r,q)f^a(r,q)rdrdq \quad \textbf{\textit{(3.17)}}$$

or

$$F_{nm}^a = F_{nm}\exp(-jma) \quad \textbf{\textit{(3.18)}}$$

The exponential term does not affect the magnitude. Then, it is proved that the magnitudes of the ART of the original image and the rotated image are same.

$$\|F_{nm}^{a}\| = \|F_{nm}\| \qquad \textbf{\textit{(3.19)}}$$

When measuring the similarity of two shapes, the ART coefficients for each order are calculated for both shapes. Then, the sum of the absolute differences of each order of descriptor elements is calculated. The result gives the distance between two shapes. If the distance is high, the similarity of the shapes is low.

Template matching is performed according to the similarity measures of the regions and the experimental results, when pixel and object based approaches are used, are given in Table 13, Table 14, Table 15 and Table 16.

Table 13: Precision, Recall and FMeasure with the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.07	0.77	0,12
2	0.10	0.89	0,18
3	0.21	0.60	0,31
4	0.12	0.18	0,15
5	0.22	0.68	0,33
6	0.06	0.70	0,12
7	0.05	0.67	0,09
8	0.02	0.93	0,04
9	0.03	0.83	0,05
10	0.02	0.80	0,03
11	0.04	0.74	0,07
12	0.02	0.90	0,05
13	0.06	0.83	0,11

Table 13: Precision, Recall and FMeasure with the Pixel Based Approaches (continued)

14	0	0	0
15	0.14	0.56	0,22
16	0.16	0.09	0,11
17	0.04	0.55	0,07
18	0.35	0.66	0,46

Table 14: Mean and Standard Deviation of Precision, Recall and FMeasure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,09	0,63	0,14
Standard Deviation	0,09	0,27	0,12

Table 15: Precision, Recall and FMeasure with the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.11	1	0,20
2	0.13	1	0,24
3	0.38	0.71	0,50
4	0.25	0.67	0,36
5	0.33	1	0,50
6	0.09	1	0,17
7	0.10	1	0,17
8	0.18	0.83	0,29
9	0.15	1	0,26
10	0.05	1	0,10
11	0.07	1	0,13
12	0.08	1	0,15
13	0.11	1	0,20
14	0	0	0
15	0.48	0.79	0,59
16	0.40	0.14	0,21
17	0.14	0.75	0,24
18	0.50	1	0,67

Table 16: Mean and Standard Deviation of Precision, Recall and FMeasure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,20	0,83	0,28
Standard Deviation	0,15	0,30	0,18

CHAPTER 4

CLASSIFIER FUSION

4.1. INTRODUCTION

Determining the true class of a given pattern by using a single feature descriptor and a particular classification procedure is the traditional approach in pattern recognition systems [17]. According to this approach, the success improvements and progress in pattern recognition and decision support systems are based on the continuous development of existing methods as well as discovering new ones [18]. But, in case of noisy inputs and large number of classes, obtaining improved performances by using a single classification procedure is difficult to achieve. This situation has led to the development of another approach, which claims that combining individual methods provides better results [18]. The multiple expert fusion received considerable attention [19]. In combination process, first, each pattern recognition problem is solved individually and then results are combined in some way to achieve reduced recognition error rates.

There are many methods developed for classifier fusion. Classifier fusion methods may be divided into two general categories. The fusion methods in the first category do not do anything with classifier outputs until they find single best classifier or a selected group of classifiers. If single best classifier is found, then its output is taken as the final output. If a group of classifiers is selected, then only their outputs are considered for the final decision or for further processing. This approach is used in dynamic and static classifier selection methods, classifier structuring and grouping and hierarchical mixture of experts methods which will be analyzed later. In the other category, methods operate on classifier outputs and use different methodologies for combining the outputs. These methods may be further classified according to the output the classifiers produce for combinations. Classifier outputs can be class labels, class rankings and soft or fuzzy classifier outputs. If only labels are available, voting methods and knowledge space obtained from the training set may be used. Sometimes, classifiers produce rankings for each label. In this case, class set reduction or class set reordering approaches may be utilized by widely used methods such as the highest rank method, Borda Count, logistic regression, intersection of neighborhoods and union of neighborhoods. If continuous outputs like posterior probabilities are supplied, then linear combination methods like average Bayes combination, or nonlinear methods like product of experts may be used [14]. Other most popular methods for soft/fuzzy outputs are Dempster-Shafer Combination, Fuzzy Templates, Fuzzy Integrals and Neural Network. Diagrammatic representation of the proposed taxonomy of classifier fusion methods is shown in Figure 21 [18].

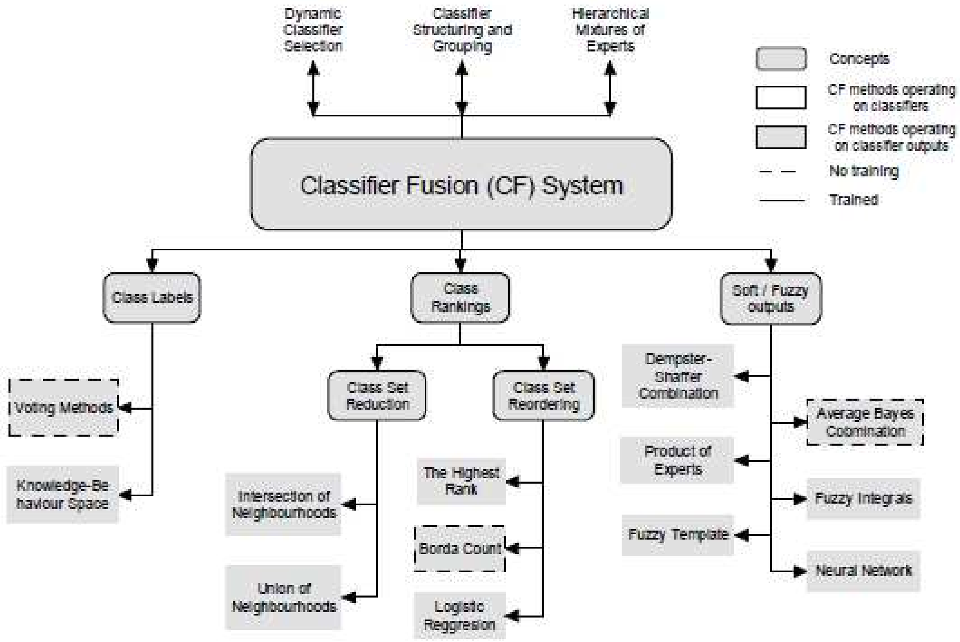


Figure 21: Taxonomy of classifier fusion methods **[18]**

The main motivation behind the classifier fusion is to obtain a system in which different classifier designs potentially offer complementary information about the patterns to be classified and improve the classification performance. In order to obtain complementary information from different classifiers, classifiers should be different from each other. In other words, classifiers should be diverse as much as possible. Otherwise, the overall decision will not be better than the individual decisions. Although, there is no consensus on the meaning of notions such as diversity, complementarity, orthogonality etc, many methods were proposed for measuring classifier diversity. There is no unique choice of a measure of diversity. As stated in Kuncheva and Whitaker's work [20], two general approaches are commonly used in diversity measure. These are pairwise and non-pairwise measures. In pairwise measuring methods, diversity or similarity is measured for each pair of classifiers. Consider that D_i and D_j are two classifiers. A 2x2 table which is shown in Table 17 summarizes the output of these classifiers. N^{11} denotes the number of cases both classifiers give the correct outputs, N^{00} denotes the number of cases both classifiers give the wrong outputs. N^{10} denotes the number of cases D_k gives wrong output and D_i gives correct output and, N^{01} denotes the number of cases D_k gives correct output and D_i gives wrong output. According to these numbers, different diversity measure methods were developed for pairwise diversity measure. One of them is Q statistics. The formula for Q statistics is given in *(4.1)*. If classifiers have a tendency to classify the same objects correctly and same objects incorrectly, Q statistics will have positive values. It means that diversity between classifiers is low. If classifiers commit errors on different objects, then Q statistics will have negative values. It means that classifier diversity is high.

Table 17: Illustration of Q statistics

	D_k correct	D_k wrong
D_i correct	N^{11}	N^{10}
D_i wrong	N^{01}	N^{00}

$$Q_{i,k} = \frac{N^{11} * N^{00} - N^{10} * N^{01}}{N^{11} * N^{00} + * N^{01}} \qquad \textbf{\textit{(4.1)}}$$

Other popular pairwise diversity measure methods are the correlation coefficient method, disagreement measure, and double fault measure [20]. In non-pairwise measures, diversity is measured on whole group of classifiers. The main advantage of non-pairwise diversity measure is to prevent losing some information about error relations when there are more than two classifiers [21]. The entropy measure is one of the important non-pairwise diversity measure methods. For a system of M parallel classifiers such that D = $\{D_1, \dots, D_M\}$ with producing binary outputs for N input samples x_i, (i=1, …, N). Each classifier produces an output $y_{i,j}$ j =1, …, M for input sample x_i. The value of $y_{i,j}$ is 1 if classifier j produces correct output for input sample x_i, and $y_{i,j}$ is 0 if classifier j produces incorrect output for input sample x_i. Let m(x_i) denote the number of classifiers producing error for the input sample x_i which is given in *(4.2)*. Then the entropy measure is as in *(4.3)*.

$$m(x_i) = M - \sum_{j=1}^{M} y_{i,j} \qquad \textbf{\textit{(4.2)}}$$

$$H = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M - \lceil M/2 \rceil} min\{m(x_i), M - m(x_i)\} \qquad \textbf{\textit{(4.3)}}$$

Other well-known non-pairwise diversity measure methods are the measure of difficulty, Kohavi-Wolpert variance, measurement of interrater agreement, generalized diversity and coincident failure diversity [20]. As mentioned before, diversity of classifiers highly affects the performance of fusion results. There are many studies supporting this idea. Ruta and

Gabrys [22] performed an experiment for analyzing the correlation of majority voting method and diversity measures. According to their research, error rates in majority voting method were highly correlated with the diversity values calculated by both pairwise and non-pairwise diversity measures.

4.2. RELATED WORK

4.2.1. Voting Methods

Although classifiers producing crisp, single class labels provide the least amount of useful information for the fusion process, they are still applied to a variety of real life problems. Voting methods and behavior knowledge space methods are two important approaches for fusion of single label output classifiers. Voting method may be applied to classifier outputs directly or after a training process. Majority voting method is the most popular method for voting without training. Majority voting gains its popularity from simplicity and performance on real data. The label outputs of classifiers are taken and the class which receives the maximum vote is taken as the final output. There are some studies for determining the lower and upper bounds of the performance of majority voting [23, 24]. Kuncheva [25] attempted to measure and analyze the performance of majority voting empirically.

Narasimhamurty [26] tried to explain the limits of majority voting theoretically and stated that majority voting method increases correct decision rate if the classifiers used are independent and the individual error rate of each classifier is below the 0.5. If the independency condition is satisfied, then the accuracy of majority voting method becomes higher with the increasing number of classifiers. But, enforcing statistical independence in a classifier ensemble is a very hard condition to be satisfied. In order to obtain a classifier set which is composed of independent classifiers, diversity measure mentioned before may be used. It is reasonably expected that when the diversity value is high, then the independency increases and the performance of majority voting becomes higher. Narasimhamurty [26] made some experiments to measure the correlation of diversity and performance of majority voting method. Two experiments were performed. In one of them pairwise diversity measure approach was used. Q statistics was calculated for every classifier couples in the classifier set and the average of Q statistics values was taken as diversity value. In the other experiment entropy measure was performed. According to experimental results, there was no correlation between the performance of majority voting and diversity value calculated by Q statistics. But, the success rate of majority voting method was increasing with the higher values of diversity calculated by entropy measure. This results point out that theoretically independency of classifiers directly affects the performance of majority voting method, but there is no widely accepted formal characterization of diversity. Hence, it is very hard to characterize the relationship between majority voting accuracy and classifier diversity.

Voting methods that needs training is very successful in improving the accuracy of certain classifiers for artificial and real world datasets [27]. Some of the methods adaptively change the distribution of training set based on the performance of previous classifiers and some of them do not. Boosting can be given as an example to the algorithms which change the distribution of training set. Boosting improves the performance of a weak learning algorithm. AdaBoost algorithm is an algorithm in boosting algorithm family developed by Freund [28], training sets are generated sequentially and classifiers are built for the generated training sets. Finally, weighted voting is performed for combining the generated classifiers. The weight of each classifier depends on the performance on the training set used to build it. In Bootstrap aggregating (bagging) algorithm, the classifiers are built in parallel for each bootstrap sample which is generated from the training set with replacement. Bauer and Kohavi [27] made some experiments for empirical comparison of two families of voting algorithms: perturb and combine (e.g. Bagging) and boosting (e.g. AdaBoost). The voting techniques in these two families were extremely successful at reducing the loss according to a mean squared error evaluation and in general, boosting algorithms were better than Bagging.

4.2.2. Knowledge Space Methods

Behavior-Knowledge Space method, which was introduced by Huang and Suen [29] is another method works on single class label output classifiers. Behavior-Knowledge Space method was developed to obtain better results by aggregating the decisions of individual classifiers. The method contains two stages. In the first stage, a knowledge space is constructed from the behavior of classifiers in the training set. The second stage is the operation stage. For each test sample, final decision is made according to the decisions generated from individual classifiers and the corresponding information in the knowledge space. In Behavior-Knowledge Space method, every possible combination of individual classifiers is regarded as a cell in the knowledge space. For every cell, the number of samples taken from training set for each class is recorded and the most representative class label is determined for that combination of individual classifiers.

In statistical point of view, Behavior-Knowledge Space method tries to estimate the distribution of each class from the frequencies of occurrence in the training set. The most important advantage of Behavior-Knowledge Space method is that it does not require any independency relation between the classifiers. The main drawback of that method is its exhaustive approach. It lists all combination of classifiers' outputs which means large space complexity. If large data sets are used, knowledge space can be modeled accurately but memory usage will be too high. If small sample size is used, the method will be extremely overfitting to the training set and generalization ability will be very poor. In order to solve this problem, Roli [30] proposed to inject noise to the training set. Yang and Zang [25] improved this approach by adding observational learning algorithm. In the proposed approach, observational learning algorithm is performed based on the classifier ensemble and enlarged dataset is generated. Then, final decision is made through the Behavior-Knowledge Space constructed by enlarged dataset. In the experiments of Yang and Zang, three learners which are linear discriminant classifiers, quadratic discriminant classifier and K-nearest neighbor classifier were adopted. According to test results, Behavior-Knowledge Space method with observational learning algorithm outperforms to the classic Behavior

Knowledge Space method on three datasets of the five ones. Behavior-Knowledge Space with observational learning algorithm did not provide high improvement when considering the additional computational complexity of learning algorithm. But, this study is very important for bringing a solution approach to small sample size problem in Behavior-Knowledge Space method.

4.2.3. Fusion Methods for Ranking Outputs

Classifiers may give the class rankings as output. This type of classifiers provide more information in their output compared to single class output classifiers. There are two approaches for fusing ranking output classifiers. One of them is class set reduction approach. The objective of class set reduction approach is to minimize the number of classes in the output list by ensuring the true class is included in the class set [12,8]. Intersection of neighborhoods and union of neighborhoods are two simple and direct methods developed for this purpose. In intersection of neighborhoods method, the lowest ranking value of each classifier given to the true class is determined. In other words, the ranking value for the worst case is determined. The worst case ranking value is taken as a threshold for that classifier. When a test sample is given, the classes above the predetermined threshold ranking value are put in the list. Then, the intersection of all classifiers' list is taken as the final class set for the given test sample. The union of neighborhoods method uses a max-min procedure. The best rank (minimum) in classifier outputs for each input sample is determined. Then, the maximum (worst) of best ranks for each classifier is obtained and taken as threshold for that classifier. When a test sample is given, each classifier puts the classes with higher rank from the threshold to the list. Finally, a class set is composed by taking the union of lists' of each classifier. The intersection of neighborhoods method process is illustrated in Figure 22 for the example classifier outputs table with six input types and four different classifiers. Studies showed that [17], the intersection approach provides small neighbor sizes missing the true class when all the classifiers have poor performance. If one or more classifiers are specialized for some kind of input samples, intersection of neighborhoods does not give reliable results and union approach is preferred in that case. Union approach focuses on the best-case behavior of each classifier.

	$rank_j^i$ of classifier C_j			
Input I_i	C_1	C_2	C_3	C_4
I_1	3	12	1	24
I_2	1	5	29	12
I_3	34	3	4	6
I_4	9	7	6	7
I_5	4	36	5	5
I_6	16	2	3	4

best rank for each input

$rowmin_j$			
C_1	C_2	C_3	C_4
0	0	1	0
1	0	0	0
0	3	0	0
0	0	6	0
4	0	0	0
0	2	0	0

worst rank for each classifier

C_1	C_2	C_3	C_4
4	3	6	0

Figure 22: Example illustration of threshold finding in intersection of neighborhoods

The other approach for fusing ranking output classifiers is class set reordering approach. The aim of class set reordering methods is to improve the rank of the correct class. Method is considered successful or not by looking how far the ranking of the true class is away from the top ranking. The highest ranking method, the Borda count method and logistics regression are examples of class set reordering methods. In highest rank method, the ranking of each classifier is considered and for each class the highest ranking given in classifiers is assigned to that class as the ranking fusion output. Then, classes are sorted according to the assigned rankings. In this assignment, ties are possible between the classes and may be broken by giving priority to some classes from the priori information. With large number of classes and few classifiers, the highest rank method is particularly useful. As long as one powerful classifier exist for the given input pattern, no matter how the other classifier perform, the highest rank method assigns high ranking value to correct class. The main disadvantage of this method is that there may be too many ties after fusion process. Working with large number of classifiers probably causes this problem.

The Borda count method is generalization of majority voting method to ranking output classifiers. It is simple to implement and requires no training. For each class, the number of classes ranked below that class is calculated in each individual classifier. Then, these numbers are summed. Ranking of classes assigned according to result of that sum value. In Borda count method, all classifiers are treated equally and they are assumed independent. The differences in individual classifier capabilities are not considered [17]. Logistic regression method is developed to overcome the problem in Borda count method treating to classifiers equally even quality of individual classifier outputs differ from each other. Weights are assigned to each classifier in order to reflect their importance in multiple decision system. Then, so-called logistic regression is performed to obtain combined ranking values. Assume that fusion system is composed of m classifiers and $(x_1, x_2, \ldots, x_m)$ are the responses from m classifiers for each classes. Then the logistic response function is as follows [18]:

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m)} \qquad \textbf{\textit{(4.4)}}$$

The logit function is as follows:

$$L(x) = log \frac{\pi(x)}{1 - \pi(x)} = (\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m) \qquad \textbf{\textit{(4.5)}}$$

According to the result of logit functions, the combined rankings are created. The model parameters are estimated in training stage by using data fitting methods based on maximum likelihood. Monwar and Gavrilova [31] recently made some experiments for analyzing and comparing the performances of Borda Count and Logistic Regression methods in biometric recognitions test. In this tests, both Borda Count and Logistic Regression improved the system accuracy and the error rates in Logistic Regression method was smaller comparing to Borda Count method.

4.2.4. Fusion Methods for Real Valued Output Classifiers

The classifiers producing outputs as real values in the range [0,1] are considered as soft output classifiers. These measures are generally referred to as fuzzy measures and give evidence about different dimensions of uncertainty by covering the evidence of probability, possibility, necessity, belief and plausibility. The fusion methods for soft/fuzzy output classifiers aim to reduce the uncertainty by maximizing the suitable measures of evidence. If the outputs of classifiers are expressed in posterior probabilities, the Bayesian methods can be used for classifier fusion. Bayes average and Bayes Belief Integration are two mostly used methods in soft output classifier fusion approach. Bayes average method is a simple method and requires no training. The mean of posterior probabilities is taken from the classifier outputs and classification is performed according to the calculated mean value. If classifiers do not produce posterior probabilities as outputs, the posterior probability may be estimated by k-NN method or some other methods by training. The transformation of a single label output classifier to posterior probability by k-NN is as follows:

$$P_k\left(x \in \frac{C_i}{x}\right) = \frac{k_i}{k_{nn}} \qquad \textit{\textbf{(4.6)}}$$

Where k_i denotes the number of prototype samples from class C_i out of all k_{nn} nearest prototype samples. The performance of Bayes average fusion method depends on the diversity of classifiers and the accuracy of classifiers in posterior probability estimation. In Bayes belif integration method, the errors are represented in a matrix called confusion matrix where how many samples coming from which class and assigned to which class is expressed for each classifier. Belief values for each classifier are calculated in the training stage and combined. The class with the highest combined belief measure for the given input sample is selected.

Some studies are performed in order to analyze and compare the performances of Bayes average and Bayes belief integration. Recently, Demirkesen and Cherifi [32] published a study about the performances of feature level fusion and classifier fusion on natural scene images by using support vector machines. They used four classifiers which were developed by performing support vector machines on four different feature descriptors. The used descriptors were color layout descriptor, edge orientation histogram, texture representation and gist feature. Texture representation was obtained by extracting four attributes namely energy, entropy, homogeneity and inertia from gray level occurrence matrix. Gist feature was a 476-dimensional vector which is a representation of the scene structure based on the output of filters tuned to different orientations and scales. Eight categories of natural scenes which were highways, streets, forests, open country, inside of cities, tall buildings, coasts and mountain images were used in the experiment. The similar images were coupled and as a result there were four classifiers and four classes. As classifier fusion methods majority voting, Bayes average and Bayes belief integration methods were used. The classifiers were producing single class output. These outputs were transformed to fuzzy outputs for Bayes average and Bayes belief integration methods. According to their test results, Bayes belief integration produces the best results compared to the other two classifier fusion methods. Bayes average outperformed majority voting, but the results of Bayes average were not higher than Bayes belief integration for all four classes. In Dempster-Shafer method, the set of all possible classes is considered as a set of mutually exclusive and exhaustive propositions. All subsets of this set are included in power set and each element of power set is called focal element. A belief value is assigned to each focal element based on the evidence and the belief values of different classifiers are combined [23]. In fuzzy templates method, decision templates are constructed for each class. The outputs of the classifiers for each sample form so-called decision profile holding the values of support of each classifier to each class. Decision template for a class is the average of decision profiles for the samples of that class. Then, when a test sample is given, the support of classifiers for each class is calculated and a decision profile is constructed. The similarity of that decision profile for each decision template is calculated and the class of most similar decision template is assigned. Many studies conducted so far to analyze the performance of fuzzy templates method [16, 2]. In most of the studies the accuracy of fuzzy templates was higher than the single best classifier [33].

Although there were some studies where fuzzy templates method were superior to majority voting, behavior knowledge space, Dempster Shafer and Bayesian methods [34], the performance of fuzzy templates are highly dependent on classifiers and the dataset. Hence, it is hard to say one winning classifier fusion method.

Other fusion methods given in the taxonomy of classifier fusion methods for soft/fuzzy output classifiers are product of experts and artificial neural networks. Product of experts method is beneficial for high dimensional problems like face recognition. Artificial neural network method is an iterative learning algorithm which makes an input output mapping. In classifier fusion, the input of artificial neural network is the outputs of classifiers. The number of outputs in artificial neural network can be equal to the number of classes which denote the support for each class. If a crisp decision is required the output with the highest value is chosen [18].

4.2.5. Classifier Selection Approach

Until now, the combination of classifier output was discussed as common operation mechanism of multiple classifier systems. Some researchers pointed out the potentialities of classifier selection as an alternative operation mechanism [6, 33]. Selection-based methods may choose the best classifier by simply comparing the performances of classifiers in the training set, or they may adaptively provide the best classifier to the given input type. The first method can be called as static classifier selection in which the type of input is unimportant. The only knowledge obtained from the training stage is the classifier which gives the best performance in the classifier ensemble. The second method considers the input type and tries to obtain the best classifier for that input. This approach is called as dynamic classifier selection or adaptive classifier selection. In some systems, especially in neural networks, the modular approaches are used in dynamic classifier selection. System is composed of specialized networks where each one is responsible for some aspect of classification task. All the nets are necessary to solve the whole task. But, modular approach is not adopted in pattern recognition field. In pattern recognition, dynamic classifier is used to develop a decision support system where each classifier is able to solve the whole classification task and the complementary behavior of each classifier is used for different input types. In order to describe the classifier as discriminative as possible, a well defined feature vector should be provided. By the help of the defined feature vector, different types of inputs can be discriminated and the performance of each classifier for different types of inputs can be analyzed. This helps to choose the most appropriate classifier for any input after determining the input type.

Srihari et al were the first to introduce the concept of dynamic classifier selection method [17]. As Srihari stated that the aim of dynamic classifier selection method is to take each classifier's best output. In order to achieve this, measurable characteristics of patterns should be detected and the correlation of classifier performance with the pattern characteristics should be defined as clear as possible. Dividing the training set into partitions by a set of mutually exclusive conditions is a way to obtain the oracle of classifiers. After training the system for each partition, the best classifier for each partition is determined and then the test sample will be classified by the corresponding classifier according to the partition of the sample. Srihari suggested partitioning the input sample by measuring the disagreement of classifiers for that input sample. Training set is partitioned according to the disagreement of classifier outputs and the best classifier for the given disagreement value is determined. The major drawback of this method is that the classifiers should be independent in order to provide discriminative partitioning.

Woods et. al. developed a dynamic classifier selection approach based on local accuracy estimates [35]. Local regions of the feature space of a test sample are defined in terms of the k-nearest neighbors in the training data. The local accuracy of each classifier for the feature space surrounding a test sample is calculated by two methods. One of them is the overall local accuracy calculation. The percentage of correctly classified k-nearest neighbor samples in the training data. The other method also considers the assigned class by each classifier and takes the percentage of correctly classified samples assigned to that class. The experiments showed that the second method is superior to the overall local accuracy method. Woods et. al. also compared the performance of local accuracy dynamic classifier selection method with other combination methods and claimed that local accuracy dynamic classifier selection method outperformed the other methods and average accuracy value was near to the average accuracy value of the oracle. In some studies, overall accuracy approach is called as a priori selection method [36]. A priori name comes from the fact that the class assigned by the classifier to the test pattern is not known. The second approach is called as a posteriori selection. Since, in probability calculation, the information of the class assigned by the classifier is considered. The experimental and empirical results supported the potentialities of dynamic classifier selection method.

There is also a study published by Giacinto and Roli provided a theoretical framework for dynamic classifier selection method. It was shown that optimal Bayes classifier can be obtained by the selection of non-optimal classifiers [37]. It is thought that each classifier divides the feature space into decision regions by discriminant functions and it is assumed that in some decision regions non optimal decisions of classifiers intersect with the optimal Bayes decision regions. The theory behind the framework comes from the assumption of a reasonable degree of complementarity among the optimal decisions of classifiers exists and optimal decision boundaries are piecewise coincident with the boundaries of the classifiers. Assume that there are M possible classes and K classifiers. For each classifier there are M discriminant functions $d_i^j(X)$, i=1,…,M, j = 1,…,K and X is the feature vector. If X feature vector is assigned to class I by the classifier j, then $d_i^j(X) > 0$, otherwise $d_i^j(X) < 0$ and in decision boundary $d_i^j(X) = 0$. R_i^j denotes the decision region assigned to class I by the classifier j. The intersection of decision regions of each classifier with the optimal Bayes decision regions is shown in *(4.7)*.

$$R_{i+}^j = R_i^j \cap R_i^B \tag{4.7}$$

The representation of complementarity of decision regions and complementarity of decision boundaries are shown in *(4.8)* and *(4.9)* respectively.

$$R_i^B = \bigcup_{j=1}^K R_{i+}^j \qquad i = 1, \dots, M \tag{4.8}$$

$$\forall X: d_i^B(X) = 0 \Longrightarrow \sum_{j=1}^K \alpha_j(X) d_i^j(X) = 0, where\ \alpha_j(X) = \{0,1\}\ and\ \sum_{j=1}^K \alpha_j(X) = 1 \tag{4.9}$$

An example for the two-dimensional classification task with three data classes and the optimal Bayes decision regions are showed in Figure 23. In Figure 24, partitioning generated by the two classifiers and the relations between the optimal and non-optimal regions of the classifiers are given.

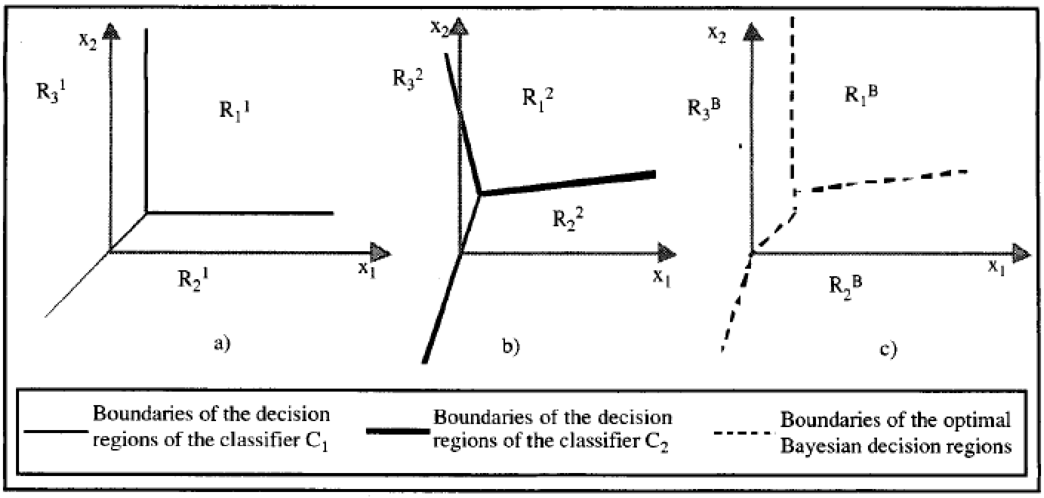


Figure 23: Example decision regions for two classifiers with three data classes in two dimensional feature space(a), (b) and the optimal Bayes decision regions (c) [37]

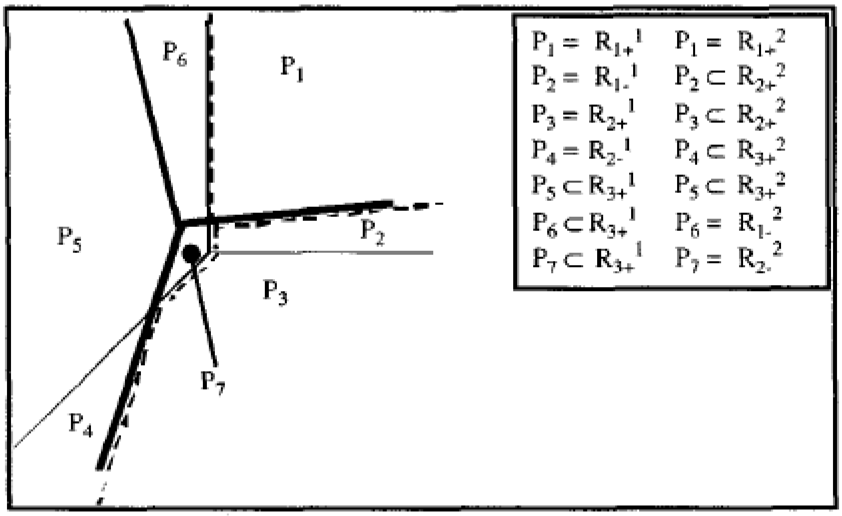


Figure 24: Partitioning generated by classifiers in the two-dimensional feature space and the relations of partition regions with the optimal and non-optimal regions of classifiers [37]

4.2.6. Hybrid Methods

In order to provide the most suitable output for the input pattern, hybrid methods were proposed in which the classifier selection and classifier combination methods are combined. Canuto et. al. developed two hybrid methods which are clustering and selection method and k-NN and selection method [38]. In clustering and selection method, the patterns are clustered by the k-means method and the test pattern is assigned to the nearest cluster. Then, a set of classifiers are formed by the most accurate classifiers for this cluster. This set is narrowed by choosing only the most diverse ones. Finally, fusion based method is applied for the remaining classifiers. In k-NN and selection method, instead of clustering the samples in the validation set, nearest neighbors of the test sample is taken to form a set of patterns. Then, the most accurate classifiers are chosen for this set of patterns. After narrowing the classifier set by only taking the most diverse ones, fusion-based method is performed. Clustering and selection method and k-NN and selection method only differ in providing the most accurate classifier set.

4.3 OUR APPROACH

In this study, the fusion of template matching methods is studied. Four template matching methods, which are explained in Section 3 with their results for our database, are used. Hence, our decision system is composed of four classifiers and two classes. All classifier fusion methods mentioned above are not appropriate for our decision system. Some of them are meaningless for our system such as class set reduction. Because, the number of possible classes is two in template matching. Two types of outputs can be obtained from the classifiers, binary outputs and soft outputs. Then, application of the classifiers working on ranking output classifiers is not possible.

The methods covered in this study are some literature methods such as majority voting, behavior-knowledge space, Bayes average, Bayes belief integration, fuzzy templates, two classifier selection methods, which are all explained in Section 4.2, and a newly proposed optimization based fusion method, which is explained in Section 4.3.2.4. These methods are implemented and the experimental results for our data set are given in the following subsections respectively.

4.3.1. Classifier fusion methods for single label output classifiers

4.3.1.1. Majority voting

Voting methods are applied to classifiers in which each classifier gives a single class label as an output and no training data are available. Vote of each classifier is the output of that classifier. The final output is produced according to the number of votes for each class and the pre-determined threshold value. In majority voting for a multiple class system, final output can be determined by counting the number of votes for each class and assigning the final output class to the class which takes the highest number of votes. In some applications of majority voting method, rejection option is also considered. If the highest number of votes is not higher than predetermined threshold value, then the input sample is not assigned to any class and rejected. For a decision system with n classes and m classifiers, a decision vector $d = [\, d_1, d_2, \dots, d_n]^T$ formed by the outputs of the classifiers for a given input sample where $d_i \in \{c_1, c_2, \dots, c_m\}$, c_i denotes the label of the i$^{\text{th}}$ class. Let binary characteristic function be defined as follows:

$$B_j(c_i) = \begin{cases} 1 \; if \; d_j = c_i \\ 0 \; if \; d_j \neq c_i \end{cases} \tag{4.10}$$

Then, the majority voting without rejection option is as follows:

$$E(d) = c_i \text{ ,} where \; \forall \; t \in \{1, \dots, m\} \; \sum_{j=1}^{n} B_j(c_t) \leq \sum_{j=1}^{n} B_j(c_i) \tag{4.11}$$

When rejection option is also considered, then the definition of majority voting is as follows:

$$f(x) = \begin{cases} c_i, \; \forall \; t \in \{1, \dots, m\} \; \sum_{j=1}^{n} B_j(c_t) \leq \sum_{j=1}^{n} B_j(c_i) \geq \alpha . m \\ r, \; otherwise \end{cases} \tag{4.12}$$

M is the number of classifiers, and α can take values in the range [0,1]. 0.5 is commonly used in majority voting. In the experiments, α is taken as 0.5. The precision, recall and f-measure results when measured by the pixel-based approach are shown in Table 18. Mean and standard deviation of precision, recall and f-measure values are shown in Table 19.

Table 18: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.11	0.67	0,19
2	0.18	0.71	0,28
3	0.47	0.74	0,58
4	0.38	0.34	0,36
5	0.44	0.49	0,46
6	0.25	0.67	0,36
7	0.21	0.61	0,32
8	0.14	0.96	0,24
9	0.21	0.92	0,34
10	0.04	1	0,08
11	0.14	0.87	0,24
12	0.06	0.34	0,10

Table 18: Precision, Recall and F-Measure by the Pixel Based Approaches (continued)

13	0.14	0.78	0,24
14	0.24	0.73	0,37
15	0.25	0.52	0,34
16	0.37	0.48	0,42
17	0.23	0.62	0,34
18	0.39	0.99	0,56

Table 19: Mean and Standard Deviation of Precision, Recall and F-Measure for the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,24	0,69	0,32
Standard Deviation	0,13	0,21	0,13

The precision, recall and f-measure values when measured by the object based approach are shown in Table 20. Mean and standard deviation of precision, recall and f-measure values of the object based approach are shown in Table 21.

Table 20: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.13	0.75	0,21
2	0.19	1	0,32
3	0.67	0.86	0,75
4	0.60	1	0,75
5	0.40	0.67	0,50
6	0.25	1	0,40
7	0.22	1	0,36
8	0.28	0.83	0,42
9	0.42	1	0,59
10	0.07	1	0,13
11	0.13	1	0,22
12	0.07	0.50	0,13

Table 20: Precision, Recall and F-Measure by the Object Based Approaches (continued)

13	0.15	1	0,26
14	0.33	1	0,50
15	0.64	0.64	0,64
16	0.87	0.93	0,90
17	0.30	0.75	0,43
18	0.50	1	0,67

Table 21: Mean and Standard Deviation of Precision, Recall and F-Measure of thr Object Based Approach

	Precision	Recall	F-Measure
Mean	0,35	0,88	0,45
Standard Deviation	0,23	0,16	0,23

4.3.1.2. Behavior Knowledge Space Method

Behavior-Knowledge Space (BKS) method has been developed by the Concordia research team. The method offers a number of advantages over the other methods in which each classifier offers only one class label as its decision. It contains two stages: (1) the knowledge-modeling stage, which extracts knowledge from the former behavior of classifiers and constructs a behavior-knowledge space; and (2) the operation stage, which is carried out for each test sample, and which combines decisions generated from individual classifiers, enters a specific unit of the constructed space, and makes a final decision by a rule which utilizes the knowledge inside the unit [9]. For a K classifier and M classes decision system, let each classifier sometimes called as experts be represented by e_k, k=1,…,K and each class be represented by $C_1, \dots, C_M$. When an unknown input pattern is given to the system, the classifier e_k produces the output j_k such that $e_k(x) = j_k$ where $j_k \in \{C_1, \dots, C_M\}$ or expert k rejects x. Then, the combination problem is to determine the final decision when K experts give their individual decisions to the unknown input. The formulation of the combination problem is given in *(4.13)*.

$$given \begin{array}{c} e_1(x) = j_1 \\ e_2(x) = j_2 \\ \vdots \\ e_K(x) = j_K \end{array} \overset{?}{\rightarrow} E(x) = j \qquad \textbf{(4.13)}$$

Where E is the panel of multiple classifiers which assigns x to one definitive class j ($j \in \{C_1, \ldots, C_M\}$) or rejects. For each classifier, MxM table is prepared in which the information of the number of samples coming from which class and assigned to which class is recorded. The example representation of tables for three classifiers and three classes system is shown in Figure 25. n_{ijk} denotes the number of samples assigned to class j by the classifier i and actually belongs to class k.

	C_1	C_2	C_3
C_1	n_{111}	n_{121}	n_{131}
C_2	n_{112}	n_{122}	n_{132}
C_3	n_{113}	n_{123}	n_{132}

Classifier 1

	C_1	C_2	C_3
C_1	n_{211}	n_{221}	n_{231}
C_2	n_{212}	n_{222}	n_{232}
C_3	n_{213}	n_{223}	n_{232}

Classifier 2

	C_1	C_2	C_3
C_1	n_{311}	n_{321}	n_{331}
C_2	n_{312}	n_{322}	n_{332}
C_3	n_{313}	n_{323}	n_{332}

Classifier 3

Figure 25: Example representation of knowledge space for each classifier for three classifiers and three classes decision system

Behavior-Knowledge space is a K-dimensional space where each dimension corresponds to the decision of one classifier. Each specific combination of experts is called a unit in BKS and denoted as BKS(e(1), …, e(K)) where e(i) represents the decision of expert i. The number of incoming samples belonging to class m for the given combination of decisions of experts is $n_{e(1),\ldots,e(K)}(m)$ and total number of samples in BKS(e(1), …, e(K)) is $T_{e(1),\ldots,e(K)}$. The best representative class for BKS(e(1), …, e(K)) is represented as $R_{e(1),\ldots,e(K)}$ and the formulation of it is shown in *(4.14)*.

$$R_{e(1),\ldots,e(K)} = \{j \mid n_{e(1),\ldots,e(K)}(j) = \max_{1 \le m \le M} n_{e(1),\ldots,e(K)}(m)\} \qquad \textbf{(4.14)}$$

$$T_{e(1),\ldots,e(K)} = \sum_{m=1}^{M} n_{e(1),\ldots,e(K)}(m) \qquad \textbf{(4.15)}$$

$T_{e(1),\ldots,e(K)}$ should be higher than 0. Otherwise, system does not have any information for that combination of experts. The mostly used option in this case is the rejection for that input sample. If rejection option is not considered, then the a priori information may be used as an option. A priori information is the percentage of classes in the training data without considering the decisions of experts. Let the total number of incoming samples be T and the number of samples from each class be $n_1, n_2, \ldots, n_M$ respectively. Then, the decision rule without rejection is given in *(4.16)*.

$$E(x) = \begin{cases} R_{e(1),\dots,e(K)}\ , when\ T_{e(1),\dots,e(K)} > 0 \\ j, where\ \ n_j/T = \max\limits_{1\le m\le M} n_m\,/T \end{cases} \tag{4.16}$$

The formulation by considering the reject option is given in (17).

$$f(x) = \begin{cases} R_{e(1),\dots,e(K)}\ ,\ \ when\ T_{e(1),\dots,e(K)} > 0\ and\ \frac{n_{e(1),\dots,e(K)}\left(R_{e(1),\dots,e(K)}\right)}{T_{e(1),\dots,e(K)}} \ge \lambda \\ Reject,\ \ Otherwise \end{cases} \tag{4.17}$$

λ is a threshold ($0 \le \lambda \le 1$) which controls the reliable degree of the decision. There are many good properties of BKS method. One of them is that it is optimal combination of multiple experts method in the context that each classifier offers only one class label as its decision. This can be seen from the experimental results given in Table 22 and Table 23 when measured by the pixel based approach and Table 24 and Table 25 for object based approach. In addition to that, it has adaptive learning ability and it does not need classifier-independence assumption.

Table 22: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.13	0.44	0,20
2	0.33	0.42	0,37
3	0.59	0.62	0,60
4	0.62	0.44	0,51
5	0.78	0.17	0,28
6	0.85	0.79	0,82
7	1	0.49	0,66
8	0.46	1	0,63
9	0.41	0.83	0,55
10	0.05	1	0,10
11	0.35	0.91	0,51
12	0.26	0.24	0,25
13	0.28	1	0,44
14	0.21	0.91	0,34

Table 22: Precision, Recall and F-Measure by the Pixel Based Approaches (continued)

15	0.23	0.55	0,32
16	0.22	0.61	0,32
17	0.47	0.82	0,60
18	0.53	1	0,69

Table 23: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,43	0,68	0,45
Standard Deviation	0,26	0,27	0,19

Table 24: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.17	0.50	0,25
2	0.38	0.60	0,46
3	1	0.57	0,73
4	0.33	0.33	0,33
5	0	0	0
6	1	1	1
7	1	0.50	0,67
8	1	0.67	0,80
9	1	1	1
10	0.13	1	0,22
11	0.33	0,50	0,40
12	0.50	0.50	0,50
13	0.20	0,67	0,31
14	0.50	1	0,67
15	0,82	0,50	0,62
16	0.57	0.29	0,38
17	0.60	0.75	0,67
18	1	1	1

Table 25: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,58	0,63	0,56
Standard Deviation	0,36	0,29	0,29

4.3.2. Classifier fusion methods for soft/fuzzy output classifiers

4.3.2.1. Simple Bayes average

If the outputs of the multiple classifier system are given as the posterior probabilities for an input sample x comes from a particular class C_i as$P(x \in C_i|x)$, then it is possible to calculate the average posterior probability of all classifiers for all classes. Let the number of classifiers be K. Then the average posterior probability is calculated simply as given in *(4.18)*.

$$P_{Avg}(x \in C_i/x) = \frac{1}{K}\sum_{k=1}^{K} P_k(x \in C_i/x) \qquad \textbf{\textit{(4.18)}}$$

After calculation of the average posterior probabilities, Bayes decision is made by choosing the class with the highest posterior probability. For Bayes classifiers, this approach can be applied directly. For other classifiers, there is a number of methods to estimate the posterior probability. k-NN classifier is the mostly used method for transformation of the output of classifiers to the posterior probabilities. Assume that the number of samples assigned to class C_i by the classifier k is k_i and the number of all nearest prototype samples is k_{nn}. Then the posterior probability of classifier k for class C_i is given in *(4.19)*.

$$P_k(x \in C_i|x) = \frac{k_i}{k_{nn}} \qquad \textbf{\textit{(4.19)}}$$

Bayes average method can be applied to the obtained posterior probabilities of classifiers and the class with the highest posterior probability is chosen. If the rejection option is considered, then the highest posterior probability is also compared with the predetermined threshold value. The decision methods for a K classifier M class system without rejection and with rejection are given in *(4.20)* and *(4.21)* respectively.

$$E(x) = j \ where \ P_{avg}\left(x \in C_j|x\right) = \ \max_{1 \le m \le M} P_{avg}(x \in C_m|x) \qquad \textbf{\textit{(4.20)}}$$

$$E(x) = \begin{cases} j \ where \ P_{avg}\left(x \in C_j | x\right) = \max_{1 \le m \le M} P_{avg}\left(x \in C_m | x\right) \ and \ P_{avg}\left(x \in C_j | x\right) \ge \lambda \\ Reject, \ Otherwise \end{cases}$$

(4.21)

λ is a predetermined threshold value. In our experiments, the template matching methods produce outputs in different ranges and these outputs are converted to binary values by using the specific threshold values. The values produced before converting the binary values are called raw outputs. In this study, raw outputs are normalized to [0,1] range and used as posterior probabilities. Then, Bayes average is applied to these so-called posterior probabilities. The results obtained from the experiments are given in Table 26, Table 27, Table 28 and Table 29.

Table 26: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.08	0.94	0,14
2	0.11	0.89	0,19
3	0.29	0.05	0,09
4	0.71	0.31	0,43
5	0.88	0.17	0,28
6	0.71	0.26	0,38
7	0	0	0
8	0	0	0
9	0.63	0.21	0,31
10	0.03	0.20	0,04
11	0.11	0.13	0,12
12	0.02	0.03	0,03
13	0.10	0.28	0,15
14	0.08	0.13	0,10
15	0.58	0.28	0,38
16	0.63	0.48	0,55
17	1	0.03	0,05
18	0	0	0

Table 27: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,33	0,24	0,18
Standard Deviation	0,35	0,28	0,17

Table 28: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.12	1	0,21
2	0.16	1	0,28
3	0.75	0.43	0,55
4	1	0.67	0,80
5	1	0.33	0,50
6	1	1	1
7	0	0	0
8	0	0	0
9	0.67	0.40	0,50
10	0.13	1	0,22
11	0.33	1	0,50
12	0.20	0.50	0,29
13	0.20	1	0,33
14	0.50	1	0,67
15	1	0.57	0,73
16	0.83	0.71	0,77
17	1	0.25	0,40
18	0	0	0

Table 29: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,49	0,60	0,43
Standard Deviation	0,41	0,38	0,29

4.3.2.2. Bayes Belief Integration

In Bayes average, all classifiers are treated equally, and the errors produced by classifiers according to the given input type or the output of the classifiers are not considered. Bayes belief integration method produces belief values for each class by considering the outputs of classifiers and then chooses the class by comparing the belief values. In Bayes belief integration method, for every classifier, a matrix called as confusion matrix is built. Confusion matrix consists of the record of the number of incoming samples where rows correspond to the classes from which the input sample is coming and the columns denote the classes to which the input sample is assigned by the classifier. For a K classifier M class decision system, the confusion matrices of them are as shown in Figure 26.

$$\mathrm{PT}_1 = \begin{bmatrix} \mathrm{n_{11}}^1 & \cdots & \mathrm{n_{1M}}^1 \\ \vdots & \ddots & \vdots \\ \mathrm{n_{M1}}^1 & \cdots & \mathrm{n_{MM}}^1 \end{bmatrix} \mathrm{PT}_2 = \begin{bmatrix} \mathrm{n_{11}}^2 & \cdots & \mathrm{n_{1M}}^2 \\ \vdots & \ddots & \vdots \\ \mathrm{n_{M1}}^2 & \cdots & \mathrm{n_{MM}}^2 \end{bmatrix} \cdots \ \mathrm{PT_K} = \begin{bmatrix} \mathrm{n_{11}}^K & \cdots & \mathrm{n_{1M}}^K \\ \vdots & \ddots & \vdots \\ \mathrm{n_{M1}}^K & \cdots & \mathrm{n_{MM}}^K \end{bmatrix}$$

Figure 26: Confusion matrices in a K classifier M class decision system

${n_{ij}}^k$ denotes the number of incoming samples from class C_i and assigned to class C_j by classifier e_k. On the basis of confusion matrix, it is possible to build the belief measures. The belief value of a test sample belonging to class C_i with the information that expert e_k assign it to class C_j is formulated in *(4.22)*.

$$Bel\big(x \in c_i \big| e_k(x)\big) = P(x \in c_i \,|e_k(x) = \ j_k) \ \ where \ i,j = 1, \dots, m \tag{4.22}$$

By using the information in the confusion matrix, $P(x \in c_i \,|e_k(x) = \ j_k)$ can be estimated as in *(4.23)*.

$$P(x \in c_i \,|e_k(x) = j) = \frac{{n_{i,j}}^k}{\sum_{i=1}^{K} {n_{i,j}}^k} \tag{4.23}$$

Having defined such a belief measure for each classifier, we can combine them in order to create new belief measure of the multiple classifier system as follows:

$$Bel(i) = P(x \in c_i) \frac{\prod_{k=1}^{K} P(x \in c_i | e_k(x) = j_k)}{\prod_{k=1}^{K} P(x \in c_i)} \tag{4.24}$$

The probabilities used in the above formula can be easily estimated from the confusion matrix. If rejection option is not considered, then the class with the highest combined belief measure is chosen as the final classification decision. The formulation of that decision is given in *(4.25)*. If the rejection option is considered, then the highest belief value obtained is compared with the predetermined threshold value and the sample is assigned to a class or rejected according to the result of comparison. The formulation of decision with rejection option is given in *(4.26)*.

$$E(x) = j \textit{ where } Bel(j) = \max_{1 \le m \le M} Bel(m) \tag{4.25}$$

$$E(x) = \begin{cases} j \textit{ where } Bel(j) = \max_{1 \le m \le M} Bel(m) \textit{ and } Bel(j) \ge \lambda \\ \textit{Reject, Otherwise} \end{cases} \tag{4.26}$$

λ is a predetermined threshold value in the interval [0,1]. In the experiments performed, the rejection option is not considered and the results are given in Table 30, Table 31 Table 32, Table 33.

Table 30: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.11	0.66	0,19
2	0.14	0.36	0,20
3	0.84	0.73	0,78
4	0.50	0.31	0,38
5	0.29	0.14	0,19
6	0.25	0.67	0,36
7	0.20	0.58	0,30
8	0.15	0.96	0,26
9	0.27	0.92	0,42
10	0.05	1	0,10
11	0.15	0.84	0,25
12	0.06	0.34	0,10

Table 30: Precision, Recall and F-Measure by the Pixel Based Approaches (continued)

13	0.18	0.72	0,29
14	0.24	0.73	0,36
15	0.31	0.48	0,38
16	0.39	0.48	0,43
17	0.24	0.60	0,34
18	0.40	0.99	0,57

Table 31: Mean and Standard Deviation of Precision, Recall and F-Measure for the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,27	0,64	0,33
Standard Deviation	0,19	0,25	0,17

Table 32: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.13	0.75	0,21
2	0.17	0.60	0,26
3	1	0.86	0,92
4	0.75	1	0,86
5	0	0	0
6	0.25	1	0,40
7	0.22	1	0,36
8	0.31	0.83	0,45
9	0.56	1	0,71
10	0.08	1	0,14
11	0.14	1	0,25
12	0.07	0.50	0,13
13	0.19	1	0,32
14	0.33	1	0,50
15	0.89	0.57	0,70
16	0.93	0.93	0,93
17	0.38	0.75	0,50
18	0.50	1	0,67

Table 33: Mean and Standard Deviation of Precision, Recall and F-Measure for the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,38	0,82	0,46
Standard Deviation	0,32	0,27	0,28

4.3.2.3. Fuzzy templates

In a K classifier $e_1, e_2, \ldots, e_K$,and M class $C_1, C_2, \ldots, C_M$ decision system, each classifier produces an output vector for an input sample x , such that $e_i(x) = [d_{i,1}(x), \ldots, d_{i,M}(x)]^T$where $d_{i,j}(x)$ represents the degree of support of classifier e_i to that input sample x comes from class j. This support can be posterior probability, belief value, certainty possibility etc. [33]. The support value does not have to be coming from statistical classifiers. The support vectors are combined by the aggregation rule and decision profile matrix is obtained as shown in (4.28).

$$e_1(x) = \begin{bmatrix} d_{1,1}(x) \\ \cdots \\ d_{1,i}(x) \\ \cdots \\ d_{1,M}(x) \end{bmatrix} e_2(x) = \begin{bmatrix} d_{2,1}(x) \\ \cdots \\ d_{2,i}(x) \\ \cdots \\ d_{2,M}(x) \end{bmatrix} \ldots e_K(x) = \begin{bmatrix} d_{K,1}(x) \\ \cdots \\ d_{K,i}(x) \\ \cdots \\ d_{K,M}(x) \end{bmatrix} \qquad \textbf{\textit{(4.27)}}$$

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \cdots & d_{1,j}(x) & \cdots & d_{1,M}(x) \\ \vdots & & \vdots & & \vdots \\ d_{i,1}(x) & \cdots & d_{i,j}(x) & \cdots & d_{i,M}(x) \\ \vdots & & \vdots & & \vdots \\ d_{K,1}(x) & \cdots & d_{K,j}(x) & \cdots & d_{K,M}(x) \end{bmatrix} \qquad \textbf{\textit{(4.28)}}$$

Let Z = $\{Z_1, Z_2, \ldots, Z_N\}$, $Z_j \in R^p$ be the crisply labeled set of training data. The fuzzy template of the class *i* is then defined as KxM matrix $F_i = \{f_i(k,s)\}$ the elements of which are obtained from:

$$f_i(k,s) = \frac{\sum_{j=1}^{N} Ind(Z_j,i) d_{k,s}(Z_j)}{\sum_{j=1}^{N} Ind(Z_j,i)} \qquad \textbf{\textit{(4.29)}}$$

Where $Ind(Z_j, i)$ is an indicator function with value 1 if Z_j comes from class i and 0 otherwise. By this function, only samples coming from class i is considered and average values of the support vectors of classifiers are used for the construction of decision profile which is called fuzzy template. After constructing the fuzzy templates for each class, the similarity of decision profile of the input sample with each class' fuzzy template is measured as shown in *(4.30)*:

$$S(F_i, DP(x)) = 1 - \frac{1}{Km}\sum_{k=1}^{K}\sum_{s=1}^{m}(f_i(k,s) - d_{k,s}(x))^2 \quad \textbf{\textit{(4.30)}}$$

The class which has maximum similar fuzzy template with the decision profile of the given input sample is chosen as the final classification decision. The formulation of fuzzy template decision without rejection is given in *(4.31)*. When rejection option is also considered, the decision formulation is as shown in *(4.32)*.

$$E(x) = j \text{ where } S(F_i, DP(x)) = \text{minx}_{1\leq m\leq M}\ S(F_m, DP(x)) \quad \textbf{\textit{(4.31)}}$$

$$E(x) = \begin{cases} j \text{ where } S(F_i, DP(x)) = \min_{1\leq m\leq M} S(F_m, DP(x)) \text{ and } S(F_i, DP(x)) \leq \lambda \\ \text{Reject, Otherwise} \end{cases} \quad \textbf{\textit{(4.32)}}$$

λ is predetermined threshold value and does not have to be in [0,1] range. Since, similarity measure results can have values in a broad range of interval. In the experiments, rejection option is not considered and the experimental results are given in Table 34, Table 35, Table 36, Table 37.

Table 34: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.07	1	0,13
2	0.12	0.91	0,21
3	0.22	0.61	0,32
4	0.20	0.72	0,31
5	0.20	0.54	0,29
6	0.12	0.67	0,20
7	0	0	0
8	0.14	0.33	0,20
9	0.04	1	0,08
10	0.01	1	0,02
11	0.03	0.87	0,06
12	0.04	0.75	0,08

Table 34: Precision, Recall and F-Measure by the Pixel Based Approaches (continued**)**

13	0.06	0.65	0,11
14	0.02	0.73	0,04
15	0.08	0.93	0,15
16	0.14	0.92	0,24
17	0.15	0.81	0,25
18	0.10	0.70	0,18

Table 35: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,10	0,73	0,16
Standard Deviation	0,07	0,26	0,10

Table 36: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.12	1	0,21
2	0.21	1	0,34
3	0.50	0.86	0,63
4	0.27	1	0,43
5	0.43	1	0,60
6	0.33	1	0,50
7	0	0	0
8	0.36	0.67	0,47
9	0.28	1	0,43
10	0.04	1	0,07
11	0.09	1	0,17
12	0.13	0.50	0,20
13	0.17	1	0,29
14	0.11	1	0,20
15	0.38	0.79	0,51
16	0.62	0.93	0,74
17	0.57	1	0,73
18	0.40	1	0,57

Table 37: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,28	0,87	0,39
Standard Deviation	0,18	0,26	0,22

4.3.2.4. Proposed optimization based fusion method

As mentioned before, the combination process can be performed in decision level or score level. In score level fusion, raw outputs or scores of each classifier is used in the fusion rule. Some researchers used the Support Vector Machine (SVM) to fuse the score level outputs of the classifiers [39]. In decision level fusion, on the other hand, final decisions of the classifiers are used. In order to obtain the optimal results, optimal thresholds and the fusion rule should be chosen. There have been very few studies in optimizing fusion system performance [40, 41]. Two formal designs were used to obtain optimal results. In one of them, two-step optimization procedure is used. First, decision thresholds are estimated and fixed. Then, optimal fusion rule is chosen. In the other design approach, optimal thresholds and optimal fusion rule are searched simultaneously [40]. For an N-classifier system, the dimension of the search space is N+1, N for the thresholds of each classifiers and 1 for the fusion rule. In the proposed method, optimization is performed for score level fusion. The combination procedure is defined as the weighted sum of the raw outputs of the classifiers and the final binary result is obtained by comparing the resultant weighted sum with threshold value. Optimization is used for finding optimum weights and thresholds for the fusion of the given template matching algorithms. In order to turn the fusion problem into an optimization problem, first of all, a cost function should be defined. Then, a search method should be chosen.

The template matching algorithms analyzed in this paper produces raw outputs for each pixel and then, they apply thresholding and produce a binary image as their detection output. After thresholding, if they detect the object, they assign '1' value to the pixels in a rectangular boundary, otherwise, pixels take '0' values. The proposed method takes the raw outputs of the algorithms as input and tries to find optimum weighting of the algorithms and afterwards optimum threshold value to reach the best fusion result.
Assume that, we have N classifiers and M training data. In the optimization method, all classifiers are combined linearly and then the output is produced by applying thresholding. Hence, we have N weights for each classifier and M inequalities where the direction of each inequality is obtained from the ground truth and one threshold value. The aim of the method is to estimate the best values for N weights and one threshold. The mathematical representation of these inequalities is shown in *(4.33)*:

$$w_1 * C_{1,1} + w_2 * C_{2,1} + \cdots + w_N * C_{N,1} \gtrless T \ (If\ ground\ truth\ says\ there\ is\ object >, otherwise <)$$

$$w_1 * C_{1,2} + w_2 * C_{2,2} + \cdots + w_N * C_{N,2} \gtrless T \ (If\ ground\ truth\ says\ there\ is\ object >, otherwise <)$$

$\cdots$

$$w_1 * C_{1,M} + w_2 * C_{2,M} + \cdots + w_N * C_{N,M} \gtrless T (If\ ground\ truth\ says\ there\ is\ object >, otherwise <) \qquad \textbf{\textit{(4.33)}}$$

w_i represents the weight for classifier "i" and $C_{i,j}$ represents the raw output of classifier "i" for the "j" th data.

Inequalities are bidirectional. In order to convert the inequalities to have just one direction, we included the information provided by the ground truth. If the ground truth includes an object, it has the value "-1" for the regions occupied by the objects, and has "1" for the regions not including the object.

$$GT(u) = -1\ if\ there\ is\ object$$
$$GT(u) = 1\ if\ there\ is\ no\ object$$

Now, we can define a function $f(w_1, w_2, \dots, w_N, u)$ such that:

$$f(w_1, w_2, \dots, w_N, u) = w_1 * C_1(u) + w_2 * C_2(u) + \cdots + w_N * C_N(u) \qquad \textbf{\textit{(4.34)}}$$

where the inequalities can be written for just one direction as follows:

$$GT(u) f(w_1, w_2, \dots, w_N, u) - GT(u) * T < 0 \qquad \textbf{\textit{(4.35)}}$$

The aim is to find optimum w and T values, so that maximum number of inequalities are satisfied. The final cost function is defined in (4.36). In order to obtain weights in a reasonable range, the constraint is defined as in (4.37).

$$J(w_1, w_2, \ldots, w_N, T) = \sum_{i=1}^{M} (\max\left(\left(GT\big(u(i)\big) * f\big(w_1, w_2, \ldots, w_N, u(i)\big) - GT\big(u(i)\big) * T\right), 0\right))^2 \quad \textbf{\textit{(4.36)}}$$

$$w_1 + w_2 + \cdots + w_N = 1 \quad \textbf{\textit{(4.37)}}$$

The final optimization problem given in (4.34) and (4.35) is a constrained optimization problem. In order to solve this problem, '*fmincon*' function given in the optimization toolbax of the matlab was used. fmincon Active Set algorithm was used as search algorithm. Detailed information for fmincon Active Set Algorithm is given in Appendix B.

In order to make the problem more continuous, the cost function is defined so as to minimize the distance between the weighted sum of the wrongly classified pixels and the threshold. This definition may cause problem in situations such that there are many wrongly classified pixels very close to the threshold versus one wrongly classified pixel far away from the threshold. In that case, the effect of distant one will be higher, although the number of pixels close to the threshold is much more.

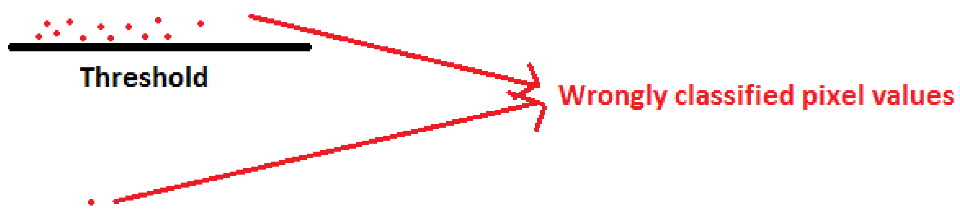


Figure 27: The initial situation for the classifier outputs and the threshold.

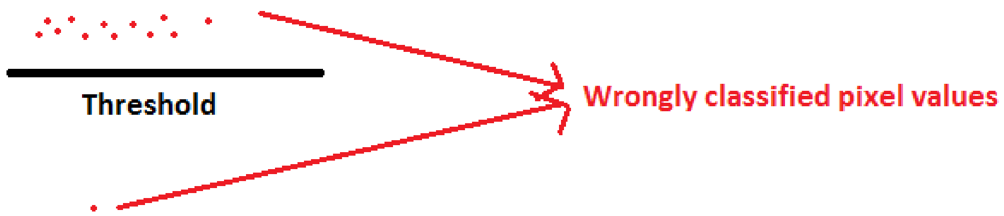


Figure 28: The threshold after optimization.

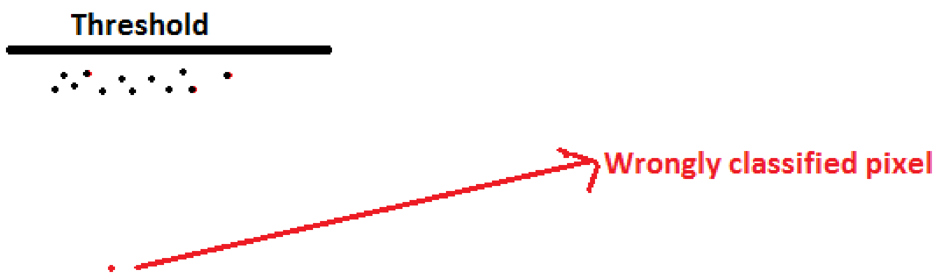


Figure 29: The desired threshold.

This problem may be solved by modifying the cost function as follows:

$$J(w_1, w_2, \dots, w_N, T) = \sum_{i=1}^{M} \mathrm{y_i}(w_1, w_2, \dots, w_N, T)$$

$$\mathrm{y_i}(w_1, w_2, \dots, w_N, T)$$
$$= \begin{cases} 0, if \ \left(\max\left(\left(GT\big(u(i)\big) * f\big(w_1, w_2, \dots, w_N, u(i)\big) - GT\big(u(i)\big) * T\right), 0\right)\right) = 0 \\ 1, if \ \left(\max\left(\left(GT\big(u(i)\big) * f\big(w_1, w_2, \dots, w_N, u(i)\big) - GT\big(u(i)\big) * T\right), 0\right)\right) > 0 \end{cases}$$

(4.35)

But, this may make the problem more discrete, and it makes hard to optimize the solution. The experiment results by using the cost function in (4.34) is shown in Tables 38, 39, 40 and 41 for pixel based approach and object based approach.

Table 38: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.11	0.87	0,20
2	0.23	0.70	0,34
3	0.14	0.03	0,05
4	0.74	0.14	0,23
5	0.80	0.09	0,17
6	0.36	0.42	0,39
7	0	0	0
8	1	0.11	0,20
9	0.36	0.38	0,37
10	0.05	0.40	0,10
11	0.35	0.19	0,25
12	0.06	0.07	0,06
13	0.17	0.22	0,19
14	0.36	0.33	0,34
15	0.28	0.48	0,36
16	0.45	0.38	0,41
17	0.31	0.19	0,24
18	0.27	0.04	0,08

Table 39: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,34	0,28	0,22
Standard Deviation	0,27	0,24	0,13

Table 40: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.15	1	0,26
2	0.36	1	0,53
3	0.50	0.14	0,22
4	0	0	0
5	0	0	0
6	0.50	1	0,67
7	0	0	0
8	1	0.33	0,50
9	0.67	0.40	0,50
10	0.25	1	0,40
11	1	0.50	0,67
12	0.50	0.50	0,50
13	0.25	0.33	0,29
14	1	1	1
15	0.58	0.50	0,54
16	1	0.36	0,53
17	1	0.50	0,67
18	1	0.50	0,67

Table 41: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,54	0,50	0,44
Standard Deviation	0,39	0,36	0,27

Proposed method was also performed by only including the correlation based template matching method and histogram based template matching method. The experimental results are given in Table 42, Table 43, Table 44 and Table 45.

Table 42: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0,10	0,92	0,18
2	0,20	0,77	0,32
3	0,26	0,42	0,32
4	0,39	0,53	0,45
5	0,33	0,32	0,32
6	0,20	0,63	0,30
7	0,09	0,16	0,11
8	0,20	0,41	0,27
9	0,07	0,88	0,13
10	0,01	0,80	0,02
11	0,05	0,65	0,10
12	0,037	0,45	0,07
13	0,08	0,52	0,13
14	0,53	0,53	0,53
15	0,12	0,80	0,21
16	0,23	0,89	0,37
17	0,16	0,77	0,27
18	0,14	0,57	0,23

Table 42: Precision, Recall and F-Measure by the Pixel Based Approaches (continued)

Table 43: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,18	0,61	0,24
Standard Deviation	0,14	0,21	0,14

Table 44: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0,13	1	0,24
2	0,29	1	0,45
3	0,63	0,71	0,67
4	0,75	1	0,86
5	1	0,67	0,80
6	0,50	1	0,67
7	0	0	0
8	0,57	0,67	0,62
9	0,40	0,80	0,53
10	0,05	1	0,09
11	0,15	1	0,27
12	0,20	0,50	0,29
13	0,23	1	0,38
14	0,25	1	0,40
15	0,41	0,79	0,54
16	0,65	0,93	0,76
17	0,67	1	0,80
18	0,40	1	0,57

Table 45: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,40	0,84	0,50
Standard Deviation	0,27	0,26	0,25

4.3.3 Classifier selection methods

4.3.3.1. Static classifier selection

The methods discussed so far are based on combining the outputs of the classifiers. Classifier selection methods, on the other hand, choose the result of only one classifier as the final output.

Classifier selection methods are performed at different levels or stages. Classifier selection may be either performed in the training stage and then the chosen classifier is used in the test stage or classifier selection may be performed in the test stage. If the classifier selection operation is performed in the training stage and the selected classifier is not changed for the input test samples, then this selection process is called as the “static classifier selection”. In the training stage, the selection process may be performed by checking different metrics for the classifier outputs. Precision, recall, f-measure, process time or big-O complexity measurement function are ruling parameters in static classifier selection and any one of them may be used depending on to the success criteria of the operation. In a binary classification system, if the false positives are not very important for a system and the system only focuses on the detection rate, then only the recall information is important for this system. Intrusion detection system may be given as an example for this kind of systems. Intrusion detection systems focus on keeping the number of intrusion instances detected by the system as high as possible. In some systems, precision of the results are very important and false alarms cause undesirable circumstances. The friend-enemy detection systems may be given as an example to this kind of systems.

F-Measure assigns importance to both precision and recall values in some amount. In the experiments, the classifier selection was performed by comparing the F-Measure values. Because, both precision and recall values were important in the evaluation of the performances of template matching algorithms. Experiments were performed for 18 images. Leave-one-out approach is used and experiments are repeated 18 times. For each image, the remaining 17 images are used for training. The average F-Measure values of these 17 images are compared and the best template matching method is chosen and used for the test image. The experiment results are given in Table 46, Table 47, Table 48 and Table 49.

Table 46: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.17	0.65	0,27
2	0.17	0.42	0,24
3	0.81	0.68	0,74
4	0.21	0.57	0,30
5	0.17	0.27	0,21
6	0.18	0.67	0,28
7	1	0.41	0,58
8	0.30	0.89	0,44
9	0.45	1	0,62
10	0.05	1	0,09
11	0.19	0.71	0,31
12	0.12	0.31	0,17

Table 46: Precision, Recall and F-Measure by the Pixel Based Approaches (continued)

13	0.11	0.70	0,19
14	0.20	0.80	0,32
15	0.21	0.56	0,31
16	0.20	0.78	0,32
17	0.52	0.83	0,64
18	0.71	0.97	0,82

Table 47: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,32	0,68	0,38
Standard Deviation	0,27	0,22	0,21

Table 48: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0.19	0.75	0,30
2	0.19	0.60	0,29
3	1	0.71	0,83
4	0.25	1	0,40
5	0	0	0
6	0.20	1	0,33
7	1	0.50	0,67
8	0.50	0.83	0,63
9	1	1	1
10	0.08	1	0,14
11	0.20	1	0,33
12	0.17	0.50	0,25
13	0.17	1	0,29
14	0.25	1	0,40
15	0.50	0.57	0,53
16	0.52	0.86	0,65
17	0.67	1	0,80
18	1	1	1

Table 49: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,44	0,80	0,49
Standard Deviation	0,35	0,28	0,29

4.3.3.2. Dynamic classifier selection

Dynamic classifier selection method has been proposed recently as an alternative approach to the Multiple Classifier Systems (MCSs). Roughly speaking, selection based MCSs are based on a function that, for each test pattern, dynamically select the classifier that correctly classifies it. It is also pointed out that selection-based MCSs do not need the assumption of independent classifiers [36]. Independency of classifiers is very important in combination-based MCSs and directly affects the performance of the classification results. In selection-based MCSs, it is assumed that for each test pattern, there is at least one classifier that correctly classifies it. It is easy to see that this assumption is much more easy to be satisfied than the independence assumption. The potentialities of dynamic classifier selection have been motivated by both theoretically [37] and experimental results [6, 33]. As mentioned in the related work, under the assumptions of decision regions complementarity and decision boundaries complementarity, the optimal Bayes classifier can be obtained by the selection of non-optimal classifiers.

There are two popular dynamic classifier selection methods. These are priori selection method and posteriori selection method. In priori selection method, the selection is performed without knowing the class assigned by any of the classifiers to the test pattern. k-Nearest Neighbors of the test pattern in the training set are chosen and for each classifier, the probability of classifying correctly the test pattern is determined by the success rate in the k-NN samples in the training set. There is an uncertainty in the definition of the neighborhood size. In order to overcome this problem, the effect of samples in the training set may be weighted by the Euclidean distances to the test sample. The other method is a posteriori selection method. In this method, the information of the class assigned by each classifier to the given test sample is exploited. The success rate of a classifier is measured by the samples in the training set which are assigned to the same class with the test sample by that classifier. The ratio of correctly classified samples to the total number of samples assigned to that class gives the success rate of the classifier. For each classifier, the success rate is calculated and the one that has the highest success rate is chosen. Choosing the training samples by k-NN sometimes causes high calculation time or some confusion in determining the neighborhood size. In order to solve these problems, the samples in the training set and the input samples can be partitioned. There are many methods for the partition forming and grouping of features of the samples is a widely used approach.
In this study, two dynamic classifier selection methods are developed by considering some important features of the input sample. First edge density and then Scale Invariant Feature Transform (SIFT) [24] features are considered. In the first method, the priori selection

method is performed by grouping the input samples and the training samples into 64 partitions based on the edge intensity of the samples. The process of the method consists of two stages. One of them is the training stage. As in other methods which include training, one image is chosen as test image and the remaining 17 images are used for training. In each image, Canny edge detection operation is performed. Example input image and the resulting image after Canny edge detector is performed are as shown in Figure 30 and Figure 31 respectively. Then, each image is divided into 238 rectangles with size 64x64 as shown in Figure 32. The edge intensity of each rectangle is calculated by the rate of bright pixels to dark pixels in the corresponding locations of the Canny result. This 64x64 rectangle is taken as an input and partitioned according to the edge intensity. The success rate of the classifier for that input is calculated by looking the intersection of the 64x64 window with the corresponding window in the ground truth image. Then, the mean of the success rate of all input samples partitioned to same type is calculated and assigned to that classifier as the final success rate for that input sample. This operation is repeated for each classifier and the success rate of each classifier for each sample type is calculated. In the test image, again Canny edge detection method is performed and the image is divided into 238 rectangles with size 64x64. Then, each window is taken as input and the type of input is determined by checking the edge intensity. The output of the classifier which gives highest average success rate for that input type is copied to the corresponding section of the output. This operation is performed for each window in the test sample and finally, the output image is produced. In output image, pixel-based and object-based performance measures are performed and the results are shown in Table 50, Table 51, Table 52 and Table 53.

Figure 30: Original image

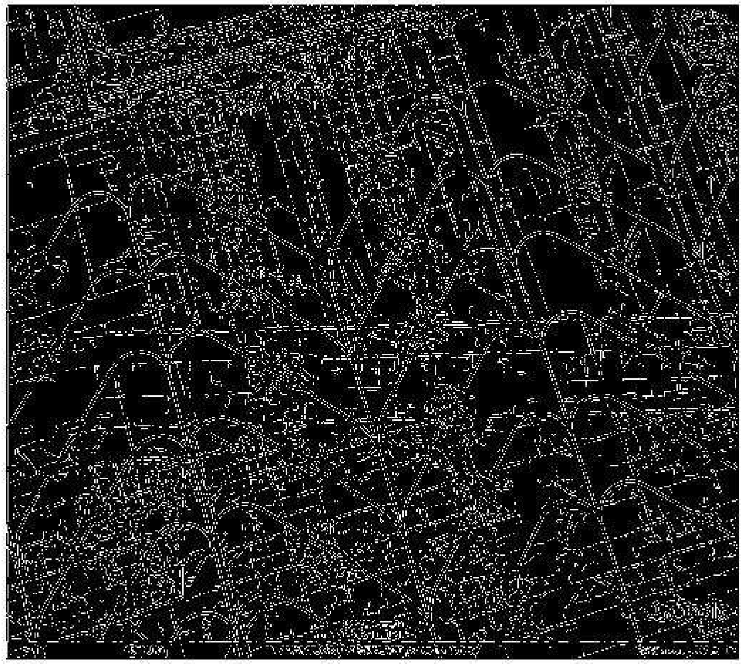


Figure 31: The image after Canny edge detection performed

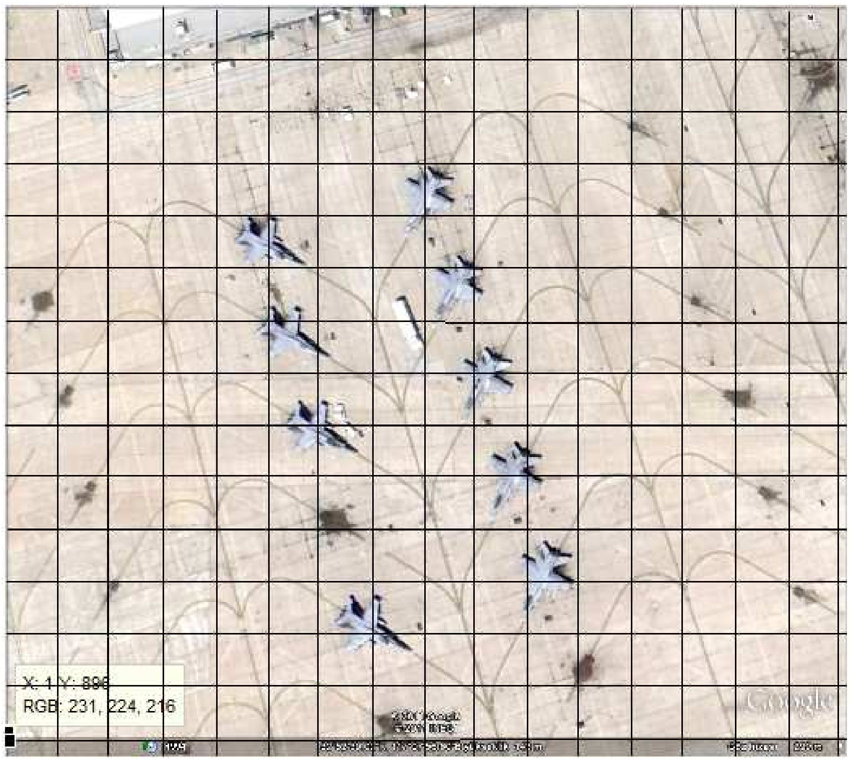


Figure 32: The image divided into 64x64 windows

Table 50: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.17	0.65	0,27
2	0.17	0.42	0,24
3	0.81	0.68	0,74
4	0.21	0.57	0,31
5	0.17	0.27	0,21
6	0.18	0.67	0,28
7	1	0.41	0,58
8	0.30	0.89	0,44
9	0.45	1	0,62
10	0.05	1	0,09
11	0.19	0.71	0,31
12	0.12	0.31	0,17
13	0.11	0.70	0,19
14	0.20	0.80	0,32
15	0.21	0.56	0,31
16	0.20	0.78	0,32
17	0.52	0.83	0,64
18	0.71	0.97	0,82

Table 51: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,32	0,68	0,38
Standard Deviation	0,27	0,22	0,21

Table 52: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0,18	0,50	0,27
2	0	0	0
3	1	0,86	0,92
4	0,11	0,33	0,17
5	0	0	0

Table 52: Precision, Recall and F-Measure by the Object Based Approaches (continued)

6	0,33	1	0,50
7	0,20	0,50	0,29
8	0,45	0,83	0,59
9	0,83	1	0,91
10	0,08	1	0,14
11	0,20	1	0,33
12	0,20	0,50	0,29
13	0,18	1	0,30
14	0,20	1	0,33
15	0,54	0,50	0,52
16	0,56	0,64	0,60
17	0,50	0,75	0,60
18	1	1	1

Table 53: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,36	0,69	0,43
Standard Deviation	0,32	0,34	0,30

In the second method, Scale Invariant Feature Transform (SIFT) [24] algorithm is used to describe local features in the images. SIFT is widely used in object recognition, video tracking and identification algorithms. SIFT transforms the image into a large collection of feature vectors, each of which is invariant to image translation, scaling and rotation. In the developed dynamic classifier selection method, the image is again divided into windows with size 64x64 and the SIFT vector is extracted from this window. The success rate of each template matching method is calculated in this window by looking the pixel based success measure results. The SIFT vector obtained from the window is assigned to the set of descriptor vectors of the template matching method which gives the highest success rate at this window. At the end of the training stage, each template matching method includes a set of feature descriptor vectors. Then, the test image is again divided into 64x64 windows. At each window SIFT vector is extracted. The differences between the extracted SIFT vector and the SIFT vectors in the sets of the template matching methods are calculated and the most similar SIFT vector is determined. The result of the template matching method which includes the most similar SIFT vector is used as a result of that window. This operation is performed for each window in the test image and then the success rates are calculated. As in the other methods including training stage, for each test image, the remaining 17 images are used for training. The experiment results are given in Table 50, Table 51, Table 52 and Table 53.

Table 54: Precision, Recall and F-Measure by the Pixel Based Approaches

Image No	Precision	Recall	F-Measure
1	0.11	0.24	0,15
2	0	0	0
3	0.80	0.76	0,78
4	0.11	0.26	0,16
5	0.10	0.13	0,11
6	0.18	0.67	0,28
7	0.28	0.43	0,34
8	0.24	0.81	0,38
9	0.45	0.88	0,59
10	0.05	1	0,10
11	0.20	0.71	0,31
12	0.11	0.31	0,17
13	0.12	0.61	0,20
14	0.13	0.80	0,22
15	0.19	0.38	0,25
16	0.21	0.53	0,30
17	0.42	0.68	0,52
18	0.47	0.97	0,63

Table 55: Mean and Standard Deviation of Precision, Recall and F-Measure of the Pixel Based Approach

	Precision	Recall	F-Measure
Mean	0,23	0,56	0,30
Standard Deviation	0,20	0,30	0,21

Table 56: Precision, Recall and F-Measure by the Object Based Approaches

Image No	Precision	Recall	F-Measure
1	0,19	0,75	0,30
2	0,19	0,60	0,29
3	1	0,71	0,83
4	0,25	1	0,40
5	0	0	0
6	0,20	1	0,33
7	1	0,50	0,67
8	0,50	0,83	0,63
9	1	1	1
10	0,08	1	0,14
11	0,20	1	0,33
12	0,17	0,50	0,25
13	0,17	1	0,29
14	0,25	1	0,40
15	0,50	0,57	0,53
16	0,52	0,86	0,65
17	0,67	1	0,80
18	1	1	1

Table 57: Mean and Standard Deviation of Precision, Recall and F-Measure of the Object Based Approach

	Precision	Recall	F-Measure
Mean	0,44	0,80	0,49
Standard Deviation	0,35	0,28	0,29

CHAPTER 5

COMPARISON OF THE RESULTS AND DISCUSSION

The mean and standard deviation of precision, recall and F-measure values of template matching methods are given in Table 58 and Table 60 by pixel based approach and object based approach respectively. Mean and standard deviation of precision, recall and F-measure values of classifier fusion methods outputs are summarized in Table 59 and Table 61 by pixel based approach and object based approaches, respectively.

When F-measure values are compared, the Behavior Knowledge Space method gives the best results for both pixel-based and object based evaluation approaches. The dynamic classifier selection by Canny edge detector is following the Behavior Knowledge Space. The static classifier selection and the Correlation Based Template Matching methods give the same results. The Correlation Based Template Matching generally dominates the other template matching methods when we evaluate the performance by the F-measure metric. Thus, the static classifier selection method always chooses the Correlation Based Template Matching method as its final output. The Behavior Knowledge Space also gives the best results according to the average precision value. The proposed Optimization Based Fusion method follows the Behavior Knowledge Space when evaluated according to the average precision value in both pixel based and object based approaches. The fuzzy templates method gives the best results according to the recall measure in both pixel based and object based approaches. However, low precision values decrease the method's overall performance. Among the classifier selection methods, the static classifier selection directly chooses the best template matching method, which is the correlation based template matching method. By dynamic classifier selection, the overall performance of the selection result is a little improved when evaluated by the pixel based approach. But, the results in object based approach do not support this improvement.

It may be better to compare the methods according to their places in the taxonomy. The correlation based template matching method gives the best results according to F-measure among the template matching methods. The primary reason of this is high precision values. Although the histogram based template matching method gives good results according to the recall values the precision values are comparably much lower than the correlation based method.

The classifier fusion methods used in this study for single class label classifiers are the majority voting and behavior knowledge space methods. The majority voting method, as expected, gives results close to the best classifier but some amount lower than that. The appealing thing in majority voting method is that, it is easy to implement, it does not need training and its running time is low. The majority voting method guarantees to improve the performance according to the worst classifier in the system. If the independence assumption is satisfied, then the improvement compared to the best classifier in the system can be satisfied. The main motivation behind this is that, by providing the independence of classifiers, the errors produced by the classifiers become also independent and by taking the majority of the decisions, we may get rid of these errors. The other classifier producing single crisp class labels is the Behavior knowledge space. The behavior knowledge space method gives better results than the majority voting. Indeed, it produces the best result over all of the methods used in this study. However, the implementation is not as easy as the majority voting method. The behavior knowledge space method needs training and needs large memory usage. Besides all of these, the main advantage of the method is that it does not need any independence assumption. It is highly dependent on the training data, but, produces optimal decisions according to the given training data. However, small training data size may cause overfitting and may cause low performance on the test data. The experimental results showed that the training set used in this study provided enough information to estimate the distribution of the classes.

Fusion methods for soft/fuzzy output classifiers are the simple Bayes average, Bayes belief integration, fuzzy templates and the proposed optimization based fusion methods. Proposed method was applied in two different ways. In one of them, four template matching methods (correlation based, edge based, histogram based and angular radial transform template matching methods) were used in weight and threshold search. In the second way, only the correlation based and histogram based template matching methods were used. According to the F-measure values, the proposed optimization based method applied to correlation and histogram based template matching methods gave the best performance results. Bayes belief integration and the proposed optimization based fusion methods applied to four methods followed it as second and third best performance results respectively. Although Bayes belief integration outperforms to proposed method according to f-measure values, proposed method gave better results according to the precision values.

Table 58: Mean and Standard Deviation of Precision, Recall and F-Measure in Template Matching Methods by the Pixel Based Approach

	Mean Precision	Std. Dev. Precision	Mean Recall	Std. Dev. Recall	Mean F-Measure	Std. Dev. F-Measure
Correlation-Based Template Matching	0,32	0,27	0,68	0,22	0,38	0,21
Edge-Based Template Matching	0,11	0,16	0,26	0,35	0,14	0,19
Histogram Based Template Matching	0,17	0,12	0,70	0,19	0,25	0,13
ART Template Matching	0,09	0,09	0,63	0,27	0,14	0,12

Table 59**:** Mean and Standard Deviation of Precision, Recall and F-Measure in Fusion Methods by the Pixel Based Approach

	Mean Precision	Std. Dev. Precision	Mean Recall	Std. Dev. Recall	Mean F-Measure	Std. Dev. F-Measure
Majority Voting Method	0,24	0,13	0,69	0,21	0,32	0,13
Behavior-Knowledge Space	0,43	0,26	0,68	0,27	0,45	0,19
Simple Bayes Average	0,33	0,35	0,24	0,28	0,18	0,17
Bayes Belief Integration	0,27	0,19	0,64	0,25	0,33	0,17
Fuzzy Templates	0,10	0,07	0,73	0,26	0,16	0,10

Table 59: Mean and Standard Deviation of Precision, Recall and F-Measure in Fusion Methods by the Pixel Based Approach (continued)

Static Classifier Selection	0,32	0,27	0,68	0,22	0,38	0,21
Dynamic Classifier Selection by Canny Edge Detector	0,32	0,27	0,68	0,22	0,38	0,21
Dynamic Classifier Selection by Finding SIFT	0,23	0,20	0,56	0,30	0,30	0,21
Proposed Optimization Based Fusion Method	0,34	0,27	0,28	0,24	0,22	0,13
Proposed Optimization Based Fusion Method for Correlation & Histogram Based Methods	0,18	0,14	0,61	0,21	0,24	0,14

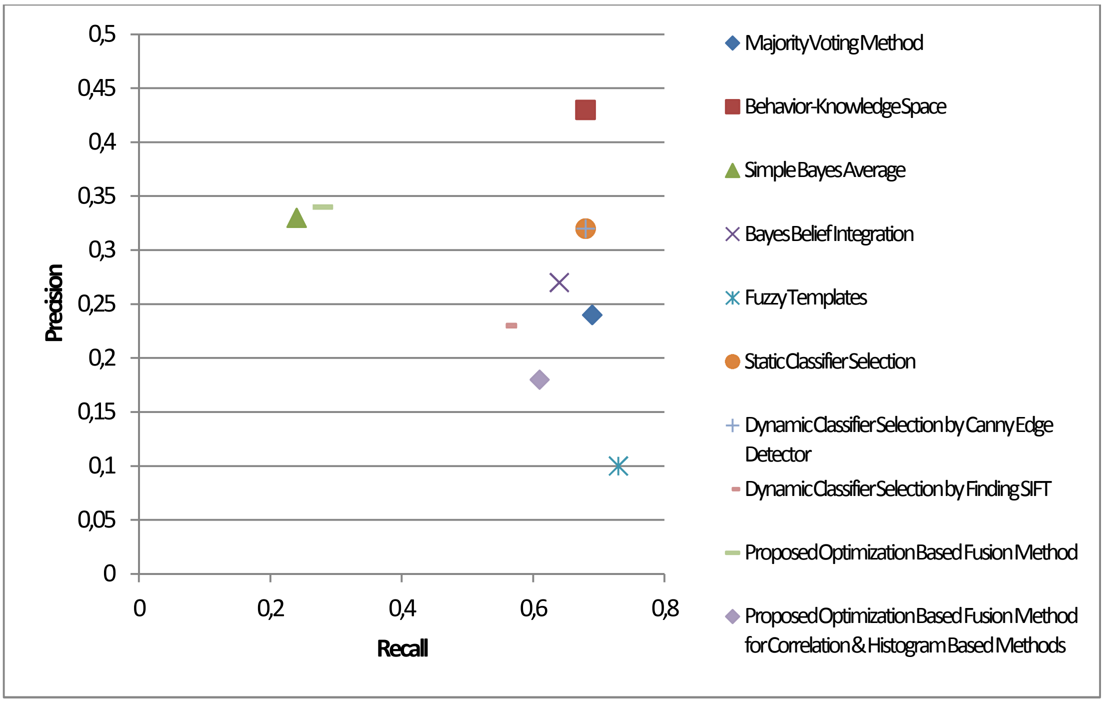


Figure 33: Precision & Recall Curve of The Methods by Pixel Based Performance Measurement

Table 60: Mean and Standard Deviation of Precision, Recall and F-Measure in Template Matching Methods by the Object Based Approach

	Mean Precision	Std. Dev. Precision	Mean Recall	Std. Dev. Recall	Mean F-Measure	Std. Dev. F-Measure
Correlation-Based Template Matching	0,44	0,35	0,80	0,28	0,49	0,29
Edge-Based Template Matching	0,20	0,24	0,30	0,39	0,21	0,26
Histogram Based Template Matching	0,26	0,20	0,93	0,13	0,38	0,22
ART Template Matching	0,20	0,15	0,83	0,30	0,28	0,18

Table 61: Mean and Standard Deviation of Precision, Recall and F-Measure in Fusion Methods by the Object Based Approach

	Mean Precision	Std. Dev. Precision	Mean Recall	Std. Dev. Recall	Mean F-Measure	Std. Dev. F-Measure
Majority Voting Method	0,35	0,23	0,88	0,16	0,45	0,23
Behavior-Knowledge Space	0,58	0,36	0,63	0,29	0,56	0,29
Simple Bayes Average	0,49	0,41	0,60	0,38	0,43	0,29
Bayes Belief Integration	0,38	0,32	0,82	0,27	0,46	0,28
Fuzzy Templates	0,28	0,18	0,87	0,26	0,39	0,22

Table 61: Mean and Standard Deviation of Precision, Recall and F-Measure in Fusion Methods by the Object Based Approach (continued)

Static Classifier Selection	0,44	0,35	0,80	0,28	0,49	0,29
Dynamic Classifier Selection by Canny Edge Detector	0,36	0,32	0,69	0,343	0,43	0,30
Dynamic Classifier Selection by Finding SIFT	0,44	0,35	0,80	0,28	0,49	0,29
Proposed Optimization Based Fusion Method	0,54	0,39	0,50	0,36	0,44	0,27
Proposed Optimization Based Fusion Method for Correlation & Histogram Based Methods	0,40	0,27	0,84	0,26	0,50	0,25

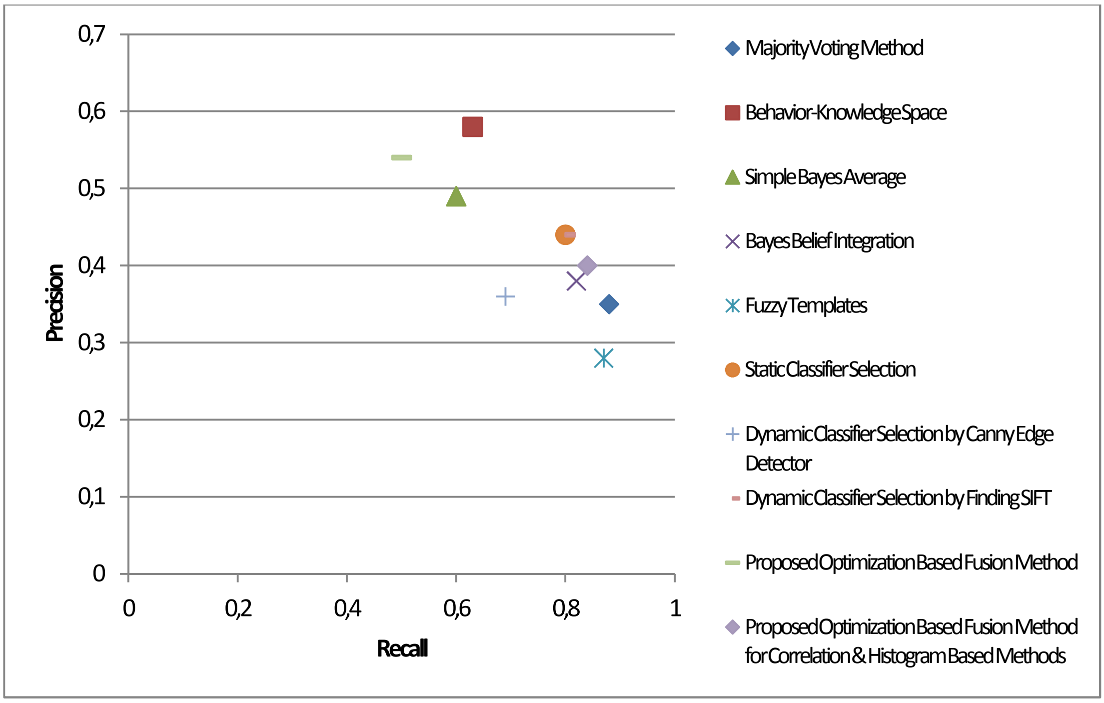


Figure 34: Precision & Recall Curve of The Methods by Object Based Performance Measurement

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, classifier fusion methods have been analyzed for one of the most important problems in computer vision, namely, template matching. Although the fusion methods in the literature were not developed for template matching applications, they have been adapted to be used for fusing template matching methods. Performance evaluation for binary results are done based on two methods, called the pixel based approach and the object based approach. Besides, a new classifier fusion method, called optimization based fusion method, is proposed.

Measuring the performance of the template matching methods is as important as progressively developing new methods. Since, absence of a convention may misdirect in the development of new methods. In order to predefine a measurement convention and then analyze the results, two performance measurement methods are developed: pixel based approach and object based approach. The main motivation of the pixel based approach was to measure the intersection of the binary output image with the binary ground truth image. The objects in the images are not considered as whole, but as a set of pixels. However, there could be some defective points in measuring the output image and the ground truth by looking at only their pixel wise intersection. Small variances in pixel locations in the output image compared to the ground truth image are highly punished in pixel based performance measurement method. In order to take the output image as a set of objects, and measure the precision and recall values by considering the clustered hit pixels as whole, object based approach was developed. By this approach, high punishment of small variance in pixel location in pixel based approach is also solved. Dependency to the boundaries of the objects in the ground truth image is reduced. When finding the best, worst method according to one of the values, precision, recall, fmeasure, or comparing two methods to find which one outperform the other, two approaches were agree but, pixel based approach produced much smaller values for precision, recall and f-measure due to high dependency to pixel locations. Although the pixel based approach may give some information about the performance measure, object based approach produces more consistent results and object based approach is more tolerant to error.

Once the literature is considered in terms of template matching methods, it can be seen that many methods have been developed so far. Correlation based method is the most popular one among them. The basic idea behind the template matching is to measure the similarity of two regions. By correlation based method, the pixel wise difference of two regions is measured. One of the main drawbacks of this method is its high computational complexity. Because of that, runtime of the algorithm is too high. In order to ease the experiments and decrease the runtime, fast normalized cross correlation method may be used in the future studies. The other methods used in the study are edge based template matching method, histogram based template matching and ART template matching. In edge based template matching method, the similarity measure is performed on the Canny edge detector output of the image and the template. In histogram based method, histograms of the image and the template are considered and the similarity of them is analyzed. In ART template matching method, the image and the template are transformed to rotation invariant space and the similarity measurement is performed afterward. At the end of the each template matching method, the obtained similarity is compared with a threshold value and detection is performed according to the result of the comparison. The fusion methods for binary outputs use the outputs of template matching methods after this comparison. Since; threshold selection for template matching methods is not in the scope of this thesis. Since, the motivation of the thesis is to study on classifier fusion methods to combine or select template matching methods to improve the performance of the system. But, for further studies, classifier fusion methods for single label/binary output classifiers may be extended to include finding the optimal thresholds. By this operation, the performance of the classifier fusion methods on weak and strong classifiers may be analyzed.

Although, there are some studies on classifier fusion published twenty or thirty years ago, classifier fusion topic has gained its popularity newly. Progressive development of new methods is becoming hard and combining the existing methods seems to be a better alternative. There are many different classifiers and classifiers may produce different type of outputs. In order to combine different outputs, different classifier fusion methods are necessary. In the taxonomy of the classifier fusion methods, the classifier outputs generally have indicative role. Three kinds of classifier outputs are considered in classifier combination. These are single label output, class ranking output and soft/fuzzy output classifier fusion methods. In single label output classifiers, majority voting and behavior knowledge space methods have been considered and analyzed. In all fusion methods in this study, classifier fusion problem is reduced to two class classifier fusion problem to adapt the solutions to template matching. Majority voting method assumes the independence of classifiers. In this study, the independence assumption was satisfied only the level in which used template matching methods provide. In future studies, the number of used template matching methods may be increased and the more independent template matching method set may be obtained. Behavior knowledge space method does not need independence assumption. The main drawback of the method is high memory usage. But, in our problem, there were four classifiers and two classes. Hence, memory usage was not excessive and the method was appropriate to our problem. The fusion methods for ranking output classifiers were not fit to combine template matching methods. Since, template matching methods are two class systems. Ranking is not different than single class label output for a two class decision system. As mentioned before, template matching methods measure the similarity and compare the similarity measure with a threshold. According to the result of the comparison, they produce binary output. In order to use the classifier fusion methods for soft/fuzzy output classifiers, the similarity values were used. These similarity values are normalized to [0,1] range and used as soft output. The studied methods were simple Bayes average, Bayes belief integration and fuzzy template methods. In addition to them, a method which works on the soft outputs was proposed. This method was based on finding the optimal weights and threshold for combining the soft outputs of the classifiers. Proposed method gave successful results compared to fuzzy templates and simple Bayes average method. By using different optimization methods or modifying the cost function, the performance of the method may be improved. As classifier selection methods, static and dynamic classification methods were applied. In dynamic approach, determining the feature vector is the essential part of the work. In order to provide a feature vector, SIFT output and Canny edge detector were used. It is planned to extend the dynamic classifier selection by determining more descriptive and discriminative feature vectors. By that way, we can utilize the strong side of each template matching method. Classifier structuring and grouping methods are not applied in this work. Since, the number of template matching methods used was not high enough to group or cluster. By increasing the number of classifiers, these methods may also be used.

In this thesis, adaptation and application of classifier fusion methods to template matching results were studied. The results were analyzed in different perspective and compared by different evaluation methods. As a future work, increasing the used template matching methods and applying the fusion methods on a broader training set and on a different template matching methods are planned. The problem analyzed so far was two-class decision system. The performance of the proposed method on multiple class system will also be analyzed.

REFERENCES

[1] T. Mahalakshmi, R. Muthaiah and P. Swaminathan. (2012). Review Article: An Overview of Template Matching Technique in. *Research Journal of Applied Sciences, Engineering and Technology 4 , 4* (24), 5469-5473

[2] P. A. Devijver, J. Kittler. (1982). Pattern Recognition, A Statistical Approach. Prentice Hall.

[3] Olson, C. F. (2000). Maximum-Likelihood Template Matching. *IEEE , 1063* (6919).

[4] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. (1993). Comparing Images Using the Hausdorff Distance. *IEEE TRANSACTIONS ON PAmRN ANALYSIS AND MACHINE INTELLIGENCE , 15* (9).

[5] XIAO-WEI WANG, ZHONG WANG, JUN-TAO SUN, HUI-MIN ZHANG. (2005). THE CORRELATION TEMPLATE MATCHING ALGORITHM BASED TD FILTER AND ESO FILTER. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics.* Guangzhou.

[6] Kai Briechle and Uwe D. Hanebeck. (2001). Template Matching using Fast Normalized Cross Correlation. *Proceedings of SPIE , 4387*, 95-102.

[7] You Lin, Xiu Chunbo. (2011). Template Matching Based Edge Detection. *International Symposium on Computer Science and Society*.

[8] Chung-Chia Kang,Wen-JuneWang. (2007). A novel edge detection method based on the maximizing objective function. *Pattern Recognition , 40*, 609 – 618.

[9] B. Deutsch, Ch. Graß, F. Bajramovic, J. Denzler. (2005). A Comparative Evaluation of Template and Histogram Based 2D Tracking Algorithms. *27th DAGM Symposium.*

[10] Sidnei Alves de Araújo, Hae Yong Kim. (2010). Color-Ciratefi: A color-based RST-invariant template matching algorithm. *IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing.* São Paulo – Brazil.

[11] Jianzhong Fang and Guopirig Qiu. (2003). HUMAN FACE DETECTION USING ANGULAR RADIAL TRANSFORM AND SUPPORT. *IEEE , 7803* (7750), 669-672.

[12] Jan-Mark Geusebroek, Rein van den Boomgaard, Arnold W.M. Smeulders and Hugo Geerts. (2001). Color Invariance. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , 23* (12), 1338-1350.

[13] Hae Yong Kim and Sidnei Alves de Araújo. *Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast.* São Paulo-Brazil: Escola Politécnica, Universidade de São Paulo.

[14] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, Jiri Matas. (1998). On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence , 20* (3), 226-239.

[15] Fruitiere, S. B. (2013). July 2013 http://edgestracing.aurelie-fruitiere.fr/canny.html

[16] Miroslaw Bober, Francoise Preteux, YM Kim. (2001). *MPEG-7 Visual Shape Descriptors.* Guildford, United Kingdom: Mitsubishi Electric Information Technology Centre Europe.

[17] Tin Kam Ho, Jonathan J. Hull and Sargur N. Srihari. (1994). Decision Combination in Multiple Classifier Systems. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , 16* (1), 66-75.

[18] Dymitry Ruta and Bogdan Gabrys. (2000). An Overview of Classifier Fusion Methods. *Computing and Information Systems , 7*, 1-10.

[19] F.M. Alkoot, J. Kittler. (1999). Experimental evaluation of expert fusion strategies. *Pattern Recognition Letters , 20*, 1361-1369.

[20] L.I. Kuncheva and C.J. Whitaker. (2001). *TEN MEASURES OF DIVERSITY IN CLASSIFIER ENSEMBLES: LIMITS FOR TWO CLASSIFIERS.* London-UK: IEE Savoy Place.

[21] Olson, C. F. (2002). Maximum-Likelihood Image Matching. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , 24* (6).

[22] Dymitr Ruta, Bogdan Gabrys. *Analysis of the correlation between majority voting error and the diversity measures in multiple classifier systems.* Paisley, Scotland, United Kingdom: Applied Computational Intelligence Research Unit.

[23] Richa Singh, Mayank Vatsa, Afzel Noore, Sanjay K. Singh. *Dempster-Shafer Theory based Classifier Fusion for Improved Fingerprint Verification Performance.* Morgantown, USA: West Virginia University.

[24] Lowe, David G. (1999). Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision 2*, (s. 1150-1157).

[25] Li-ying Yang, Jun-ying Zhang. (2006). Improved Behavior Knowledge Space Combination Method with Observational Learning. *IMACS Multiconference on "Computational Engineering in Systems Applications (CESA).* Beijing, China.

[26] Narasimhamurty, A. (2005). Theoretical Bounds of Majority Voting Performance for a Binary Classification Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence , 27* (12), 1988-1995.

[27] Eric Bauer, Ron Kohavi. (1998). *An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants.* Boston-USA: Kluwer Academic Publishers.

[28] Freund, Y. (1996). Boosting a Weak Learning Algorithm by Majority. *Information and Computation , 121* (2), 256-285.

[29] Y.S. Huang and Y.C. Suen. (1993). *The Behavior Knowledge Space Method for Combination of Multiple Classifiers.* Quebec-Canada: Centre for Pattern Recognition and Machine Intelligence.

[30] F. Roli, S. Raudys, G. Marcialis. (tarih yok). An experimental comparison of fixed and trained fusion rules for crisp classifier outputs. *LNCS , 2364*, s. 232-241.

[31] Can Demirkesen, Hocine Cherifi. (2009). Fusing Image Representations for Classification Using Support Vector Machines. *24th International Conference Image and Vision Computing.* New Zealand.

[32] Ludmila I. Kuncheva, James C. Bezdek, Melanie A. Sutton. (1998). On Combining Multiple Classifiers by Fuzzy Templates. *IEEE* , 193-197.

[33] Kuncheva, L. I. (2001). Decision Templates for Multiple Classifier Fusion: An Experimental Comparison. *Pattern Recognition , 34* (2), 299-314.

[34] Kevin Woods, W. Philip Kegelmeyer Jr., Kewin Bowyer . (1997). Combination of Multiple Classifiers Using Local Accuracy Estimates. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , 19* (4), 405-410.

[35] Giorgio Giacinto and Fabio Roli. *Methods for Dynamic Classifier Selection.* Cagliari-Italy: Dept. of Electrical and Electronics eng., University of Cagliari.

[36] Giorgio Giacinto and Fabio Roli. (2000). A Theoretical Framework for Dynamic Classifier Selection. *IEEE* , 8-11.

[37] Marcilio C.P. de Souto, Rogrigo G.F. Soares, Alixandre Santana and Anne M.P. Canuto. (2008). Empirical Comparison of Dynamic Classifier Selection Methods based on Diversity and Accuracy for Building Ensembles. *International Joint Conference on Neural Networks*, (pp. 1480-1487).

[38] Kisku D.R., Gupta P., Sing J. K. (2009). Fusion of Multiple Matchers using SVM for Offline Signature Identification. *International Conference on Security Technology (SecTech).*

[39] Kalyan Veeramachaneni, Weizhong Yan, Kai Goebel, Lisa Osadciw. (2007). Improving Classifier Fusion Using Particle Swarm Optimization. *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, (s. 128-135).

[40] Kalyan Veeramachaneni, Lisa Osadciw, Arun Ross, Nisha Srinivas . (June 2008). Decision-level Fusion Strategies for Correlated Biometric Classifiers . *Proc. of IEEE Computer Society Workshop on Biometrics at the Computer Vision and Pattern Recogniton (CVPR) conference* .

[41] Chris Harris & Mike Stephens. (1988). *A COMBINED CORNER AND EDGE DETECTOR.* United Kingdom: The Plessey Company pic.

[42] Gertjan J. Burghouts, Jan-Mark Geusebroek. (2009). Performance evaluation of local colour invariants. *Computer Vision and Image Understanding , 113*, 48-62.

[43] Giorgio Giacinto and Fabio Roli. *Adaptive Selection of Image Classifiers.* Cagliari-Italy: Dept. of Electrical and Electronics Eng, University of Cagliari.

[44] J. Kittler and F.M. Alkoot. (2003). Sum versus Vote Fusion in Multiple Classifier Systems. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , 25* (1), 110-115.

[45] Kuncheva, L. I. (2002). A Theoretical Study on Six Classifier Fusion Strategies. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , 24* (2), 281-286.

[46] L.I. Kuncheva, C.J. Whitaker, C. Shipp and R. Duin. (2003). Limits on the Majority Vote Accuracy in Classifier Fusion. *Pattern Analysis and Applications , 6* (1), 22-31.

[47] Sangho Park, J.K. Aggarwal. (2004). A hierarchical Bayesian network for event recognition of human actions and interactions. *Multimedia Systems , 10*, 164-179.

[48] MATLAB Help

APPENDIX A

USED IMAGES AND TEMPLATES

In this Appendix, the images used as training and test and the templates in each image are given.

Figure 35: Image 1 and the template for this image

Figure 36: Image 2 and the template for this image

Figure 37: Image 3 and the template for this image

Figure 38: Image 4 and the template for this image

Figure 39: Image 5 and the template for this image

Figure 40: Image 6 and the template for this image

Figure 41: Image 7 and the template for this image

Figure 42: Image 8 and the template for this image

Figure 43: Image 9 and the template for this image

Figure 44: Image 10 and the template for this image

Figure 45: Image 11 and the template for this image

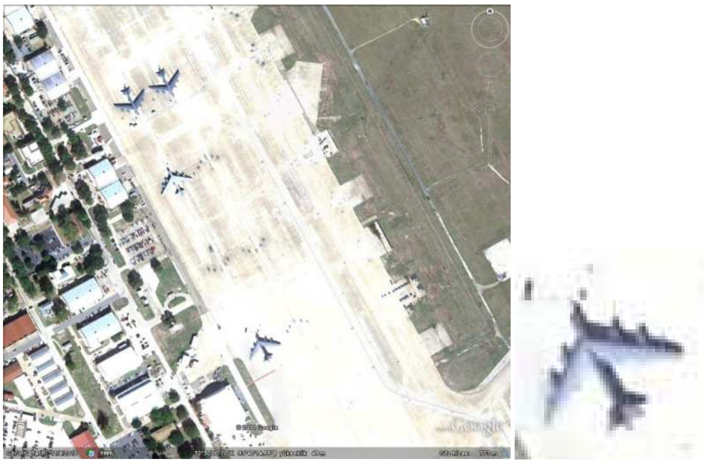


Figure 46: Image 12 and the template for this image

Figure 47: Image 13 and the template for this image

Figure 48: Image 14 and the template for this image

Figure 49: Image 15 and the template for this image

Figure 50: Image 16 and the template for this image

Figure 51: Image 17 and the template for this image

Figure 52: Image 18 and the template for this image

APPENDIX B

FMINCON ACTIVE SET ALGORITHM [48]

In this Appendix, fmincon Active Set Algorithm is given.

i. Introduction

Optimization techniques are used to find a set of design parameters, $x = \{x_1,x_1,...,x_n\}$, that can in some way be defined as optimal. In a simple case this might be the minimization or maximization of some system characteristic that is dependent on x. In a more advanced formulation the objective function, $f(x)$, to be minimized or maximized, might be subject to constraints in the form of equality constraints, $G_i(x) = 0$ ($i = 1,...,m_e$); inequality constraints, $G_i(x) \leq 0$ ($i = m_e + 1,...,m$); and/or parameter bounds, x_l, x_u.

A General Problem (GP) description is stated as

$$\begin{gathered} \min_x f(x), \\ subject\ to \\ G_i(x) = 0\ i = 1, \dots, m_e \\ G_i(x) \leq 0\ i = m_e + 1, \dots, m \end{gathered} \tag{B.1}$$

where x is the vector of length n design parameters, $f(x)$ is the objective function, which returns a scalar value, and the vector function $G(x)$ returns a vector of length m containing the values of the equality and inequality constraints evaluated at x.

In constrained optimization, the general aim is to transform the problem into an easier subproblem that can then be solved and used as the basis of an iterative process. A characteristic of a large class of early methods is the translation of the constrained problem to a basic unconstrained problem by using a penalty function for constraints that are near or beyond the constraint boundary. In this way the constrained problem is solved using a sequence of parameterized unconstrained optimizations, which in the limit (of the sequence) converge to the constrained problem. These methods are now considered relatively inefficient and have been replaced by methods that have focused on the solution of the Karush-Kuhn-Tucker (KKT) equations. The KKT equations are necessary conditions for optimality for a constrained optimization problem. If the problem is a so-called convex programming problem, that is, $f(x)$ and $G_i(x)$, $i = 1,...,m$, are convex functions, then the KKT equations are both necessary and sufficient for a global solution point.

Referring to general problem (B-1), the Kuhn-Tucker equations can be stated as in (B-2) in addition to the original constraints in *(B.1)*.

$$\begin{gathered} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i . \nabla G_i(x^*) = 0 \\ \lambda_i . \nabla G_i(x^*) = 0, i = 1, \ldots, m_e \\ \lambda_i \geq 0, i = m_e + 1, \ldots, m \end{gathered} \tag{B.2}$$

The first equation describes a canceling of the gradients between the objective function and the active constraints at the solution point. For the gradients to be canceled, Lagrange multipliers (λ_i, $i = 1,...,m$) are necessary to balance the deviations in magnitude of the objective function and constraint gradients. Because only active constraints are included in this canceling operation, constraints that are not active must not be included in this operation and so are given Lagrange multipliers equal to 0. This is stated implicitly in the last two Kuhn-Tucker equations.

The solution of the KKT equations forms the basis to many nonlinear programming algorithms. These algorithms attempt to compute the Lagrange multipliers directly. Constrained quasi-Newton methods guarantee superlinear convergence by accumulating second-order information regarding the KKT equations using a quasi-Newton updating procedure. These methods are commonly referred to as Sequential Quadratic Programming (SQP) methods, since a QP subproblem is solved at each major iteration (also known as Iterative Quadratic Programming, Recursive Quadratic Programming, and Constrained Variable Metric methods).

ii. Sequential Quadratic Programming (SQP)

Given the problem description in GP (B.1) the principal idea is the formulation of a QP subproblem based on a quadratic approximation of the Lagrangian function.

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i . g_i(x) \tag{B.3}$$

Here you simplify *(B.1)* by assuming that bound constraints have been expressed as inequality constraints. You obtain the QP subproblem by linearizing the nonlinear constraints.

iii. Quadratic Programming (QP) Subproblem

$$\min_{d \in R^n} \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d$$
$$\nabla g_i(x_k)^T d + g_i(x_k) = 0, \; i = 1, \ldots, m_e \qquad \textbf{\textit{(B.4)}}$$
$$\nabla g_i(x_k)^T d + g_i(x_k) \leq 0, \; i = m_e + 1, \ldots, m$$

This subproblem can be solved using any QP algorithm. The solution is used to form a new iterate:

$$x_{k+1} = x_k + \alpha_k d_k \qquad \textbf{\textit{(B.5)}}$$

The step length parameter α_k is determined by an appropriate line search procedure so that a sufficient decrease in a merit function is obtained. The matrix H_k is a positive definite approximation of the Hessian matrix of the Lagrangian function *(B.3)*. H_k can be updated by any of the quasi-Newton methods, although the BFGS method appears to be the most popular.

A nonlinearly constrained problem can often be solved in fewer iterations than an unconstrained problem using SQP. One of the reasons for this is that, because of limits on the feasible area, the optimizer can make informed decisions regarding directions of search and step length.

iv. SQP Implementation

The SQP implementation consists of three main stages, which are discussed briefly in the following subsections:

- Updating the Hessian Matrix
- Quadratic Programming Solution
- Line Search and Merit Function

At each major iteration a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, H, is calculated using the BFGS method, where λ_i, i = 1,...,m, is an estimate of the Lagrange multipliers.

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T s_k^T s_k H_k}{s_k^T H_k s_k}$$
$$where$$
$$s_k = x_{k+1} - x_k$$
$$q_k = (\nabla f(x_{k+1}) + \textstyle\sum_{i=1}^{m} \lambda_i . g_i(x_{k+1})) - (\nabla f(x_{k+1}) + \sum_{i=1}^{m} \lambda_i . g_i(x_{k+1}))$$

(B.6)

A positive definite Hessian is maintained providing $q_k^T s_k$ is positive at each update and that H is initialized with a positive definite matrix. When $q_k^T s_k$ is not positive, q_k is modified on an element-by-element basis so that $q_k^T s_k > 0$. The general aim of this modification is to distort the elements of q_k, which contribute to a positive definite update, as little as possible. Therefore, in the initial phase of the modification, the most negative element of q_k*s_k is repeatedly halved. This procedure is continued until is $q_k^T s_k$ greater than or equal to a small negative tolerance. If, after this procedure, $q_k^T s_k$is still not positive, modify q_k by adding a vector *v* multiplied by a constant scalar *w*, that is,

$$
\begin{gathered}
q_k = q_k + wv \\
where \\
v_i = \nabla g_i(x_{k+1}).\, g_i(x_{k+1}) - \nabla g_i(x_k).\, g_i(x_k) \; if \; (q_k)_i . w < 0 \; and \; (q_k)_i . (s_k)_i < 0, i = 1, \dots, m \\
v_i = 0, otherwise
\end{gathered}
\qquad \textbf{\textit{(B.7)}}
$$

and increase *w* systematically until $q_k^T s_k$ becomes positive.

The functions `fmincon`, `fminimax`, `fgoalattain`, and `fseminf` all use SQP. If `Display` is set to `'iter'` in `options`, then various information is given such as function values and the maximum constraint violation. When the Hessian has to be modified using the first phase of the preceding procedure to keep it positive definite, then `Hessian modified` is displayed. If the Hessian has to be modified again using the second phase of the approach described above, then `Hessian modified twice` is displayed. When the QP subproblem is infeasible, then `infeasible` is displayed. Such displays are usually not a cause for concern but indicate that the problem is highly nonlinear and that convergence might take longer than usual. Sometimes the message `no update` is displayed, indicating that $q_k^T s_k$ is nearly zero. This can be an indication that the problem setup is wrong or you are trying to minimize a noncontinuous function.

At each major iteration of the SQP method, a QP problem of the following form is solved, where A_i refers to the `i`th row of the *m*-by-*n* matrix *A*.

$$
\begin{gathered}
\min_{d \in R^n} q\,(d) = \frac{1}{2} d^T H d + c^T d, \\
A_i d = b_i, i = 1, \dots, m_e, \\
A_i d \le b_i, i = m_e + 1, \dots, m
\end{gathered}
\qquad \textbf{\textit{(B.7)}}
$$

The method used in Optimization Toolbox functions is an active set strategy (also known as a projection method). It has been modified for both Linear Programming (LP) and Quadratic Programming (QP) problems.

The solution procedure involves two phases. The first phase involves the calculation of a feasible point (if one exists). The second phase involves the generation of an iterative sequence of feasible points that converge to the solution. In this method an active set $\overline{A_k}$, is maintained that is an estimate of the active constraints (i.e., those that are on the constraint boundaries) at the solution point. Virtually all QP algorithms are active set methods. This point is emphasized because there exist many different methods that are very similar in structure but that are described in widely different terms. $\overline{A_k}$ is updated at each iteration *k*, and this is used to form a basis for a search direction $\widehat{d_k}$. Equality constraints always remain in the active set $\overline{A_k}$. The notation for the variable $\widehat{d_k}$ is used here to distinguish it from d_k in the major iterations of the SQP method. The search direction d_k is calculated and minimizes the objective function while remaining on any active constraint boundaries. The feasible subspace for d_k is formed from a basis Z_k whose columns are orthogonal to the estimate of the active set $\overline{A_k}$(i.e., $\overline{A_k}Z_k = 0$). Thus a search direction, which is formed from a linear summation of any combination of the columns of Z_k, is guaranteed to remain on the boundaries of the active constraints.

The matrix Z_k is formed from the last $m - l$ columns of the QR decomposition of the matrix $\overline{A_k}^T$, where l is the number of active constraints and $l < m$. That is, Z_k is given by

$$Z_k = Q[:, l+1:m], \quad where \quad Q^T\overline{A_k}^T = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{B.8}$$

Once Z_k is found, a new search direction $\widehat{d_k}$ is sought that minimizes $q(d)$ where $\widehat{d_k}$ is in the null space of the active constraints. That is, $\widehat{d_k}$ is a linear combination of the columns of Z_k: $\widehat{d_k}=Z_k p$ for some vector p.

Then if you view the quadratic as a function of p, by substituting for $\widehat{d_k}$, you have

$$q(p) = \frac{1}{2}p_T Z_k^T H Z_k p + c^T Z_k p \tag{B.9}$$

Differentiating this with respect to p yields

$$\nabla q(p) = Z_k^T H Z_k p + Z_k^T c \tag{B.10}$$

$\nabla q(p)$ is referred to as the projected gradient of the quadratic function because it is the gradient projected in the subspace defined by Z_k. The term $Z_k^T H Z_k$ is called the projected Hessian. Assuming the Hessian matrix H is positive definite (which is the case in this implementation of SQP), then the minimum of the function $q(p)$ in the subspace defined by Z_k occurs when $\nabla q(p) = 0$, which is the solution of the system of linear equations

$$Z_k^T H Z_k p = -Z_k^T c \tag{B.11}$$

A step is then taken of the form

$$x_{k+1} = x_k + \alpha\widehat{d_k}, where\ \widehat{d_k} = Z_k^T c \tag{B.12}$$

At each iteration, because of the quadratic nature of the objective function, there are only two choices of step length α. A step of unity along $\widehat{d_k}$ is the exact step to the minimum of the function restricted to the null space of $\overline{A_k}$. If such a step can be taken, without violation of the constraints, then this is the solution to QP *(B.8)*.

Otherwise, the step along $\widehat{d_k}$ to the nearest constraint is less than unity and a new constraint is included in the active set at the next iteration. The distance to the constraint boundaries in any direction $\widehat{d_k}$ is given by

$$\alpha = \min_{i\in\{1,\dots,m\}}\left\{\frac{-(A_i x_k - b_i)}{A_i d_k}\right\} \qquad \textbf{\textit{(B.13)}}$$

which is defined for constraints not in the active set, and where the direction $\widehat{d_k}$ is towards the constraint boundary, i.e., $A_i d_k$>0, i=1, …, m.

When *n* independent constraints are included in the active set, without location of the minimum, Lagrange multipliers, λ_k, are calculated that satisfy the nonsingular set of linear equations

$$\overline{A_k}^T \lambda_k = c \qquad \textbf{\textit{(B.14)}}$$

If all elements of λ_k are positive, x_k is the optimal solution of QP *(B.8)*. However, if any component of λ_k is negative, and the component does not correspond to an equality constraint, then the corresponding element is deleted from the active set and a new iterate is sought.

v. Initialization:

The algorithm requires a feasible point to start. If the current point from the SQP method is not feasible, then you can find a point by solving the linear programming problem

$$\begin{gathered}\min_{\gamma\in R, x\in R^n} \gamma\,, such\ that \\ A_i x = b_i, i = 1, \dots, m_e \\ A_i x - \gamma \le b_i, i = m_e + 1, \dots, m\end{gathered} \qquad \textbf{\textit{(B.15)}}$$

The notation A_i indicates the *i*th row of the matrix *A*. You can find a feasible point (if one exists) to *(B.15)* by setting *x* to a value that satisfies the equality constraints. You can determine this value by solving an under- or overdetermined set of linear equations formed from the set of equality constraints. If there is a solution to this problem, then the slack variable γ is set to the maximum inequality constraint at this point.

You can modify the preceding QP algorithm for LP problems by setting the search direction to the steepest descent direction at each iteration, where g_k is the gradient of the objective function (equal to the coefficients of the linear objective function).

$$\widehat{d_k} = -Z_k Z_k^T g_k \qquad \textbf{\textit{(B.16)}}$$

If a feasible point is found using the preceding LP method, the main QP phase is entered. The search direction $\widehat{d_k}$ is initialized with a search direction $\widehat{d_1}$ found from solving the set of linear equations

$$H\widehat{d_1} = -g_k \tag{B.17}$$

where g_k is the gradient of the objective function at the current iterate x_k (i.e., $Hx_k + c$).

If a feasible solution is not found for the QP problem, the direction of search for the main SQP routine $\widehat{d_k}$ is taken as one that minimizes γ.

vi. Line Search and Merit Function:

The solution to the QP subproblem produces a vector d_k, which is used to form a new iterate

$$x_{k+1} = x_k + \alpha d_k \tag{B.18}$$

The step length parameter α_k is determined in order to produce a sufficient decrease in a merit function. The following form is used in this implementation.

$$\Psi(x) = f(x) + \sum_{i=1}^{m_e} r_i . g_i(x) + \sum_{i=m_e+1}^{m} r_i . \max\,[0, g_i(x)] \tag{B.19}$$

The penalty function is set as given in *(B.20).*

$$r_i = (r_{k+1})_i = \max_i \left\{\lambda_i, \frac{(r_k)_i + \lambda_i}{2}\right\}, i = 1, \dots, m \tag{B.20}$$

This allows positive contribution from constraints that are inactive in the QP solution but were recently active. In this implementation, the penalty parameter r_i is initially set to

$$r_i = \frac{\|\nabla f(x)\|}{\|g_i(x)\|}, \tag{B.21}$$

where $\| \quad \|$ represents the Euclidean norm.

This ensures larger contributions to the penalty parameter from constraints with smaller gradients, which would be the case for active constraints at the solution point.

APPENDIX C

OUTPUTS OF ALGORITHMS FOR AN EXAMPLE IMAGE

In this Appendix, the outputs of template matching methods and classifier fusion methods for the test image and template image given below:

Figure 53: Example test image and the template of it

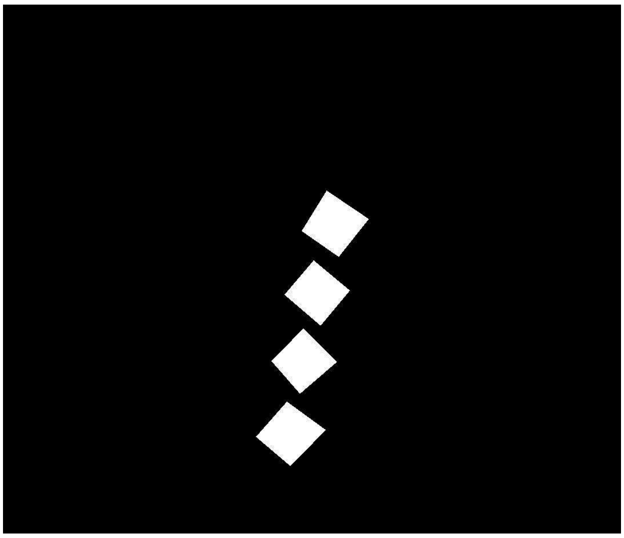


Figure 54: Ground truth of the example test image

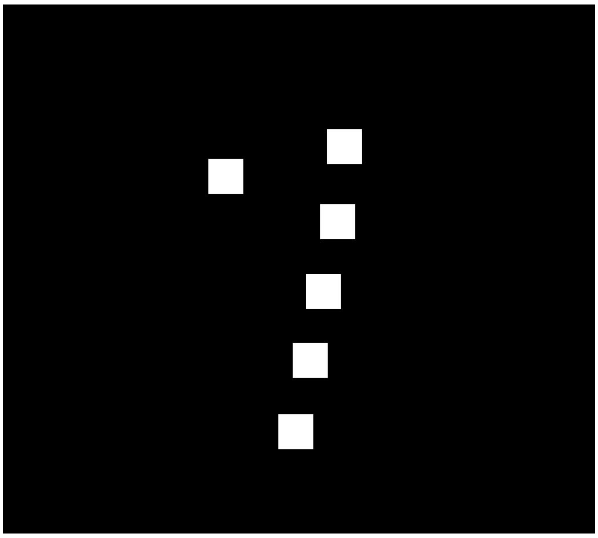


Figure 55: Correlation Based Template Matching Method’s Output for the Example Test Image and Template

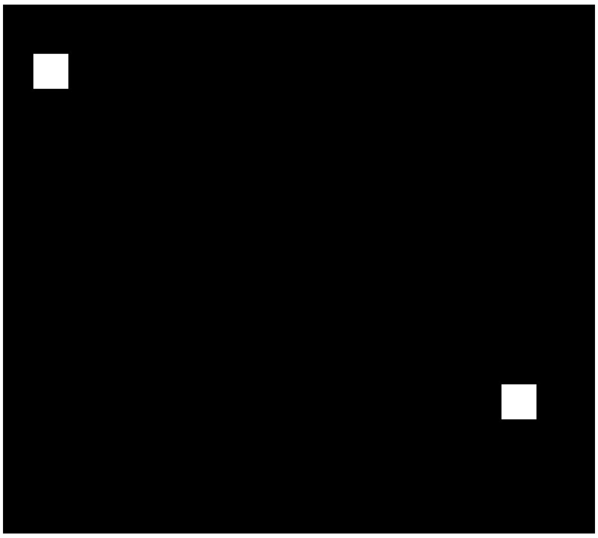


Figure 56: Edge Based Template Matching Method’s Output for the Example Test Image and Template

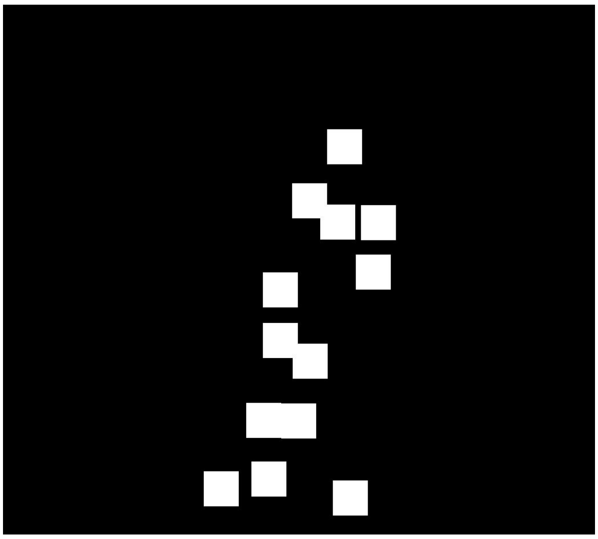


Figure 57: Histogram Based Template Matching Method’s Output for the Example Test Image and Template

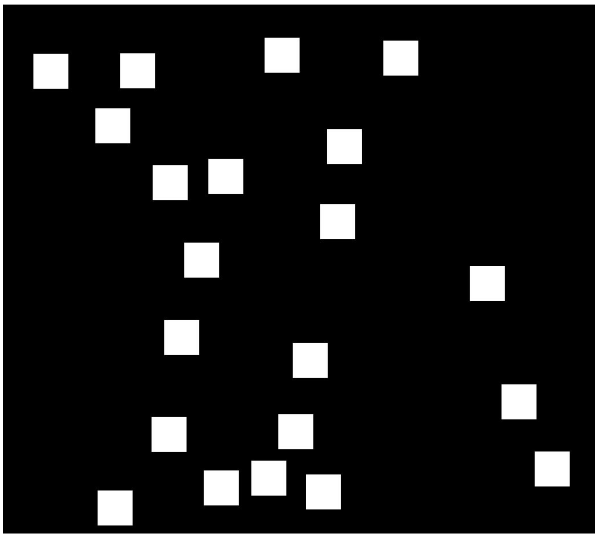


Figure 58: Angular Radial Transform Template Matching Method's Output for the Example Test Image and Template

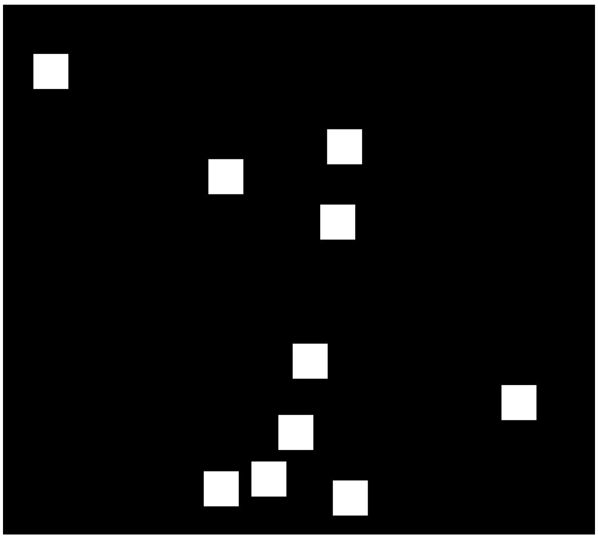


Figure 59: Majority Voting Method's Output for the Example Test Image and Template

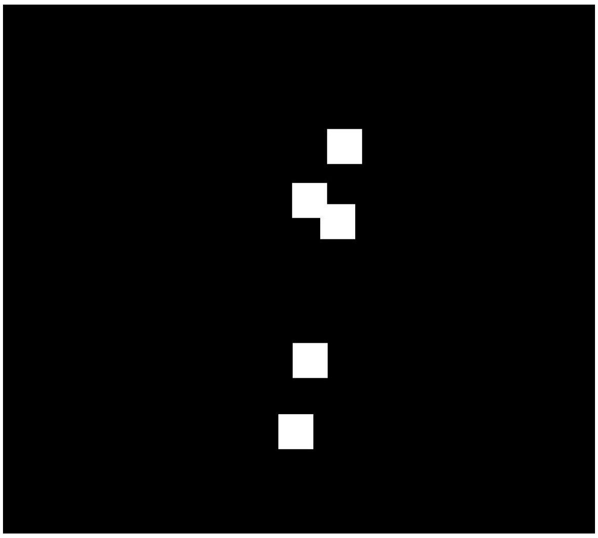


Figure 60: Behavior Knowledge Space Method’s Output for the Example Test Image and Template

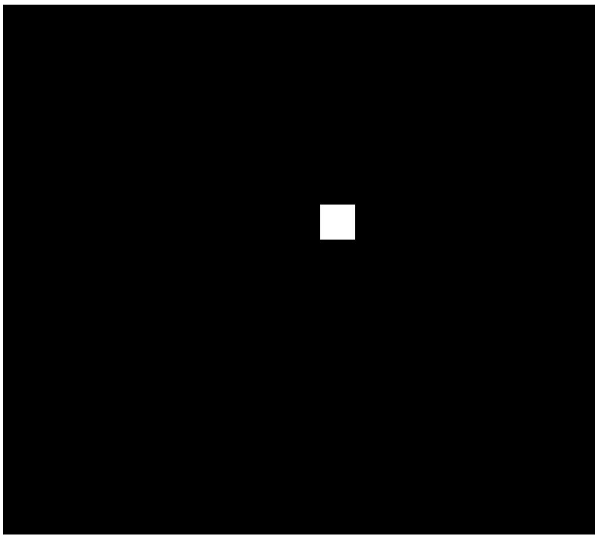


Figure 61: Simple Bayes Average Method’s Output for the Example Test Image and Template

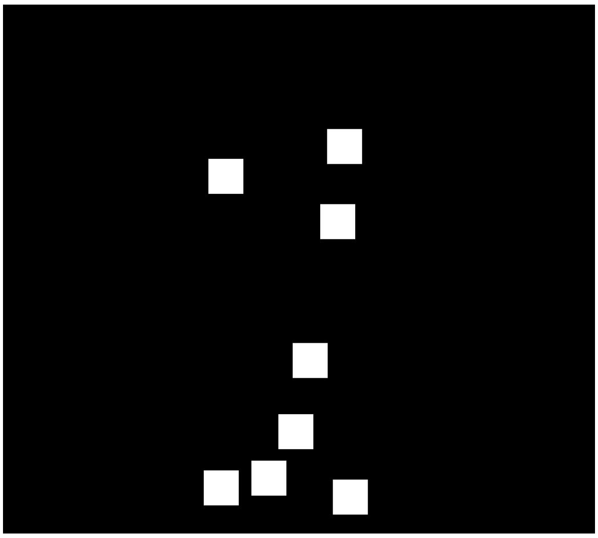


Figure 62: Bayes Belief Integration Method's Output for the Example Test Image and Template

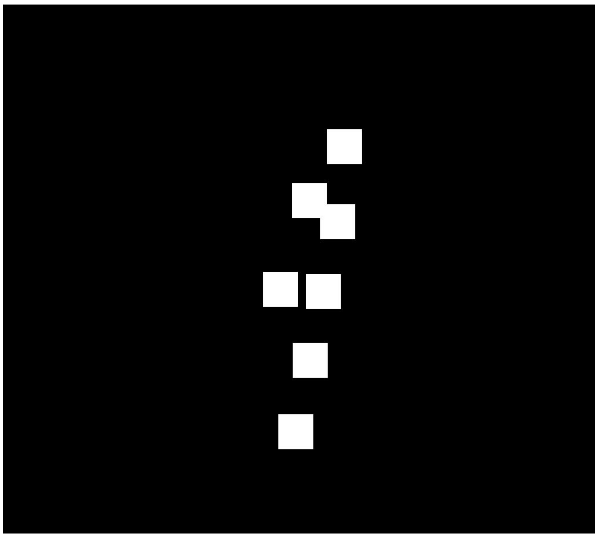


Figure 63: Fuzzy Templates Method's Output for the Example Test Image and Template

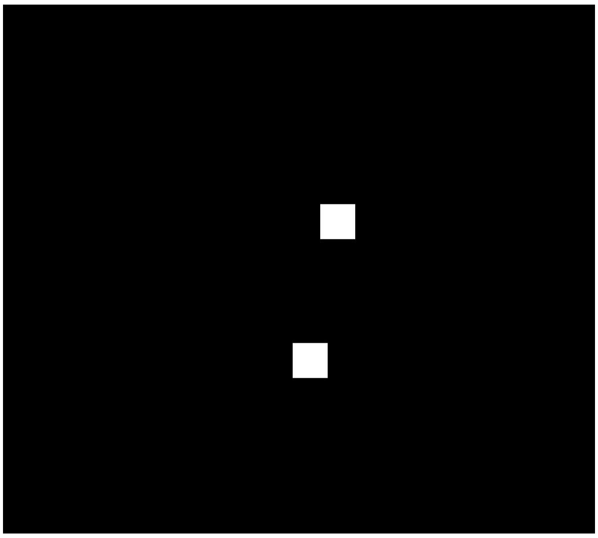


Figure 64: Optimization Based Template Matching Method's Output for the Example Test Image and Template

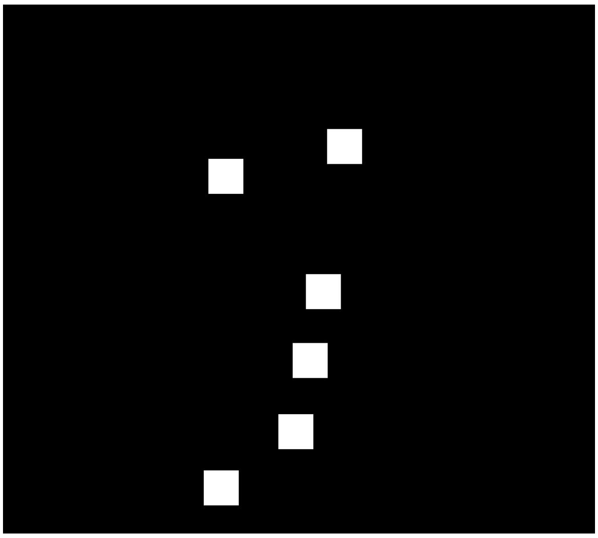


Figure 65: Static Classifier Selection Method's Output for the Example Test Image and Template

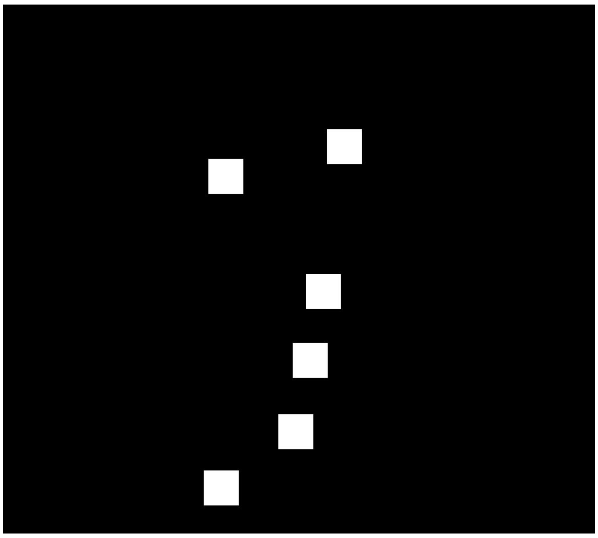


Figure 66: Dynamic Classifier Selection by Edge Intensity Method's Output for the Example Test Image and Template

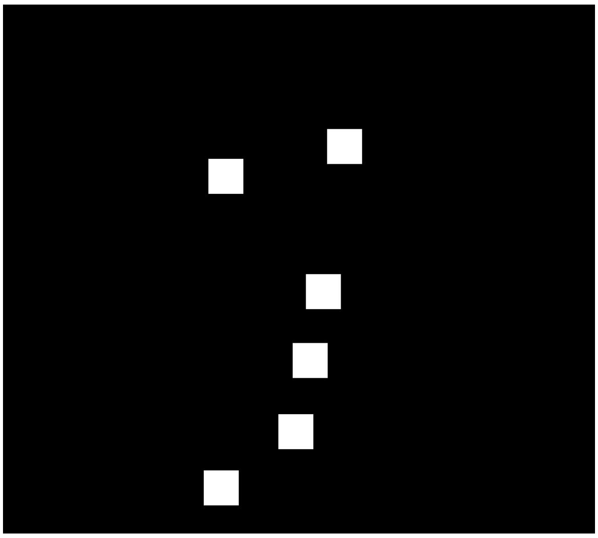


Figure 67: Dynamic Classifier Selection by SIFT Descriptor Method's Output for the Example Test Image and Template