SELF-SUPERVISED BUILDING DETECTION WITH DECISION FUSION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞLAR ŞENARAS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
PHILOSOPHY OF DOCTORATE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2013

SELF-SUPERVISED BUILDING DETECTION WITH DECISION FUSION

Submitted by **Çağlar Şenaras** in partial fulfillment of the requirements for the degree of **Philosophy of Doctorate**, in **Information Systems, Middle East Technical University** by,

Prof.Dr. Nazife Baykal _____
Director, **Informatics Institute**

Prof. Dr. Yasemin Yardımcı Çetin _____
Head of Department, **Information Systems**

Prof. Dr. Fatoş T. Yarman Vural _____
Supervisor, **Computer Engineering Systems, METU**

Assist. Prof. Dr. Erhan Eren _____
Co-Supervisor, **Information Systems, METU**

Examining Committee Members

Prof. Dr. Yasemin Yardımcı Çetin _____
Information Systems, METU

Prof. Dr. Fatoş T. Yarman Vural _____
Computer Engineering, METU

Dr. Erkut Erdem _____
Computer Engineering, Hacettepe University

Assoc. Prof. Dr. Altan Koçyiğit _____
Information Systems, METU

Assist. Prof. Dr. Alptekin Temizel _____
Work Based Learning, METU

Date: _____

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, Last Name: ÇAĞLAR ŞENARAS

Signature :

ABSTRACT

SELF-SUPERVISED BUILDING DETECTION WITH DECISION FUSION

Şenaras, Çağlar
Ph.D., Department of Information Systems
Supervisor : Prof. Dr. Fatoş T. Yarman Vural
Co-Supervisor : Assist. Prof. Dr. Erhan Eren

September 2013, 97 pages

This thesis proposes a new building detection framework for monocular satellite images, called Self-Supervised Decision Fusion (SSDF). The model is based on the idea of self-supervision, which aims to generate training data automatically from each individual test image, without any human interaction. This principle allows us to use the advantages of the supervised classifiers in a fully automated framework. The technical shortcomings of the available supervised and unsupervised algorithms, such as difficulties in manual labeling of the images to extract the training data, large interclass variances and a wide variety of buildings, prevent the previous studies to satisfy the need of robust autonomous detection systems. We attempt to overcome these problems by combining our previous supervised and unsupervised building detection frameworks to suggest a self-supervised learning architecture. We borrow the major strength of the unsupervised approaches in order to obtain one of the most important clues, the relation of a building and its cast shadow in order to solve the major problem of training of the supervised approaches. Furthermore, supervised study allows us to combine the detection results of multiple classifiers under a hierarchical architecture, called Fuzzy Stacked Generalization (FSG).

The suggested method involves three major steps: In the first step, after pan-sharpening and segmentation process several masks are extracted to represent the invariant information about the building object. These masks are vegetation, shadow and rectangular structure masks. In the second step, by employing these masks negative and positive samples are selected from each image layout. Finally, the training data extracted in the second step is used to train FSG.

Keywords: building detection, self-supervision, decision fusion, remote sensing

ÖZ

KENDINDEN DENETIMLI KARAR FÜZYONU ILE BINALARIN TESPITI

Şenaras, Çağlar
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş T. Yarman Vural
Ortak Tez Yöneticisi : Yar. Doç. Dr. Erhan Eren

Eylül 2013, 97 sayfa

Bu tez çalışmasında monoküler uydu görüntüleri için otomatik bina tespiti yapan Kendinden Denetimli Karar Füzyonu (KDKF) isimli bir çatı önerilmiştir. Bu model, her bir resim için eğitim kümesini insan etkileşimi olmadan otomatik üretmeyi hedef alan, kendinden denetim fikri üzerine kurulmuştur. Bu yaklaşım, denetimli sınıflandırıcıların avantajlarından tam anlamıyla otomatik bir şekilde faydalanma imkânı sağlamaktadır. Elle görüntüleri etiketlemenin zorluğu, diğer objelerin binalara benzeyebilmesi ve binaların çok farklı yapılarda olabilmesi mevcut denetimli ve denetimsiz çalışmaların otomatik tespit ihtiyacını gerçek anlamda karşılamasını engellemiştir. Bu zorlukları aşmak için daha önce yapmış olduğumuz denetimli ve denetimsiz bina tespiti çalışmalarımızı birleştirerek yeni bir kendinden denetimli öğrenme mimarisi önermekteyiz. Denetimsiz çalışmamızın binaların bulunması için oldukça önemli bir ipucu olan bina ile gölgesi arasındaki ilişkiyi kullanabilme yeteneğini alıp, bunu denetimli yaklaşımlar için çok önemli olan eğitim verisinin oluşturulması probleminde kullanmaktayız. Denetimli çalışma ise bize farklı sınıflandırıcıların sonuçlarını, Bulanık Yığılmış Genelleme (BYG) adındaki sıradüzensel bir mimaride birleştirme imkânı sağlamaktadır.

Önerilen metot üç adım içermektedir: ilk adımda, pan-keskinleştirme ve bölütleme sonrası yeşil alan, gölge ve dikdörtgen alanları içeren maskeler çıkartılır. Sonraki adımda bu maskeler kullanılarak her bir resim için pozitif ve negatif örnekler seçilir. Son adımda, çıkartılan bu örneklerden oluşan eğitim verisi BYG'de kullanılır.

Önerilen bina bulma algoritması 18 görüntüde test edilmiştir. Bu deneyler sonucunda KDKF'nin son teknoloji ürünü denetimli ve denetimsiz yaklaşımlardan daha yüksek bir performans verildiği gözlemlemekteyiz KDKF'nin performansı çeşitli faktörlere bağlıdır. Öncelikle, pozitif örneklerin doğruluğunun algoritmanın performansı üzerinde önemli bir etkisi vardır. Yaptığımız testlerde pozitif örneklerin doğruluğunun az olduğu durumlarda KDKF algoritması sonuçlarında kesinlik değerinin de düşük olduğunu gözlemlenmiştir. Bununla birlikte doğru seçilmiş olan negatif örneklerinin sayısının da algoritma performansı üzerinde olumlu bir etkisi olduğu gözlemlenmiştir.

Anahtar Kelimeler: bina tespiti, kendinden denetimli, uzaktan algılama, karar füzyonu, uydu görüntüleri

dedicated to my wife...

ACKNOWLEDGMENTS

I would like to express my deepest grateful to my thesis advisor, Prof. Dr. Fatoş T. Yarman Vural, for her guidance, support and encouragement during my thesis. She is one of the most important people in my life and changed my point of view on life. I hope that, someday I could be as enthusiastic, positive and wise as her.

Thanks to my Co-Supervisor Assist. Prof. Dr. Erhan Eren and my thesis committee Assist. Prof. Dr. Alptekin Temizel and Assoc. Prof. Dr. Altan Koçyiğit for all the meetings they guide and support my progress in the thesis.

I thank to all my friends in METU Image lab, specially, Dr. Özge Öztimur Karadağ, Dr. Gulsah Tumuklu, Dr. Sarper Alkan, Orhan Firat, Ruşen Aktaş and Okan Akalin for their support and comments during my studies. Also, I would like to thank to Dr. Mete Ozay and Dr. Ali Özgün Ok for their friendship and collaboration.

I thank to all my officemates and colleagues in HAVELSAN, especially, Dr. Emre Başeksi, Dr. Selçuk Sümengen, Dr. Kartal Tabak and Dr. İsmail Cenik.

My parents deserve spacial thanks for their kindness, patient, love and endless support through the years.

Finally, I would like to give my very special thanks to my wife Eda for her love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Recently, remote sensing images are widely used in a diverse set of applications, such as environmental control, traffic monitoring, city planning, military surveillances, agriculture, earthquake analysis etc. Availability of the high resolution images, the advantages of the multispectral satellites and also the reduced image prices increase the popularity of this technology for many disciplines. Most of the application areas make use of the power of Geographic Information System (GIS) tools, which allows us to store spatiotemporal coordinates, such as location of highways and buildings at a specific time instant. Then, by employing this spatiotemporal data, it is possible to extract domain specific information, such as relative location of a certain address with respect to a moving vehicle. However, some of the applications may require further information to be extracted from the geospatial data. Examples include, but not limited to the recognition of the natural and man-made objects in a scene or detection of changes between two images of the same scene, taken in different time instances. For decades, the extraction of such complex information from the geospatial data has been done manually which is a very slow, tedious, erroneous and expensive operation. For most of the objects, the automation of detection, recognition or localization is still an open research area. The automatic detection of the buildings in satellite imagery is one of these challenging problems and also a hot research topic for the computer vision and the remote sensing communities. Object recognition in an image database is a very challenging problem in computer science and it can only be partially solved by employing the domain specific approaches. One of the most difficult application domains is the remote sensing imagery, where the data sets have diverse variations with

respect to the objects to be recognized and highly cluttered backgrounds. In other words, the complexity of the objects with high inter-class similarity and variance in a class complicates the classification problem in a remotely sensed scene. Also, a remote sensing image contains large amount of data and unpredictable number of different classes. The characteristics of the buildings in an image may be relatively similar to each other compared to more complex objects, such as airports, sports complexes, parking lots etc. However, buildings in two distinct image scenes show significant differences. Even when, we compare two images of the same building, which are acquired at different dates, they might be distinctly different due to the illumination, atmospheric and climatic changes. Moreover, the physical properties of the buildings may show a great variance due to the functional usage, architectural and cultural approaches to the construction.

1.1 Machine Learning for Building Detection Problem

Machine Learning Methodologies are immensely employed in most of the remote sensing areas, as well as in the building detection problem, under supervised and unsupervised learning paradigms. Most of the supervised building detection algorithms are based on extracting the training data from a region with "similar" statistical and structural properties of the target object in the test and training images. This approach has a very crucial assumption on the similarity of the target objects, which require heavy heuristics in the training phase. If the training and test data are selected from "dissimilar" and/or irrelevant regions (where data and domain shift occurs), the performance values decrease drastically, due to the change in the terrain structures and building variations. In practice, this challenge occurs if the buildings in training and test images have different sizes and color/shading properties. Similar problems are addressed in various remote sensing applications such as land use classification in [3], [4]. On the other hand, unsupervised techniques do not require this unrealistic assumption about the training phase. Therefore, it seems to be more suitable to the object detection problem in remote sensing. However, the unsupervised techniques are not capable of extracting the target specific information from the image data set, specifically when the image comes from a cluttered scene with a variety of other

objects similar to the target object(s).

1.2 Self-Supervision Paradigm for Object Detection in Remote Sensing

In this thesis, our motivation is to develop a self-supervised automatic building detection framework, which eliminates the drawbacks of both unsupervised and supervised approaches. The suggested framework employs one of the most important clues, the relation of a building and its cast shadow in order to generate its own training samples and eliminate the need of manual training generation for the supervised approaches. This elimination allows us to employ the most powerful supervised classifiers to the building detection problem. In order to improve the performances of a set of individual classifiers, we prefer to use a state of the art solution, which combines the detection results of multiple classifiers under a hierarchical architecture, called Fuzzy Stacked Generalization (FSG).

This Ph.D. Dissertation is organized as follows: After this brief introduction, we provide background information on building detection problem in the next chapter. The technical analysis prior to our development, together with our motivation is given in Chapter 3. In Chapter 4, we introduce the suggested self-supervised building detection method, Chapter 5, provides experimental studies performed on the suggested method. Finally, Chapter 6 concludes our study, remarking the superiorities and weaknesses of the suggested method, together with our future research direction. Let us now emphasize the major assumptions and contribution of this study:

1. The major assumption of the suggested method is that an image layout, obtained from the satellite, is self-consistent and has sufficient statistics about the target object. In other words, the regions in an image layout corresponding to the building objects are represented as random variables, which are independent and identically distributed. These layouts are also large enough to carry statistically sufficient positive and negative exemplars.

2. One of the major contributions of this study is to process each image layout independent of the rest of the images in the database. The suggested Self Supervised

Fusion Method (SSFM) receives each individual statistically stable image from the data set, extracts the training data and uses this data to detect the rest of the buildings independent of the rest of the images in the dataset.

3. Another contribution of this study is to extract the training data from each image automatically, using an unsupervised learning paradigm. Each image layout is considered as an independent input to a classifier in the training and test phases. Therefore, each image is self-trained by extracting the building regions with high confidentiality to detect the rest of the buildings based on this self-extracted training data. This task is achieved by finding the building and cast shadow pairs. The suggested unsupervised approach provides us with a statistically stable and reliable training data. It also, eliminates the data shift problem of the classical supervised learning methods.

4. Employing an ensemble learning architecture, called Fuzzy Stacked Generalization, enables us to learn more than one characteristic feature of the buildings. The complementary information coming from the multiple feature space is used to boost the performance the meta-layer of the classifier.

We hope that employing a self-supervised learning paradigm to building detection problem together with a supervised decision fusion framework will bring a new perspective for building detection literature. Our hope is accentuated by observing and validating the robustness of the suggested algorithms, during the experiments performed in various complex scenes.

CHAPTER 2

AN OVERVIEW AND BACKGROUND ON BUILDING DETECTION STUDIES

In this chapter, we provide a brief overview about the state-of-the art building detection research. It is well-known that this problem, is one of the most studied areas in the bulk of object detection methods in remote sensing domain. Since our contribution in this domain is in the pattern recognition side, we focus on the literature from this perspective and overview the material in terms of the types of learning methodologies.

In the following sections, we first give a brief literature survey on building detection under the headings of unsupervised and supervised methods. Then, we supply the material for the pre-processing technologies, employed in this study. Finally, we provide the metrics for performance evaluation of the suggested building detection technique.

2.1 An Overview of Methods for Building Detection Problem

Building detection is one of the challenging problems of target detection in remote sensing applications. The studies about the building detection problem started in the late 80's. However, the problem is still preserving its popularity. Majority of the well-known studies, developed in the last two decades, are summarized in survey studies. Mayer presented one of the important survey reports [5] at the end of 90's. In this study, he analyzed previous approaches according to the characterization of the models, their complexity and strategies. In 2004, Baltsavias prepared a review report focused on aerial images for building detection and road detection algorithms

[6]. In that study, he mentioned the importance of existing knowledge that can be used for object extraction (e.g. use of time and coordinate information for shadow analysis). In 2009, Koc San introduced a detailed survey for building detection in her Ph.D thesis [7]. Different from Mayer and Baltsavias, she grouped the previous works based on the used sensor type, such as aerial images, space imagery, lidar/DEM and multi-sensor technologies. Finally, Ünsalan and Boyer reviewed the building and road detection literature in 2011 [8]. They preferred to follow Mayer's report format and additionally reviewed recent studies.

The state of the art methods approach this problem from many different points of views. A common approach is to use multiple sensors for data fusion in different levels, which are pixel level, feature level and decision level using hierarchical algorithms [9]. The goal of these methods is to extract information from different data sources and feature spaces, which cannot be obtained from single data source or feature space, using multiple classifiers or decision systems [10]. Li *et al.* [11] aggregate LIDAR data and orthoimage for building reconstruction. Rottensteiner *et al.* [12] fuse laser scanner data and multi-spectral images for building detection. The class conditional probabilities of the pixels are calculated by using initial land cover classification. Next, the classification results are post-processed and combined using Dempster-Shafer theory. Li *et al.* [13] fuse spectral and texture features for the detection of collapsed buildings, damaged in the earthquakes. Hansch and Hellwich [14] use Random Forests to combine polarimetry, intensity and texture features extracted from Polarimetric Synthetic Aperture Radar (SAR) images for building detection.

On the other hand, the approaches based on the monocular optical images are still very popular due to the wide availability of the source data. Since this thesis is devoted to the detection of buildings from a single optical image, we limit the rest of the literature survey and discuss only the previous studies aimed to automatically detect and extract buildings from monocular optical images. The state of the art methods in this field show a great variety of solutions. However, these methods can be categorized into two major headings, namely unsupervised and supervised approaches. The unsupervised algorithms basically, detect the buildings using predefined rule-based models and unsupervised classifiers. On the other hand, the studies, which are categorized as supervised approaches, use the prior information given by a supervisor. In the

following two sections, we summarize the literature according to this categorization.

2.1.1 Unsupervised Approaches for Building Detection

A significant percentage of the studies in the literature can be categorized as unsupervised. The studies, which use predefined information to generate hypothesis; including rule based models, and also the studies based on an unsupervised classifier can be grouped in this category. Furthermore, most of these studies combine more than one strategy, described above.

In the monocular image context, previous studies mostly preferred the data-driven approach, which relies on the extraction of low level features, such as lines and rectangular structure of the roofs. For example, Jung and Schramm [15]) employ Hough transform to detect the rectangular shape of buildings. Also, Mayunga *et al.* [16] suggest an active contour model. In another study, Saeedi and Zwick [17] detect the lines and line perpendicularities on the boundary of the segments. A common disadvantage of these studies is that, their final outputs are generally limited to a specific type of building hypothesis such as a parallelogram structure.

Shadow information derived from monocular images is extensively used during the verification of the building hypotheses. Jin and Davis [18] use differential morphological profiles to find candidate buildings. In order to verify the candidates, they apply shape analysis, and used the shadow evidence. Akcay and Aksoy [19] propose another algorithm, which uses the shadow and sun azimuth angle information to detect buildings. The algorithm detects shadow regions on an over segmented image. Afterwards, candidate building regions are found using directional spatial constraints. Finally, the building outlines are determined after clustering the candidate regions using minimum spanning trees.

Some of the unsupervised algorithms employ techniques to discriminate and remove the irrelevant regions from the image, then focus on the regions which include buildings. For example, the algorithm proposed by Aytekin *et al.* [20] removes the vegetation and shadow regions. According to their assumption, the rest of the image consist of man-made regions only. In the next step, these regions are partitioned by

mean-shift segmentation algorithm. Finally, the long segments are eliminated in order to remove roads. Although, their algorithm removes some of the irrelevant regions, it has a critical problem. Due to their assumptions, many non-building objects like small bare lands, pools, parking lots will be detected as building.

Cote and Saeedi propose one of the recent unsupervised studies [21]. Their study aims to handle arbitrary illumination and complex building shapes without shape prior. They assume that, the rooftops have distinctive edges on the boundaries and a rooftop is constructed by a single type material, which implies that the color and reflection properties of a rooftop is uniform. Their algorithm starts with a clustering step. For each cluster, they apply some morphological operations and filters to extract possible building blobs. The centroids of the blobs are used as reference points for the rest of the algorithm. In the next step, they select the potential corners around the reference point. By using these potential corners, they estimate the outline of the buildings. Finally, they refine the results in order to eliminate false reference points. Although their algorithm has promising results, there are some vulnerabilities which may need additional attention. First of all, detection of the reference points is a very critical issue for the rest of the algorithm. Therefore,finding the number of clusters is a very important task for selecting good reference points. However, their method for selection of number of clusters may fail in large areas. Secondly, they apply filters to eliminate some of the blobs. For example, if the center of gravity of a blob is out-of-the-blob, then this blob is discarded. However, this type of a filter can eliminate L-shaped buildings. Finally, there are some parameters, which are selected manually according to the smoothness of the rooftops in an image. As a result, a manual parameter selection process is required to detect rooftops.

Huang and Zhang propose another unsupervised study which employs morphological operators in order to detect buildings [22]. In that study, they assume that the buildings are near to shadow regions. Moreover, they hypothesize that most of the buildings are bright regions and shadows are dark regions. After the brightness map generation, White Top-Hat and Differential Morphological Profiles (DMP) operators with different directions are used to construct Morphological Building Index (MBI). Similarly, Morphological Shadow Index(MSI) is generated after the Black Top-Hat and Differential Morphological Profiles(DMP) operators. Their algorithm uses a bottom-

up region merging algorithm to segment the image. The segments are detected as building if their MBI value is higher than a predefined threshold, and also the distance to a shadow region is lower than a predefined threshold. Finally, in order to reduce false alarm, they applied shape based post processing. Although their proposed study uses the shadow information, there are some shortcoming which cause performance problems. First of all, pre-defined thresholds, used for selecting segments has a crucial role. Then, they are not using the sun direction during the estimation of shadow distance. As a result, the bright regions, which are near to a shadow region could be detected as a building. Finally, during the post processing, each segment is scored based on its rectangularity, and if a building has a non rectangular shape, it can be eliminated during this stage.

Sirmacek and Unsalan [23] propose a probabilistic framework for building detection problem. They initially assume that the urban regions are previously detected with another algorithm. Therefore, they only focus on finding the location of buildings in the urban regions. Their algorithm stars with extracting local feature vectors from the given image using four different methods which are based on Harris corner detection, gradient-magnitude-based support regions, Gabor filtering and features from accelerated segment test (FAST). According to the authors, each local feature vector indicates a building for these locations. Therefore, for a location, the probability of being a building is related to the amount of local feature vectors nearby. For that reason, they consider each local feature vector as observation and use the variable-kernel density estimation method to detect buildings. Finally, they mix four different pdfs by different feature methods and obtain a final pdf. One of the shortcomings of that study is that the algorithm is not able to detect the boundaries of the buildings, which is one of the major requirement for real world applications. Moreover, the number of local feature vectors on non-building objects (like parking lots, highways,parks) will be very high in dense urban areas which causes false alarms. Also, the algorithm depends on an urban area detection algorithm. Therefore, the algorithm can miss buildings in low dense rural regions.

In a recent study, Ok et al.[2] propose a new approach to detect arbitrarily shaped buildings. The algorithm starts with detection of vegetation regions and also the shadows. The algorithm models the spatial arrangement between the shadow and the

related object. During this stage, the algorithm uses the sun direction to generate object direction and generates a fuzzy landscape. For that purpose, a new line-based non-flat structuring element is designed. In order to eliminate the landscapes that may occur due to shadows cast by non-building objects, vegetation mask is used. Moreover, solar elevation angle is used to calculate the length of the shadow of the possible lowest building and shadows smaller than that value are neglected. After the pruning stage, each landscape is analyzed individually and the building is detected by using Grab Cut partitioning algorithm. Ok et al. test the algorithm in more than 16 different regions and validate its robustness. However, the algorithm assumes that all of the buildings' shadows are observable and also can be detected by a shadow detection algorithm. Whereas, the shadow of a building can be blocked by various objects such as, trees larger buildings, bridges etc. In such cases, the algorithm is not capable of detecting the buildings.

2.1.2 Supervised Approaches for Building Detection

The studies, categorized under the heading of supervised approaches, use the prior information given by a supervisor. In most of the cases, a set of training samples(i.e. pixels, segments or features with ground truth data) is used to train the classifiers which are employed to classify a given test sample.

Sun et al. [24] propose a supervised interpretation model for the classification of targets in urban areas. They extract segments using pyramid-cut, and then compute color, texture, shape and location features for each segment. Finally, the segments are classified using a boosting algorithm. Chen and Blong [25], use RGB bands, and simple texture features such as the mean color value to extract buildings. Moreover, they use edge information to improve the results. Inglada [26] extracts low and high level geometric features which are aggregated and classified using SVMs. Fauvel *et al.* [27] propose a decision fusion approach in which the features are extracted by morphological operators. Then, the features are classified using neural networks and a fuzzy classifier. Finally, the posterior probabilities obtained from the outputs of the neural networks, and the class membership degrees obtained for the fuzzy classifier are fused. Turker and San [28] use color intensity values, normalized digital surface model

and Normalized Difference Vegetation Index (NDVI) for feature extraction, and the buildings are classified using SVMs. Most of the studies, published before 2010 are criticized in the previous survey reports [6] [7] [8].

One of the recent studies is presented by Cretu and Payeur[29]. Their main motivation is proposing an innovative combination of visual attention model, which is inspired from the human visual system, and machine learning approach for the building detection problem. Their algorithm starts with the watershed segmentation. Then, the regions responding to vegetation and shadow are eliminated. During this elimination, they use V and S color invariants, two indexes build upon the intensity values of bands. According to their solution the remaining segments can be classified as buildings, roads or distractors which can include parking lots, pools, driveways, etc. In order to classify, a bounding box is built around each of these segments. Then, computational attention model of Itti [30] is extracted from each bounding box. The saliency map is binarized and down-sampled to a map of size 16x16, which is used to extract a feature vector with 256 bins. Finally, the segments are classified by a least-squares support vector machine. Although the study is interesting, there are some important limitations. First of all, the algorithm is only evaluated in residential areas. Therefore, the vegetation filtering eliminates most of the regions. As a result the distractor problem is very limited. Also, it is known that the texture of the residential areas and urban areas are very different. Therefore, their descriptor, based on human visual system, does not proof its validity. Finally, the experiments show that the performance of the algorithm highly depends on the initial segmentation result. However, their proposed model does not provide a solution for under segmentation and over segmentation problem.

We also propose a supervised approach in [1], which is called Building Detection With Decision Fusion(BDDF). This method combines the detection results of multiple classifiers under a hierarchical architecture, called Fuzzy Stacked Generalization (FSG). In the first step of the algorithm, the image is segmented using Mean-shift segmentation method [31]. In order to optimize the parameters of the Mean-shift algorithm, they suggest a computationally efficient approach, called Overall Segmentation Quality(OSQ) for building detection problem. In order to select the optimum segmentation parameters, the training images are segmented by different parameter tuples. Then,

OSQ is calculated using these segmentation results. The optimal parameters set, which gives the highest OSQ, is used in the segmentation of the test images. Hence, the need of manual parameter selection is eliminated. After that, the segments, which belong to the vegetation and shadow regions are identified and discarded from the image. Next, various multi-modal color, shape and texture features are extracted from each segment. During the classification they use a hierarchical ensemble learning algorithm, called Fuzzy Stacked Generalization (FSG) [32]. Each feature space is separately used to compute the decision of an individual base-layer classifier which is represented by a class membership vector. Then, the decisions of base-layer classifiers are fused by aggregation to construct a set of meta-layer input feature vectors, which is fed to a meta-layer classifier. Although, that study is one of the recent state of the art studies, the requirement of generating training data prevents us to use it as an autonomous system.

Stankov and He also suggested a slightly different supervised method for building detection [33] compared to the methods mentioned above. In this approach, the supervisor information is not used for training. Instead, they use supervisor information for selecting samples for different type of buildings,with different roof colors. Hence, several classes are defined according to the roof colors. Then for each class, a gray scale image is generated based on the color similarity, called spectral similarity ratio (SSR). In these gray images, potential building locations have higher intensity values. Next, a set of morphological operation, called The hit-or-miss transform(HMT), which was previously used for building detection from panchromatic images [34], is performed on each gray scale images. Candidate buildings are obtained from each gray scale images. Finally, a size criterion is applied to candidate buildings in order to eliminate disconnected roads. Although, Stankov and He categorize their study as a supervised approach, the algorithm does not use any advantage of supervised classifiers. Also, the used structuring element in HMT affects the shape of the candidate buildings. As a result, selecting the structuring element become an important problem for extracting the correct boundaries.

The statistical properties and the problems of the training data is rarely considered in remote sensing community. In recent years, some of the researchers propose solutions for very specific cases. Tuia et al. deals with land use classification [35]. In their case,

the training samples are obtained from a test image. However, they remind that, the validity of training samples collected in field campaigns is crucial and a training set ,which covers small sub sets of the scene, will not be able to represent the whole image. In such cases a shift between the distribution of the training set and the test may occur. This problem is also known as covariance shift in machine learning. They propose to use active queries to learn the shift and sample new training examples in unknown areas of the image. In another recent study, Bruzzone and Marconcini propose a method for automatic updating of land-cover maps [36]. Their model deals with multi-temporal remote sensing images, which are periodically acquired over the same investigated area. The training samples are collected from the initial image, and the generated training set is used for the rest of the images. For this purpose, they propose a new classifier, domain adaptation support vector machine and the problem is modeled in the domain-adaptation framework.

Also, there are other studies, which import the data shift and the domain adaptation concepts from machine learning to the remote sensing [37, 3, 4]. These type of studies generally focus on specific problems, which deal with spatially similar or multi-temporal images. The algorithms try to adapt or shift the initial training data to the new test cases. Although these approaches are very promising, the current studies are dedicated for land use classification. Moreover, the intra-class dissimilarity of buildings, and lack of a powerful descriptor prevent to apply these strategies for our problem.

2.2 The Concept of Self-Supervision

As mentioned in the previous section, we know that the supervised classification techniques require manually generated training data. However, if we need to develop a autonomous system, the need of human interaction become one of the important problems. One of the possible solutions is finding a way to automatically generate training data. This concept is called as self-supervised and popularly used for robotic applications in order to generate autonomous devices [38, 39, 40, 41].

On the other hand, the power of self-supervised algorithms are not fully realized in remote sensing community. Only a few studies based on self-supervised approach

are available in remote sensing literature. One of them is presented by Seo et al., which is a self-supervised algorithm for parking lot structure extraction from aerial images [42]. Their algorithm has two layers. In the first layer, the algorithm uses geometrical meta-information for easy-to-find parking spots. The result of this layer is highly accurate, however most of the parking lots are missed. The second layer uses the result of first layer as a prime source of examples for self-supervised training. Doucette et al. propose a road detection approach, that uses the advantage of self-supervised approaches [43]. The algorithm starts with the selection of the candidate road components, performed with Anti-parallel-edge Centerline Extraction (ACE). In the next step, a road vector topology with a fuzzy grouping model is generated which links nodes from a self-organized mapping. This is followed by the Self-Supervised Road Classification (SSRC)step which automatically generates the training data and refines the road class. Also, Shackelford and Davis present a land cover classification algorithm by using self-supervised classification [44]. However, most of the implementation details are not mentioned in this study. Their land cover classification problem includes 6 classes which are Road, Building, Grass, Tree, Water, and Shadow. For each class, they use different methodologies during the training data generation step. Finally, the image is classified by a fuzzy pixel-based classifier.

To conclude, there are only a few self-supervised studies in remote sensing and none of them prove their robustness on a large test set. Besides their performance is highly related with the training sample selection criteria. However, there is an increasing demand for automatic solutions in remote sensing. Neither supervised nor unsupervised methods fully meet the needs of real world applications. Therefore, self-supervised concept is one of the possible solution.

2.3 Pre-processing Steps for Building Detection Problem

There is a diverse set of methodologies applied on the satellite images prior to the pattern recognition algorithm for detecting the buildings. The methodologies vary depending on the characteristics of the data acquisition devices (e.g sensor type, number of bands and resolution) or the type of the scene (e.g. residential, industrial, rural etc.). The pattern recognition methodology itself may require a specific method for

preprocessing such as, segmentation, vegetation elimination, histogram equalization. In the following subsections, we provide the necessary background for preprocessing required for our building detection method. The steps of preprocessing in this work include segmentation, line segment detection and feature extraction. The methods employed in these steps are briefly described next.

2.3.1 MeanShift Segmentation

Image segmentation is one of the typically used low-level vision tasks prior to object detection algorithms. For this reason, many different segmentation algorithms have been proposed in the literature [45, 46, 47]. However, the segmentation task is an ill-posed problem and there is no a unique ground truth [48]. As a result, the evaluation of segmentation results is a subjective topic.

On the other hand, the detection performance of the buildings highly depends on the segmentation output. In other words, the features extracted from the segments, which represent the buildings in a single region are more discriminative than the features extracted from the output of under-segmented or over-segmented images. However, the current segmentation algorithms are not able to represent buildings with a single region without any parameter optimization. Also, this manual parameter optimization process has to repeat for each different image in order to obtain single region building segments. In a recent study, we suggest a computationally efficient approach, which is designed for building detection problem to estimate the *optimal* meanshift segmentation parameters [1]. However, this approach is specialized for supervised classification and can not be used in our case. Therefore, we prefer to deal with over-segmentation results, which allow us to correctly detect a building boundary in the next steps. Meanshift is one of the appropriate segmentation algorithms, because its modularity makes the control of segmentation output very simple [46].

The Meanshift algorithm is a nonparameteric density estimation technique, which was proposed by Fukunaga and Hostetler [49]. However, Comaniciu and Meer successfully used this procedure for image segmentation purpose[46].

Given a set of vectors $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, ..., n$, with bandwidth values $h_i > 0$, the estimator

is

$$\hat{f}_K(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}k\Big(\big\|\mathbf{x}-\mathbf{x}_i\big\|^2\Big),$$

where $k(x)$, $x \geq 0$, is called the profile of the spherically symmetric kernel K:

$$K(\mathbf{x}) = c_{k,d}k\Big(\|\mathbf{x}\|^2\Big)\,,$$

where $K(\mathbf{x}) > 0$, $\|\mathbf{x}\| \leq 1$ and $c_{k,d}$ is the corresponding normalization constant.

The differential of the kernel profile is defined by the function

$$g(x) = -k^{'}(x),$$

when the differential of $k(x)$ exists.

When the gradient of the estimator is computed, we get the following mean shift vector:

$$\hat{\mathbf{x}} = \frac{\sum_{i=1}^{n}\mathbf{x}_i\frac{1}{h_i^{(d+2)}}g\Big(\|\frac{\mathbf{x}-\mathbf{x}_i}{h_i}\|^2\Big)}{\sum_{i=1}^{n}\frac{1}{h_i^{(d+2)}}g\Big(\|\frac{\mathbf{x}-\mathbf{x}_i}{h_i}\|^2\Big)} - \mathbf{x}\,,$$

which points towards the direction of maximum increase in the density [46].

Then, we achieve the nearest stationary point of the density iteratively via

$$\hat{\mathbf{x}}_{j+1} = \frac{\sum_{i=1}^{n}\mathbf{x}_i\frac{1}{h_i^{(d+2)}}g\Big(\|\frac{\hat{\mathbf{x}}_j-\mathbf{x}_i}{h_i}\|^2\Big)}{\sum_{i=1}^{n}\frac{1}{h_i^{(d+2)}}g\Big(\|\frac{\hat{\mathbf{x}}_j-\mathbf{x}_i}{h_i}\|^2\Big)}\,,\ j = 1, 2, \ldots\,.$$

Image segmentation based on mean shift procedure based is a straightforward extension of the discontinuity preserving smoothing algorithm [46]. Therefore, we estimate the the density in a joint domain, and define the density estimation kernel as the product of two radially symmetric kernels with a single bandwidth parameter for each domain as follows,

$$K(\mathbf{x}) = \frac{C}{h_s^2 h_r^p}k\Big(\Big\|\frac{\mathbf{x}^s}{h_s}\Big\|^2\Big)k\Big(\Big\|\frac{\mathbf{x}^r}{h_r}\Big\|^2\Big)\,, \tag{2.1}$$

where $h_s$is the spatial resolution parameter, which affects the smoothing and the connectivity of the segments, and h_r is the range resolution parameter [46]. Finally,

the algorithm eliminates the spatial regions, which are containing less than M, smallest significant feature size parameter, and merges them with the nearest regions in order to deal. As mentioned above an important advantage of this segmentation algorithm is its modularity which allow us the control of segmentation output. The smallest significant feature size parameter M and the range resolution parameter, h_r affect the number of regions in the segmented image [46]. Therefore, it is possible obtain under-segmented images by selecting small h_r and M values.

2.3.2 Line Segment Detector

Line detection is one of the recurrent problem in computer vision [50]. There are many approaches based on Canny edge detection [51] and Hough transform [52]. However, performance of such approaches are not sufficient in complex images, like satellite images. On the other hand, Line Segment Detector(LSD), one of the studies proposed by Gioi et al. has promising results in such complex regions[50]. Also, there are several studies which use LSD for object detection on remote sensing data [53].

The algorithm starts with computing the gradient magnitudes and generates an ordered list of pixels according to magnitudes. Starting from the first pixels in the list, which has the higher gradient magnitude, a region growing approach is used to obtain a line-support region. In this growing algorithm, if the candidate pixel share the same gradient angle with the entire region up to a certain tolerance, then it will be added into line-support region and the region angle is updated. In the next step, the algorithm finds a line segment, actually a rectangle, that best approximates line-support regions. In order to find the position of the rectangle, the center of mass is used, also the first inertia axis of the line-support region represents the direction of the rectangle. Finally, the algorithm validates all of the detected potential line segments

2.3.3 Feature Spaces

The features employed in this study (given in table 2.1) are collected from various papers on building detection [28, 24, 1]. The experiments, given in these studies, show that the selected features are suitable for building detection problem, when the

training and test sets are generated from same image. On the other hand, since our goal is to show the performance improving effect of the suggested architecture, we did not spend effort for feature selection problem. However, one should note that designing the feature space is a very crucial problem in remote sensing applications. It is possible to improve the performances if these problems are worked out. The implementation details of these features are given in section 4.3.1.

Table 2.1: Selected features from the building detection literature

feature Name	**Type of Feature**	**Used in**
Mean Color	Color	[24] [28]
Standard Dev. of Color	Color	[24][1]
Color Histogram	Texture	[24][1]
Area	Shape	[1]
Rectangularity	Shape	[1]
Axis Lengths	Shape	[1]
Direction	Shape	[1]

2.4 Performance Evaluation of Building Detection Algorithms

In the literature, different performance metrics have been introduced for building detection problem, such as[54], [55] and [56]. However in our study , we follow the well-known three metrics (Precision, Recall, and F-score)[57]. These metrics enable us to compare our results with two recent studies, reported in the literature [2][1]. In order to generate the ground truth, each segment is labeled as a building or non-building by an expert human operator.

The classifier decisions are grouped in four distinct categories as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) as given in Table 2.2. The pixels which are detected as building, and also marked as building in the ground truth data are categorized as TP. On the other hand, if a building pixel is not detected correctly, then the decision on the pixel is categorized as FN. If the algorithm labels a non-building pixel as a building, the decision on the pixel is categorized as FP. Finally, the decisions on the pixels, which are not detected as a building, and do not represent a building in the reference data, are categorized as TN.

After each pixel is classified, we evaluate the performances by using the f-score metric

Table 2.2: Definitions of TP, FP, TN, and FN

	Building	Background
Classified as Building	TP	FP
Classified as Background	FN	TN

[57] which is defined as

$$f - score = 2 * \frac{precision * recall}{precision + recall} \tag{2.2}$$

where,

$$precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$recall = \frac{TP}{TP + FN} \tag{2.4}$$

Note that, the above metrics require sufficient amount of ground truth data to evaluate the performance of a building detection method.

2.5 Chapter Summary

In this chapter, we presented a brief literature survey on building detection by considering the advantages and disadvantages of state of the art approaches. Then, we described the self-supervision concept and give some of the example approaches, which uses self-supervision in remote sensing. Moreover, we supplied some of the pre-processing technologies employed in this study, such as meanshift segmentation, extracted features and line detection. Finally, we provided precision, recall and f-score metrics for performance evaluation of the suggested building detection technique.

CHAPTER 3

MOTIVATION AND ANALYSIS

In this chapter, we present our motivation for proposing a novel approach for building detection. The technical inadequacies of the available algorithms, the problems of training data generation and intra-class dissimilarity of buildings prevent the previous studies to satisfy the need of robust autonomous detection systems. The basic idea for our motivation is given in the following section. Furthermore, the shortcomings of the available supervised approaches are analyzed in terms of a fully autonomous building detection system. 3.2.

3.1 Motivation For An Autonomous Building Detection System

During the literature survey, we observed that both of the unsupervised and supervised approaches have similar problems. A high proportion of the unsupervised approaches are limited to specific type of building hypotheses, such as a parallelogram structure [15, 16]. However, the shape of the buildings show a great variance. Besides, some of the studies make some critical assumptions and only designed for a specific case [20] [23]. As a result, many other objects, like pools, parks or parking lots, can be detected as buildings by these algorithms. Also, the variables, like the number of clusters or pre-defined parameters, become an important problem for unsupervised algorithms [22]. Nevertheless, the unsupervised algorithms do not depend on training sets, which requires a human interaction and effort for the long and tedious labeling process.

On the other hand, supervised approaches are capable of employing the state of the

art classifiers, which enable us to fuse different decision algorithms [1]. However, the performances of the supervised algorithms inherently depend on the various characteristics of the training data, such as the distribution functions, sample size, data collection errors. Generating a training data for each image is a very time consuming and expensive task. Even if this is possible, most of the supervised approaches assume that at least one-tenth of the image is used for training. However, the requirement of training data generation precludes them from being a fully automated building detection solution. Moreover, being able to use any current supervised approach with a global training set, which includes all type of buildings, is an unrealistic expectation.

One of the main reasons for the problems described above is the intra-class dissimilarity. Although the characteristics of the buildings in an image are relatively similar; buildings in two distinct images may show differences. Even when, we compare two images of the same building, which are acquired at different dates, they might be distinctly different due to the illumination, atmospheric and climatic changes (Figure 3.1). Moreover, the physical properties of the buildings can show a great variance due to the usage purposes, cultural differences and climatic conditions. Also, it is important to note that, apart from the other object detection problems, any man-made object which is used or intended for supporting or sheltering any use or continuous occupancy [58] can be defined as building. Therefore, the recent color, shape and texture features are not sufficient to describe this high level similarity. This limitation prevents to propose a generic unsupervised hypothesis or use a training data, which is generated from a variety of scenes.

(a) Image Sample 1 (b) Image Sample 2

Figure 3.1: Two different images of the same region, captured at different dates.

Fortunately, there is an important invariant, which makes detection of buildings possible. All of the buildings have an elevation and the elevation of the building cause a shadow on terrain, unless the sun elevation angle is 90°. In fact, shadow analysis has been considered to be one of the most important clues of buildings in monocular image processing. The relation of the building and its shadow is adequately used in [19] and [2]. However, these approaches may completely fail to recover building regions, if their cast shadow is not correctly detected or completely occluded by nearby objects such as trees.

Our motivation, in this thesis, is to propose a self-supervised automatic building detection framework, which eliminates the disadvantages of both unsupervised and supervised approaches. The framework that we propose effectively use one of the most important clues, the relation of a building and its cast shadow in order to solve the major problem of the supervised approaches, which is the training phase. Automatically generating training data allow us to use supervised classifiers for new test images without any human interaction. Further, this allows us to use shape, color and texture features (described in Section 2.3.3) whose performances are satisfying for building detection problem, when the training data is generated from the test image. Therefore, using the self-supervised concept in building detection problem and validating the robustness of the algorithm in different scenes will bring a new perspective for building detection literature.

3.2 Analysis of Training Data Sets

In this section, we mainly focus on the reasons why the current supervised building detection approaches are not suitable for automatic detection. Actually, there are many supervised algorithms available, which reports impressive performance results. The motivation of most of these algorithms is to propose an automated solution for building detection problem. However, these approaches are usually reckon without the adversity of training data generation and assume that a significant amount of the test image is already labeled for training. Besides, the manual training data generation is a very time consuming and expensive task. Moreover, due to the need of human interaction, they are actually semi-automated solutions.

On the other hand, being able to use any current supervised approach with a large training set, which includes a large variety of buildings, is an unrealistic expectation. In order to analyze the statistical properties of the learning set, we consider two scenarios and make several experiments based on them. In the first scenario, we use the same image for training and testing, which is the usual case for supervised algorithms. We incrementally increase the ratio of training data, and then analyze the similarity of statistical properties of the training data and testing set. Also, we use the training set for classification purposes and examine the relation between size of the training set and classification performance. In the second scenario, we inspect the possibility of using a global training data, which includes training samples obtained from different images. In these experiments, we prefer to use mean color, standart dev. of color and color histogram features, since the classification performances based on these features are relatively high [1]. In the following subsections, first we mention about our analysis methodology, then explain the results of the experiments for the first and second scenario.

3.2.1 The Methodology for Analyzing the Data

One of the basic assumptions in pattern recognition is that the performance of the supervised classifiers is related with the size of training data. Because, using larger training data allows us to extract better statistical information about our class. On the other hand, larger training data does not always provide better results in remote sensing applications due to several reasons, such as covariant shift [59]. For analyzing the properties of the data set, a set of experiments are performed, which aims to evaluate two important issue:(i) the statistical properties of the training data sets and (ii) the classification performances based on these sets.

In these experiments, 18 different multi-spectral images are used. 16 of these images are acquired from different regions, which provides us an important information for intra-class dissimilarity. All of these images are over-segmented by using mean-shift segmentation algorithm ($h_s = 4, h_r = 4, M = 75$). In order to generate the ground truth, each segment is labeled as a building or non-building by an expert human operator.

A segmented image, I_i, is represented by $S_i = \{s_{ji}, y_{ji}\}_{j=1}^{N}$ consisting of N segments

where y_{ji} is the ground truth label of the corresponding segment and the feature set, F_i , is constructed;

$$S_i = \{s_{ji}, y_{ji}\}_{j=1}^{N} \xrightarrow{\tau} F_i = \{\mathbf{x}_{ji}\}_{j=1}^{N}, \tag{3.1}$$

where $\mathbf{x}_{ji} \in R^d$ is the feature vector extracted from the j^{th} segment in the set and τ is the low-level information extractor.

Lets assume that, $\mathbf{D_{tr}}$ and $\mathbf{D_{te}}$ are the data sets of independent and identically distributed random variables with an unknown probability density function f. The consistency of $\mathbf{D_{tr}}$ and $\mathbf{D_{te}}$ is also related with the estimated statistics of f_{tr} and f_{te}. Therefore, in order to measure the consistency, we estimate the densities of the feature vector of each data set. In the literature, several density estimation approaches are available [60][61]. In this study, Kernel Density Estimation [62] is used, which is defined by $\hat{f}(x) = \frac{1}{nh} * \sum_{i=1}^{n} K(\frac{x-x_i}{h})$ where K is the kernel function and h is the bandwidth. Besides, the type of kernel is selected as Gaussian and bandwith is selected by "rule of thumb" approach [62].

Furthermore, a distance metric is required to evaluate the similarity of two different density functions. Due to the its simplicity, we prefer to use Integrated Square Error(ISE) in this study [63]. Although, the ISE is a distance, which gives an error for the estimation of a real density; it can also be used to calculate the similarity of two densities. Formally, ISE is defined as

$$ISE(f_a, f_b) = \int (f_a(x) - f_b(x))^2 \tag{3.2}$$

In addition to the similarity of training and testing sets' distributions, the effect of training data on classification performance has to be considered. For this reason, one of the popular approaches, Fuzzy K Nearest Neighborhood (fuzzy k-nn) algorithm [64] is selected and the classifier is trained by using D_{tr} and segment based classification performances are calculated according to (2.2).

In the following subsections, two scenarios are implemented on an image dataset consisting of 18 different scenes. The first scenario, deals with only one image. Therefore, the training and testing data sets are generated from the same image. The amount of

training sample is increased incrementally and its effects are evaluated. In the second scenario, it is assumed that the testing and initial training sets are generated from the same region, then new training samples , which are extracted from different images, are added to training set.

3.2.2 Scenario 1

In the first scenario; It is assumed that, we are able to extract training and testing sets from the same image, where the illumination, atmospheric conditions are same, and the buildings are similar to each other. In practice, most of the studies based on a supervised classifier consider a similar assumption. As a result, it is possible to expect that the properties of the buildings are similar to each other in both sets. In order to understand the effect of training set on a single image classification problem, a set of experiments are conducted.

In this experiments, for a selected image ,I_i, training data set ,$\mathbf{D_{tr}} = F_i^{tr}$, and the testing data set ,$\mathbf{D_{te}} = F_i^{te}$, are randomly generated where

$F_i^{tr} \subset F_i$,$\left|F_i^{tr}\right| = w_{tr} * |F_i|$,

$F_i^{te} \subset F_i - F_i^{tr}$,$\left|F_i^{te}\right| = w_{te} * |F_i|$

, w_{tr} is the ratio of training size:

$$w_{tr} = \frac{\text{number of training samples selected from the image}}{\text{number of all samples in the image}} \tag{3.3}$$

and $w_{te} = 0.3$ is the ratio of testing size.

F_i^{tr} is generated with different w_{tr} values between 0.01 and 0.6 and for each different w_{tr} value, this experiment is repeated 10 times with different randomly generated subsets.

During the experiments, the mean color descriptor is selected as τ. Then, the effect of w_{tr} on $ISE(f_{tr}, f_{te})$ and $ISE(f_{tr}, f_i)$ is analyzed, where f_i is the distribution of F_i. As expected, for larger w_{tr} values, the distribution of the training set approaches to the distribution of the whole image (Figure 3.2).

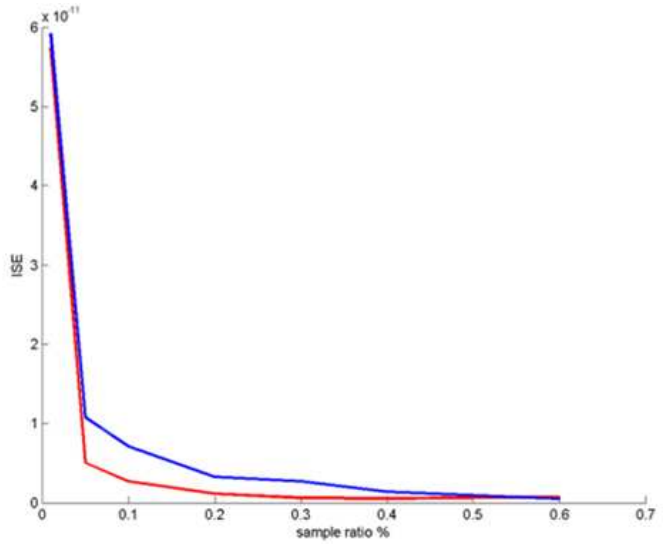


Figure 3.2: Effect of training sample ratio (w_{tr}) on similarity of $ISE(f_{tr}, f_{te})$ and $ISE(f_{tr}, f_i)$, which are represented by red and blue lines respectively.(Scenario 1)

Additionally, the effect of the selected training data on the classification problem is analyzed. In order to understand the effect, the w_{tr} value is increased incrementally. For each w_{tr} value, training and testing sets are randomly selected and samples in testing set is classified by fuzzy k-nn method. Subsequently, segment based classification performances are calculated according to (2.2). This classification procedure is repeated 10 times with different randomly selected training and testing samples. The highest performance values and also the average classification performances of these 10 experiment is recorded.

Figure 3.3 shows the f-score values with respect to w_{tr} for a given image (Image 2 in Figure 5.1). When the amount of w_{tr} is increased, the distribution of the training set converges to the distribution of testing set and the classification performance increases.

3.2.3 Scenario 2

In this scenario, our purpose is to investigate the existence of a universal training set for building detection problem. For this purpose, we incrementally increase the number of images in the training set. Meanwhile, we analyze the statistical behavior of the training set without considering the testing set, and then calculate the effect of

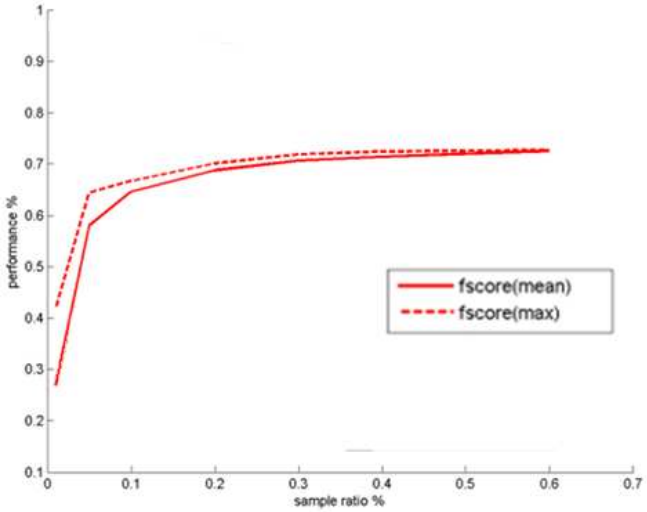


Figure 3.3: Effect of training sample ratio (w_{tr}) on classification performance for Scenario 1)

the generated training sets on the classification of the test sets.

One of the important criteria for a training set is the distinguishability of the building and non-building samples. For a selected training set, let f^+ represents the distribution of the positive samples (which means the buildings), f^- represents the distribution of the negative samples in the training set and **ISE(D)** represents the $ISE(f^+, f^-)$ in date set D. Therefore, the larger **ISE(D)** values show that negative and positives samples are more distinctive. As a result, we have a smaller training error. On the other hand, smaller **ISE(D)** values evince the similarity of positive and negative samples which causes larger training error.

Let's assume that for training purposes, we want to use d different images, $d < 18$. In such a case, g_n^d represents the n^{th} possible image combination which includes d images. Then, for a selected test image, I_i, possible training sets can be defined as,

$$D_{tr}{}_n^d = \bigcup_{j=1}^{d} F_{g_n^d(j)} \tag{3.4}$$

where $F_{g_n^d(j)}$ is the feature set of an image , which is the j^{th} element of the n^{th} combination of the possible sets with d images.

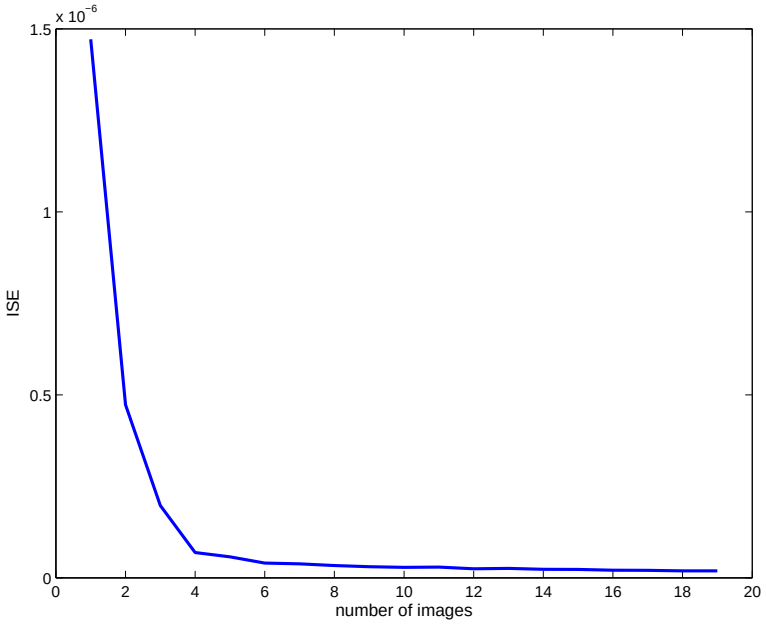


Figure 3.4: $\mathbb{ISE}$,which shows similarity of f^+ and f^-, with varying image number in training set for scenario 2

The average distinguishability of training data with d images is related with

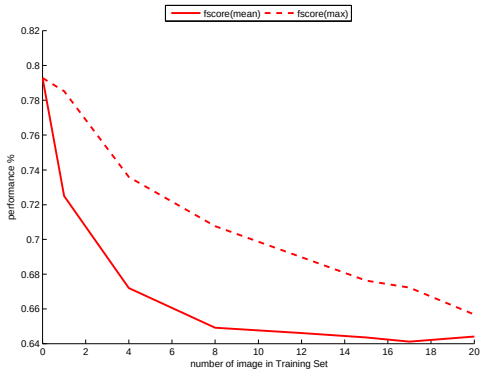
$$\mathbb{ISE}(D_{tr}^{d}) = \frac{1}{N'} \sum_{n=1}^{N'} \mathbf{ISE}(D_{tr}{}_{n}^{d}) \tag{3.5}$$

where N' is the number of combinations evaluated in the experiment - logarithm of the possible combinations ($log(\frac{M!}{d!*(M-d)!})$.

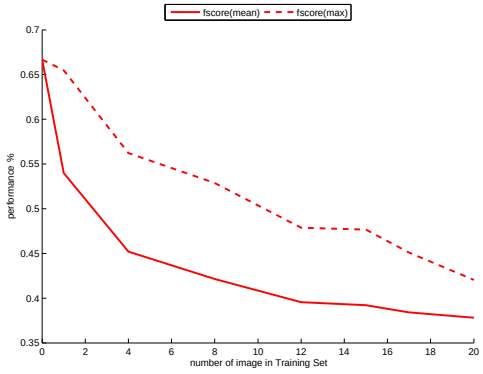
We increase the number of images used for training and analyze the change in $\mathbb{ISE}$. For all selected feature descriptors, distinguishability of training data is reduced when the number of images is increased. Figure 3.4 shows the change in $\mathbb{ISE}$ when mean color descriptor is used.

In order to understand the effect of indistinguishability of f^+ and f^- on classification performance, we also make several experiments. In these experiments, for a selected image ,I_i, 40% of the positive and negative samples in that image are selected for initial training, D_{tr}^{Init}. The rest of the samples used for testing, D_{te}. The the training set is extended with new samples in randomly selected d addition image. Finally, the test set is classified with the updated training set and the performance is recorded. The performance value may change by the selected images. Figure 3.5 plots the classification performances based on the mean color feature. The solid lines shows

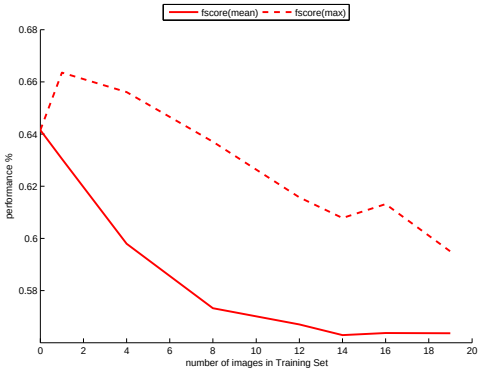
the average f-score value for N' combination, on the other hand dash lines shows the maximum f-score value. As expected, the classification performance falls down when the number of different images is increased.



(a) Mean Color



(b) Standart Dev. Color



(c) Color Histogram

Figure 3.5: Classification Performances for Scenario 2

3.3 Summary

In this chapter, we summarize the technical inadequacies of the available approaches and then present our motivation for proposing a self-supervising approach. Also, we define a methodology for analyzing training data and make several experiments in order to understand the reasons why the current supervised building detection approaches are not suitable for automatic detection in details.

CHAPTER 4

SELF-SUPERVISED DECISION FUSION FOR BUILDING DETECTION

In this chapter, we present a new building detection framework for monocular satellite images, called Self-Supervised Decision Fusion (SSDF). The proposed model is based on the idea of self-supervision, which aims to generate training data without any human interaction. This principle allows us to use the advantages of the supervised classifiers in a fully automated framework. In order to generate training data, the algorithm uses one of the most important clues, the relation of a building and its cast shadow.

The proposed approach includes three main steps (Figure 4.1) information extraction, automatic training sample selection and classification with decision fusion. Information extraction step begins with a pan-sharpening process. Basically, pan-sharpening is an image fusion approach, which combines a high resolution panchromatic image with a low resolution multispectral image, to obtain a high resolution multispectral image. After the pan-sharped image is generated, the algorithm detects vegetation in order to reduce the false alarms. In a similar way, spectral information is used to find shadows in the image. As described above, shadow is one of the important clues for our problem. If the solar illumination angles (azimuth and elevation) are known, then it is possible to estimate the position of the objects which generate shadow. Almost all satellite image products are shipped with a text metadata file that includes the required parameters. In order to estimate the possible location, a state of the art landscape generation approach is used [2]. Our previous experiments show that the

unsupervised approach, based on this landscape generation, can detect the buildings with a high precision [2]. On the other hand, the shadow of a building can be blocked by various large objects in the scene, such as trees, large buildings and bridges. Additionally, the shadow detection algorithm may miss some of the buildings' cast shadows. However, the detected shadow regions provide us a valuable information, which can employ for self training. Also, finding rectangular shapes is the core part of some of the unsupervised algorithms, but as mentioned in Section 2.1.1, the performance of such an algorithm will be limited. Nevertheless, the rectangular regions in an image may need attention. Therefore, the possible rectangular regions are simply detected by using connected perpendicular lines. Finally, mean-shift segmentation algorithm is used to obtain over-segmented regions. In the second step, the algorithm analyzes each segment using the extracted information and then aims to select the positive and negative samples with a "high" precision whih results in a reliable training set. The last step of the algorithm, extracts the features from the over-segments and trains the multi-layer decision fusion classifier by using automatically selected samples.

4.1 Information Extraction

In this section, we explain the steps of extracting information, which is necessary for automated training sample selection. In the first step, the algorithm generates a high resolution multispectral image. Then, algorithm finds the vegetation and shadow regions by using the spectral properties of the pixels. In the next step, algorithm uses the sun direction and models the spatial arrangement between the shadow and the related object. The algorithm also detects the possible rectangular regions, which is used for non-building objects selection. Finally, mean-shift segmentation algorithm is used to obtain over-segmented regions.

4.1.1 Pansharpening

Recently, almost all of the available high resolution optical sensors acquire the high resolution panchromatic image and the low resolution multispectral image independent from each other. Therefore, to obtain a single image that provides the highest spatial

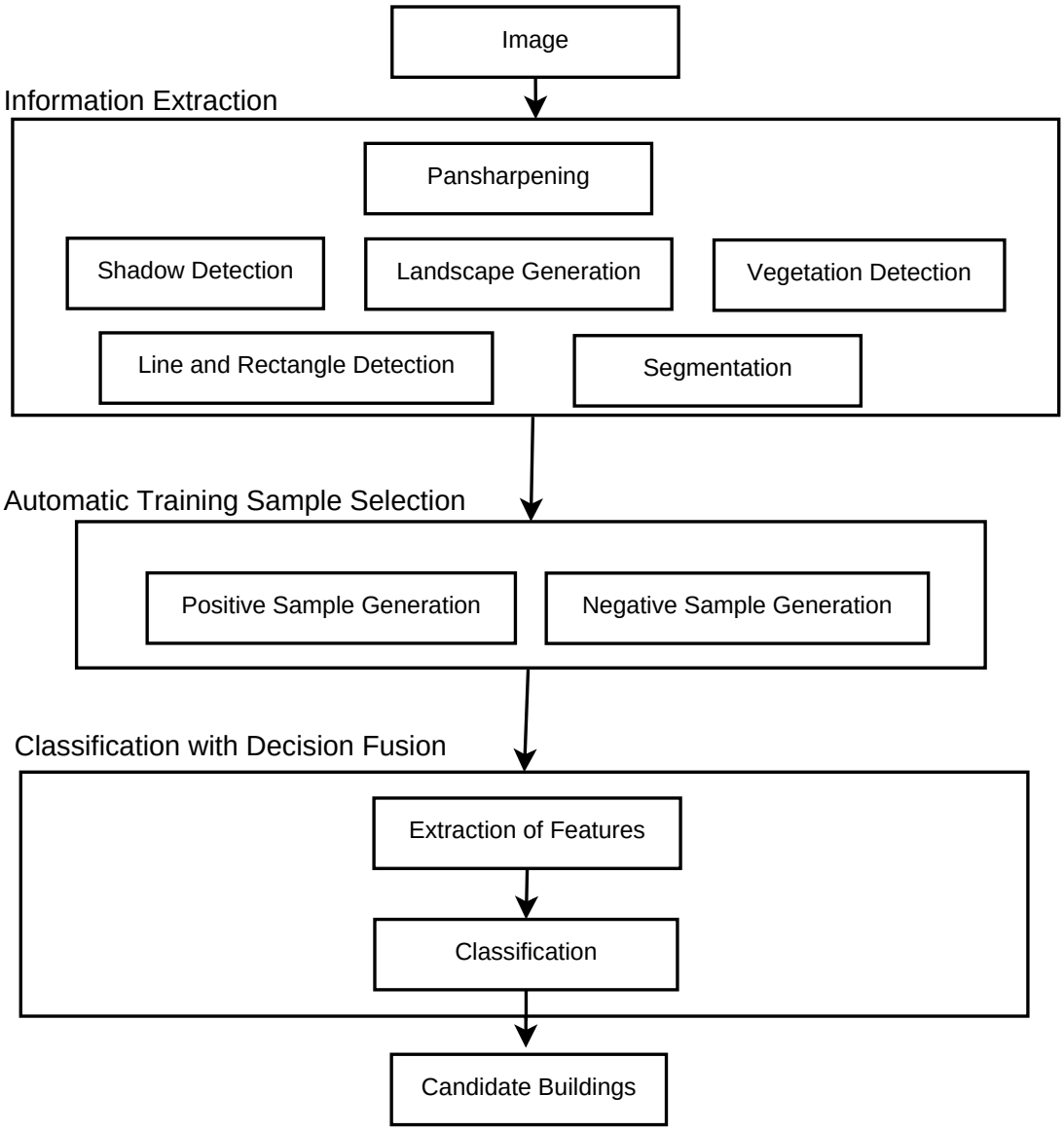


Figure 4.1: Flow Chart of the Self-Supervised Decision Fusion Method

and spectral resolution characteristics at the same time, a pan-sharpening process is necessary. In the literature, so far, a large number of different pan-sharpening methods are proposed [65, 66, 67, 68], and a number of comparison papers are published [69]. However, as clearly stated in [70], every pan-sharpening approach distorts spectral signatures of the images at least slightly. Also, our previous studies show that the output of the shadow detection algorithms, are affected by the spectral distortion.

In this study, one of our previous pansharpening algorithm is used [68]. The algorithm is basically based on Smoothing Filter-based Intensity Modulation (SFIM) pan-sharpening approach [65]. The general formula of the approach can be written

as,

$$P_{pan-sharped} = P_{ms}\frac{P_{pan}}{P_{lowpan}} \quad , \tag{4.1}$$

where P_{ms} is a pixel value of a lower resolution multispectral image, P_{pan} is the co-registered pixel of a higher resolution panchromatic image and P_{lowpan} is the same pixel of the smoothed panchromatic image (Figure 4.2). As different from SFIM, which use a square kernel, the proposed algorithm uses a disk kernel to apply a low pass filter and obtain smoothed panchromatic image. Besides, the optimum disk size is selected in order to minimize spectral distortions.

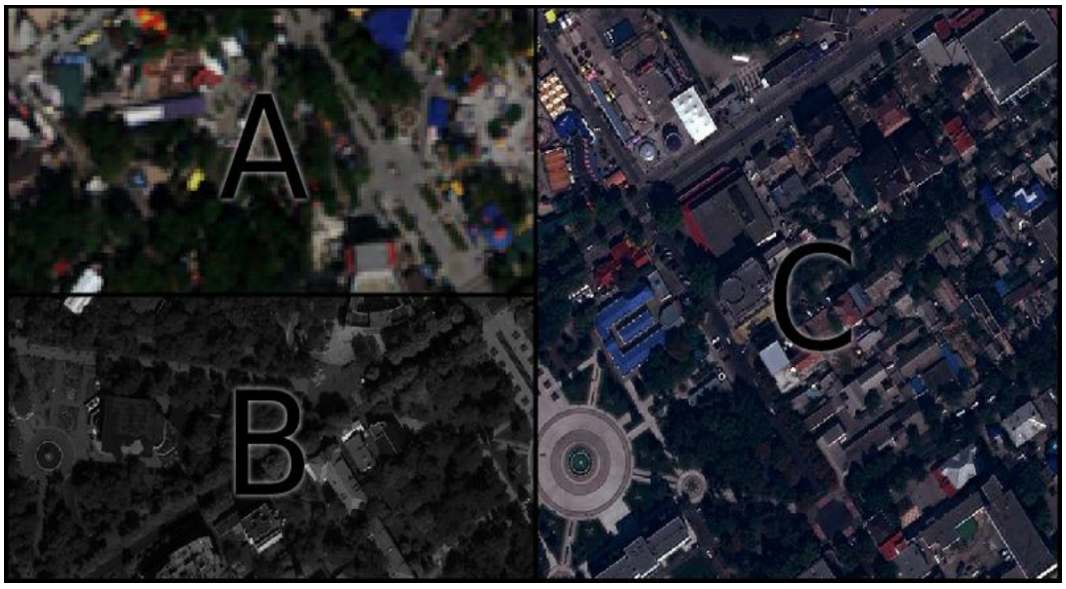


Figure 4.2: The result of pansharpening algoritm (A: Multispectral Image, B: Panchromatic Image, C: Pan-sharpened image)

4.1.2 Vegetation Detection

The live green plants need to do photosynthesis and during this process, they absorb solar radiation between 400- 700 nanometers and reflect the solar radiation after 700 nanometers wavelength [71]. As a result of this fact, vegetation detection is one of the popular usage area of 4 band multispectral satellite images. In order to detect vegetation regions, Normalized Difference Vegetation Index (NDVI) is utilized [72] as

,

$$NDVI = \frac{NIR - R}{NIR + R} \quad , \tag{4.2}$$

where R and NIR represent red and near-infrared bands, respectively. For each image, a threshold value t_{NDVI} is computed using the Otsu method [73]. Then, the pixels whose $NDVI$ values are greater than t_{NDVI} are marked as vegetation pixels, and a vegetation map ,$\mathbf{M_{veg}}$, is generated (Figure 4.3).

(a) Original Image

(b) Output, where the false color green indicates the vegetation regions

Figure 4.3: The result of vegetation detection algorithm

4.1.3 Shadow Detection Method

We use a multispectral false color shadow detection algorithm proposed by Teke *et al.* [74] due to two main reasons:(i) their approach utilizes the power of the NIR band, and (ii) the shadow detection algorithm does not require any user defined thresholds. In this approach, NIR, R and G are used to generate a false color image. Once the false color image (NIR-R-G) is obtained, it is converted into HSI color space. The *ratiomap* is defined as,

$$RatioMap = \frac{S - I}{S + I} \quad , \tag{4.3}$$

where S is the normalized saturation, and I is the normalized intensity. The extracted $RatioMap$ is binarized using Otsu method to obtain a set of pixels which represents both shadows and vegetation. Finally, the regions that belong to the vegetation objects are subtracted to obtain a binary shadow map, $\mathbf{M_{sdw}}$ (Figure 4.4).

(a) Original Image

(b) Output, where the false color red indicates the shadow regions

Figure 4.4: The result of shadow detection algorithm

4.1.4 Fuzzy Landscape Generation

In this section, we investigate the shadow evidence to focus on building regions. In this respect, we model the directional spatial relationship between buildings and their shadows with the prior knowledge of illumination direction which comes from the metadata of the image. For this goal, we use a fuzzy landscape generation approach especially designed for modeling the directional relationship between buildings and their shadows [2]. Once all landscapes are collected, a pruning process is applied to eliminate the landscapes that may occur due to non-building objects.

4.1.4.1 Initial Fuzzy Landscape Generation

In this section, we model the spatial arrangement between buildings and their shadows based on a morphological fuzzy relation approach suggested in [2]. Given a reference (shadow) object B and a direction specified by an angle α, the landscape $\beta_\alpha(B)$ around the reference object along the given direction can be defined as a fuzzy set of membership values in the image space [75]. The landscape membership values are defined in the range of 0 and 1, and the values determine the degree of satisfaction of the spatial relation. According to [75], the degree of satisfaction can be evaluated in terms of the angle $\theta_\alpha(x, b)$ measured between the unit vector along the direction α with respect to the horizontal axis and the vector from a point b in the reference object to the image point x. Therefore, a landscape with linearly decreasing membership values around the reference object based on the angle $\theta_\alpha(x, b)$ can be defined as :

$$\beta_\alpha(B)(x) = max\{0, 1 - \frac{2}{\pi} min\theta_\alpha(x, b)\}. \tag{4.4}$$

It is also shown that $\beta_\alpha(B)(x)$ can be computed using the morphological dilation ($\oplus$) of reference object B,

$$\beta_\alpha(B)(x) = (B \oplus v_\alpha)(x) \bigcap B^c \tag{4.5}$$

with a non-flat fuzzy structuring element v_α, where B^c is the complementary of B.

In [2], we proposed a line-based non-flat structuring element that involves membership values only in the direction defined by α. For this purposes, first, we proposed an isotropic non-flat structuring element formed by an exponentially decreasing non-linear function based on the Euclidean distances computed from point x to the center of the structuring element o,

$$v_\sigma(x) = e^{(-\frac{\|\vec{ox}\|}{\sigma})} \quad , \tag{4.6}$$

where the parameter σ determines the decrease rate of the exponential function and $|\vec{ox}|$ is the Euclidean distance of point x to the center of the structuring element. To guarantee zero membership values at the sides of the structuring element, we further multiply the exponential function with a linear term,

$$v_{(\sigma,\kappa)}(x) = e^{(-\frac{\|\vec{ox}\|}{\sigma})} max\{0, 1 - \frac{2\,\|\vec{ox}\|}{\kappa}\} \quad , \tag{4.7}$$

where κ is the size of the structuring element utilized. The structuring element in (4.7) is isotropic and provides membership values free from any directional term (Fig. 4.5 a-c). Therefore, we propose a different flat structuring element that keeps the directional information by defining a single straight line segment in the direction defined by α. For that purpose, we use the Bresenham line discretization algorithm [76], which selects the points in a structuring element in order to form a close approximation to a straight line.

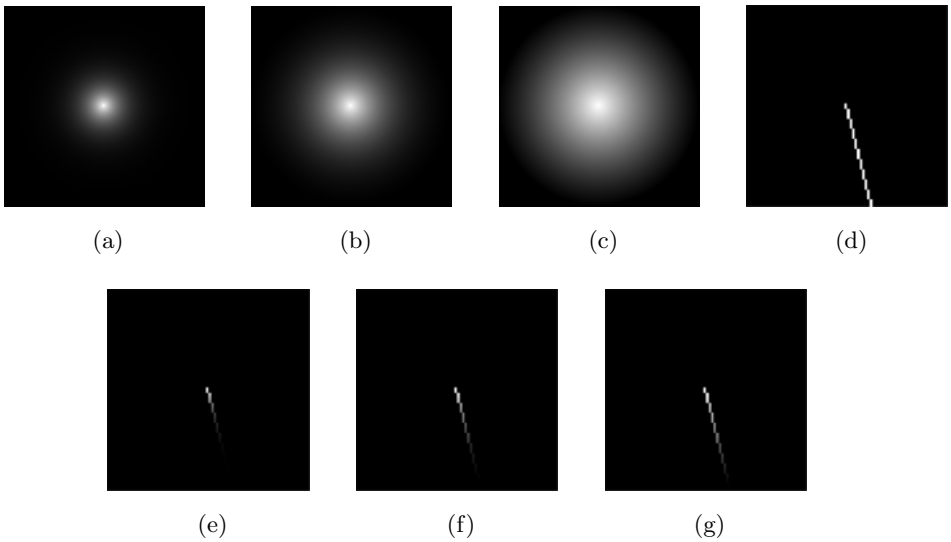


Figure 4.5: (a-c)Exponentially decreasing isotropic non-flat structuring elements (κ = 80 pixels) for different σ parameter values (10, 25 and 100), and (d) the directional flat structuring element for $\alpha = 165.6°$. (e-g) Proposed line-based non-flat structuring elements for each sigma value utilized

As a formal definition, let an infinite line whose normal is oriented at angle φ and passing through the kernel origin $o \in R^2$ is denoted by L_φ and defined as $L_\varphi = \{(x, y) \in R^2 \mid x\cos\varphi + y\sin\varphi = 0\}$, and let D is the discretization operator defining the Bresenham line discretization algorithm, a flat structuring element with a kernel size κ that describes the line segment can be achieved by $v_{L,\kappa}(x) = D(L_\varphi)$. Since L_φ has an infinite extent, a directional flat structuring element in the direction α can be computed as:

$$v_{L,\kappa,\alpha}(x) = round(1 - (\theta_\alpha(x, o))/\pi)D(L_\varphi) \quad , \tag{4.8}$$

where the $round(.)$ operator maps the computed membership values to the nearest

integer (Fig. 4.5d). Thus, we define our new line-based non-flat structuring element that only involves membership values in the direction defined by angle α by combining the structuring elements given in (4.7) and (4.8),

$$v_{L,\alpha,\sigma,\kappa,}(x) = v_{L,\kappa,\alpha} * v_{\sigma,\kappa}(x) \quad , \tag{4.9}$$

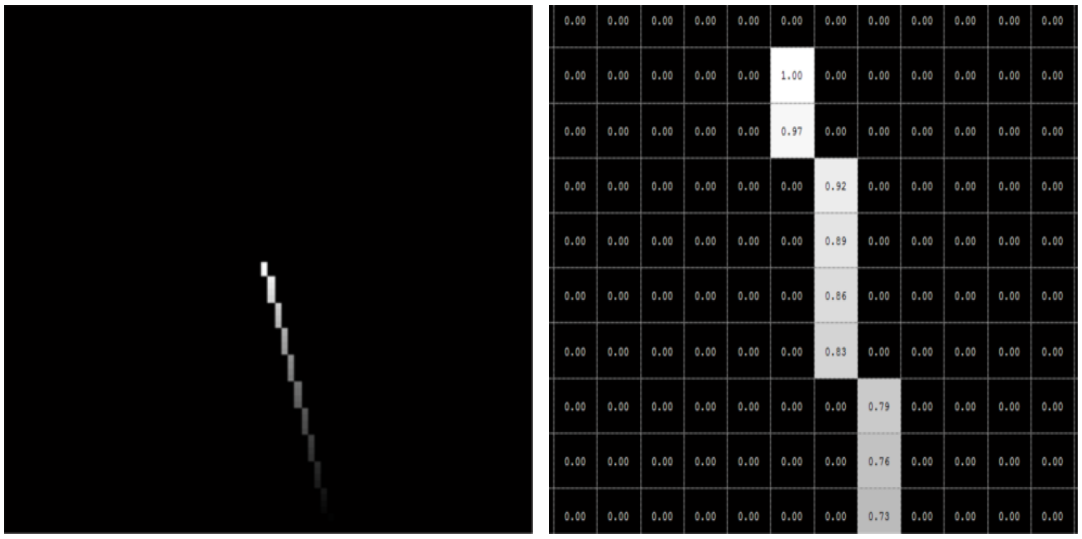
where $*$ is the element-wise multiplication operator (Fig. 4.5e-g). The proposed structuring element in (4.9) has several important properties. First, the exponential function and the linear term defined in (4.7) rely on the computed Euclidean distances between the points and the kernel center o. Therefore, it ensures that the membership values have continually decreasing behavior while moving away from the center of the structuring element (Fig. 4.6). Second, after the element-wise multiplication in (4.9), the computed values except in the direction defined by angle α have constantly set to zero membership values. Thus, the landscapes generated after the morphological dilation have membership values only in a region defined by the extent of the reference object B and the direction defined by angle α. Third, due to the continually decreasing behavior and the line-based structure, the landscapes $\beta_\alpha(B)(x)$ can be efficiently computed for very large images and/or large kernel sizes κ. To do that, we perform the morphological dilation with the structuring element given in (4.9) on just the pixels that define the perimeter of the reference object B,

$$\beta_\alpha(B)(x) = (B^{per} \oplus v_{L,\alpha,\sigma,\kappa})(x) \bigcap B^c \quad , \tag{4.10}$$

where B^{per} represents the perimeter pixels of the reference object B computed in an 8-neighbourhood connectivity, and in this study, a pixel in a reference object is labeled as a part of the perimeter if it is connected to at least one zero-valued neighborhood pixel.

4.1.4.2 Pruning the Fuzzy Landscapes

In an urban environment, buildings are not the only objects that cast shadows. Other objects such as trees, vehicles, garden walls, pools, bridges etc. have also elevation values different than the terrain height, and therefore, they also cast shadows. For those reasons, in an urban area, it is essential for a building detection task to eliminate the landscapes that may occur due to shadow casts by non-building objects.



(a) (b)

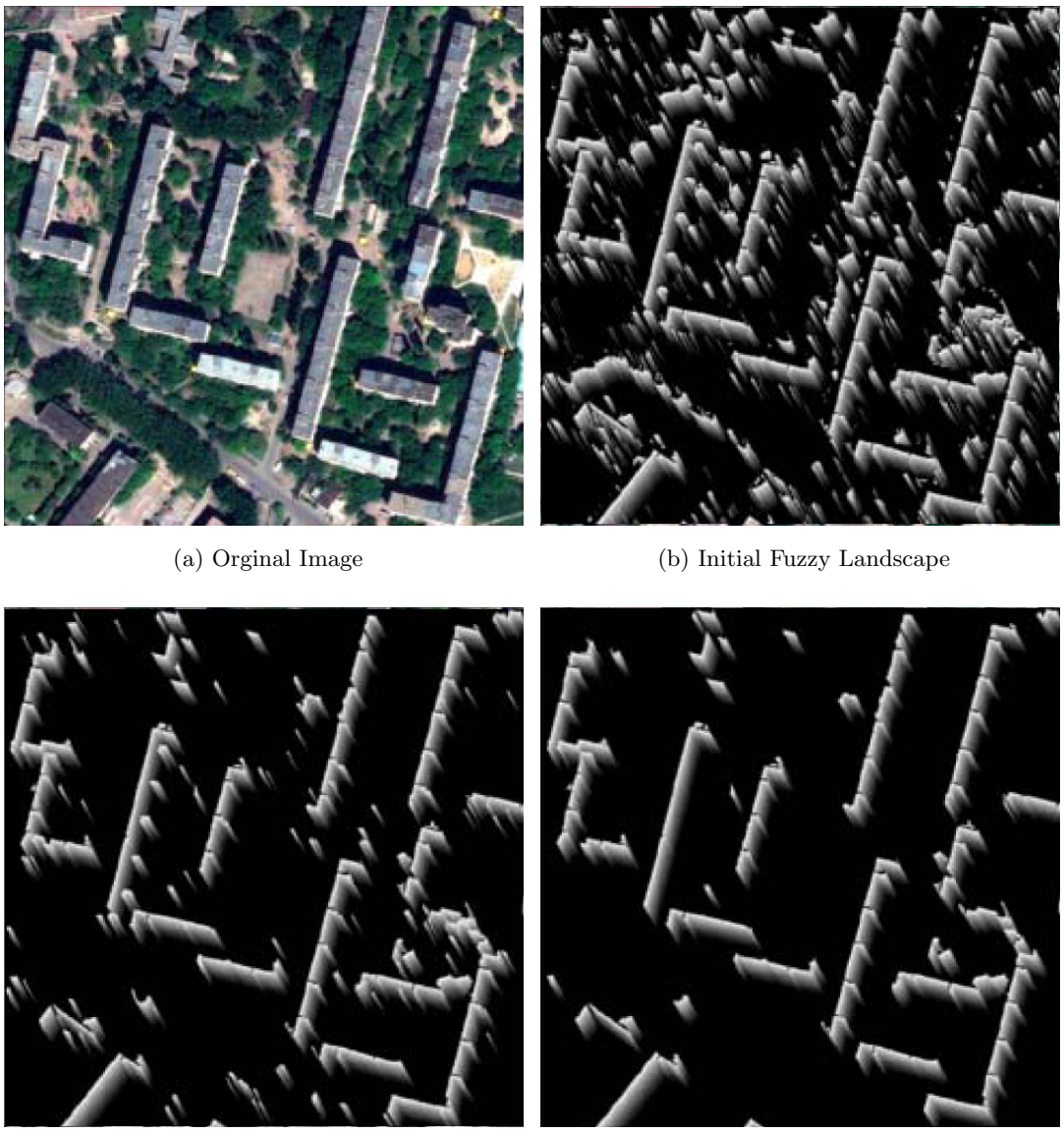
Figure 4.6: For a given Solar Azimuth Angle of 165.6° and a kernel size $\kappa = 80$ pixels, (a) the line-based non-flat structuring element ($\sigma = 100$), and the membership values of the pixels marked by the rectangle are superimposed over each pixel in (b).

In order to eliminate the landscapes generated by distinct vegetation objects, for each shadow region, we independently investigate the vegetation evidence within the close neighborhoods of the shadow regions. For this purpose, we define a search region in the immediate vicinity of each shadow object whose extent are outlined after applying a double thresholding (Tlow, Thigh) to the generated fuzzy landscapes generated. In all our experiments, we use 0.7 and 0.9 fuzzy membership values respectively for T_{low} and T_{high} in image space, and once the search region is defined, we check for vegetation evidence within the defined region with the help of the pre-computed binary vegetation mask, MV. In this study, we reject a fuzzy landscape region generated from a cast shadow if the computed ratio is equal or larger than a ratio threshold ($T_{veg} = 0.7$). Figure 4.7b illustrates the fuzzy landscapes generated from cast shadows before and after the pruning of vegetations.

Shadow information can also be used for estimating the height of an object from monocular images. To separate the landscapes of building and other non-building objects, we assess the height difference of the objects compared to the terrain height. Our aim is to investigate the length of the shadow objects in the direction of illumination to enforce a pre-defined height threshold value. For this purpose, for a given solar

elevation angle (ϕ) and a minimum building height threshold (T_{height}), we compute the minimum shadow length(L_{min}) that should be cast on a flat surface:

$$L_{min} = \frac{T_{height}}{tan(\phi)}. \tag{4.11}$$



(a) Orginal Image

(b) Initial Fuzzy Landscape

(c) After Vegetation Analysis

(d) After Height Analysis

Figure 4.7: An example result of Initial Fuzzy Landscape and Pruning the Fuzzy Landscapes

Once the minimum shadow length is computed, we generate a directional flat structuring element similar to $v_{L,\kappa,\alpha}(x)$ in (4.8), whose length is equal to L_{min} in the

direction of illumination. Since the shadow boundaries is a key-point in the process and the perimeter pixels of the shadow objects are already computed (B_{per}), for each shadow object, we use the generated directional flat structuring element to search the number of perimeter pixels that satisfies the length L_{min}. In this paper, if none of the perimeter pixels of a shadow object is found to be satisfying the length L_{min}, we assume that the shadow is cast from a non-building object, and thus, the generated fuzzy landscape is rejected. In an urban area, most of the vehicles (e.g., cars, trams, and single-decker buses), garden walls and pools, and some of the bridges used rather than vehicular traffic have height differences of less than 3–4 m compared to the terrain height. Therefore, in this study, we utilize a single height threshold (T_{height}) to eliminate the landscapes generated by non-building objects and obtain final fuzzy landscape map, $\mathbf{M_{fzy}}$. Figure 4.7 illustrates the fuzzy landscapes generated from shadow objects before and after applying a height threshold, $T_{height} = 3$ m.

4.1.5 Line and Rectangle Detection

As mentioned in Section 2.1.1, there are many approaches based on line and rectangle detection. In fact, both lines and rectangles are strong clues for an existing building. However, we also know that not only all of the buildings are not rectangle, but also there is no perfect rectangle detection algorithm. For that reason,we employ the following strategies in our method:

- We detect the lines as supplementary clue for selecting building samples.
- We use rectangles to make sure that any non-building samples are selected from these regions.

The algorithm first detects the set of lines, L, in the image by using LSD [50]. After that, for each connected line pairs, the algorithm checks whether they are perpendicular or not. In the next step, the algorithm fits two detected perpendicular lines $line_1$ and $line_2$ into a rectangle $R = \{P_1, P_2, P_3, P_4\}$ by finding the coordinates of 4^{th} corner, P_4:

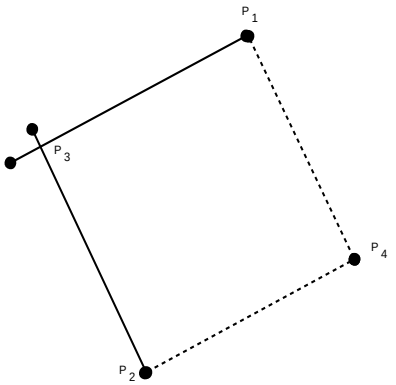


Figure 4.8: Estimating the rectangle by checking the perpendicularity of two joining lines.

$$P_4(x) = ((m_2 * P_1(x) - P_1(y)) - \frac{m_1 * P_2(x) - P_2(y)}{m_2 - m_1} \quad , \tag{4.12}$$

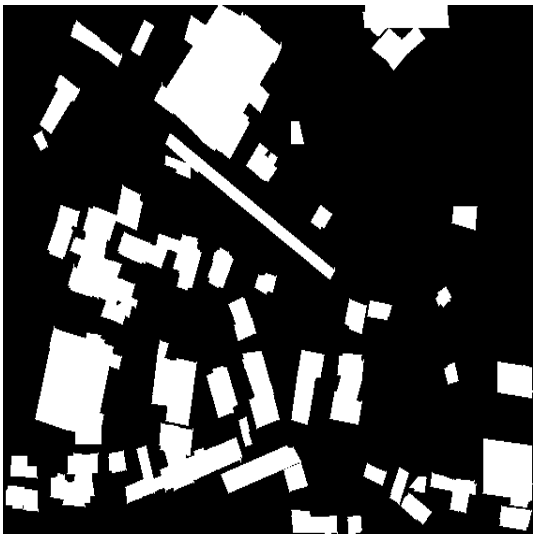
$$P_4(y) = m_1 * \frac{P_1(y) - m_2 * P_1(x)}{m_1 - m_2} - m_2 * \frac{P_2(y) - m_1 * P_2(x)}{m_1 - m_2} \quad , \tag{4.13}$$

where m_1, and m_2 are the slopes of the detected lines, P_1 and P_2 is the end points of $line_1$ and $line_2$ respectively. Also, P_3 is the connection point of $line_1$ and $line_2$. Finally, algorithm generates a binary map, $\mathbf{M_{rec}}$, which includes all possible rectangles (Figure 4.9).

4.1.6 Mean-shift Segmentation

In this study, Meanshift segmentation algorithm is used for image segmentation[46]. In the employment of the Mean-shift algorithm to image segmentation, R, G and B pixel values are transformed into Luv color space. A fixed kernel, with bandwidth parameters h_s and h_r is constructed. Then, the kernel is shifted to the point corresponding to the mean value of the sample points (feature values of the pixels in Luv space) which are within the corresponding bandwidth of each sample point. This process is iteratively applied until the Mean-shift vector converges to a point at which the density estimations are stationary, i.e. do not change. After the convergence is achieved, the pre-segments which are within the pre-defined spatial and spectral proximity, are merged. Finally, some of the *smallest* regions are eliminated or merged

(a) Original Image



(b) Output

Figure 4.9: The result of rectangle detection algorithm

with the nearest regions to compensate for the noisy patches. As mentioned in Section 2.3.1, we prefer to deal with over-segmentation results, which allow us to correctly detect a building boundary in the next steps. Therefore, meanshift parameters are selected as $h_s = 4, h_r = 4, M = 75$.

4.2 Automated Training Sample Selection

In the previous section, the algorithm extracts the information from the image relevant to the building object. The maps of vegetation, fuzzy landscape and rectangles provide important clues for the self-supervision task. For example, it is possible to claim that the roof of a building is not covered by plants. Moreover, the fuzzy landscape may represents the locations for buildings with some degree of accuracy. Hence, the regions with high landscape probability can be a candidate positive sample. On the other hand, a rectangle region has a high possibility of being a building, which has to be evaluated by the classifier. This type of expert analysis indicates that information provided by the vegetation maps, fuzzy landscape maps and existence of rectangles can be employed during the extraction of training set.

In this section, our purpose is developing an algorithm to select the positive and negative samples. Also, we know that the amount of the selected samples and the degree of their correctness are the two important factors, which affect the performance of the classifiers. Therefore, we aim to optimize both of the requirements. Before selecting the positive and negative samples, the algorithm assign extracted information into the segments. For a given segment, s_i , the vegetation score, R^i_{veg}, is calculated as:

$$R^i_{veg} = \frac{1}{M_i} * \sum_{j=1}^{M_i} (\mathbf{M_{veg}}(x_{ij})) \quad , \tag{4.14}$$

where M_i is the number of pixels in segment s_i and $\mathbf{M_{veg}}(x_{ij})$ is the intensity value of j^{th} pixel in s_i, extracted from vegetation map, $\mathbf{M_{veg}}$. Similarly, the landscape, R^i_{land}, and rectangularity scores, R^i_{rect}, are calculated as:

$$R^i_{land} = \frac{1}{M_i} * \sum_{j=1}^{M_i} (\mathbf{M_{fzy}}(x_{ij})) \quad , \tag{4.15}$$

and

$$R^i_{rect} = \frac{1}{M_i} * \sum_{j=1}^{M_i} (\mathbf{M_{rect}}(x_{ij})) \quad . \tag{4.16}$$

Algorithm 1 (Positive and Negative Sample Selection)

1. For $i = 1$ *to* N*, where* N *is the number of segment*

2. *If* $R^i_{rect} < T^n_{rect}$*, where* $T^n_{rect} = 0.20$

3. *If* $R^i_{land} < T^n_{land}$*, where* $T^n_{land} = 0.01$

8. *Select s_i as a negative sample*

2. *If $R^i_{veg} < T^b_{veg}$, where $T^b_{veg} = 0.01$*

3. *If $R^i_{land} > T^b_{land}$, where $T^b_{land} = 0.28$*

4. *Extract the boundary of s_i, B_i*

5. *Find the lines, L_i, over B_i*

6. *Calculate shape score R_{shp}*

7. *If $R^i_{shp} > T^b_{shp}$, where $T^b_{shp} = 0.80$*

8. *Select s_i as a positive sample*

Then, the algorithm begins by selecting positive and negative samples (Algorithm 1). In order to select a segment as a negative sample, the algorithm looks for two criteria: Rectangularity score R^i_{rect} must be smaller than an acceptable rectangularity threshold for non-buildings(T^n_{rect}) and the landscape score must be smaller than T^n_{land}. For each candidate positive segment, whose vegetation score, $\mathbf{M_{veg}}$, is smaller than the acceptable vegetation threshold for a building segment (T^b_{veg}) and R^i_{land} is higher than a minimum landscape threshold for a building segment(T^b_{land}) the algorithm makes a final verification. As mentioned above, the correctness of the training data is very critical. Although the fuzzy landscape approach prunes most of the noise, we can face with incorrect fuzzy landscapes, due to false alarms caused by the shadow detection algorithm. Therefore, we calculate a final score by using the detected lines, L. The shape score R^i_{shp} is defined as:

$$R^i_{shp} = \frac{|\zeta \bigcap \Gamma|^{\eta+1}}{|\zeta|} \quad , \tag{4.17}$$

where ζ defines the set of pixels in the boundaries of segment in s_i, Γ is a set of pixels in L ,$|\cdot|$ is the set cardinality. Also, η is the number of intersected line pairs over the boundary, where the angle between these two lines is larger than 70°. The calculated shape score must be higher than minimum shape threshold, T^b_{shp}, in order to be used as a positive sample. This final elimination also can loss some of the correct positive candidates, however the risk of using mislabeled samples is more critical.

In this study, since we do not utilize any a priori information of the size and shape of the building objects, the threshold T^b_{land} can be selected intuitively. Moreover, the

threshold T_{rect}^{n} affects the amount and correctness of the negative samples. The effects of different values of these thresholds on the building detection performances should be carefully investigated.

4.3 Classification with Decision Fusion

One of the major contribution of this study is to employ a decision fusion method for building detection problem. In this section, we describe the steps of this algorithm, which fuses the decision of more than one classifier under a fuzzy stacked generalization (FSG) architecture (Figure 4.10). The suggested FSG method is constructed in three major steps

- Extraction of multiple feature spaces
- Training and Recognition at each individual feature space
- Fusing the decision of multiple classifiers.

The following subsections explains the above mentioned steps.

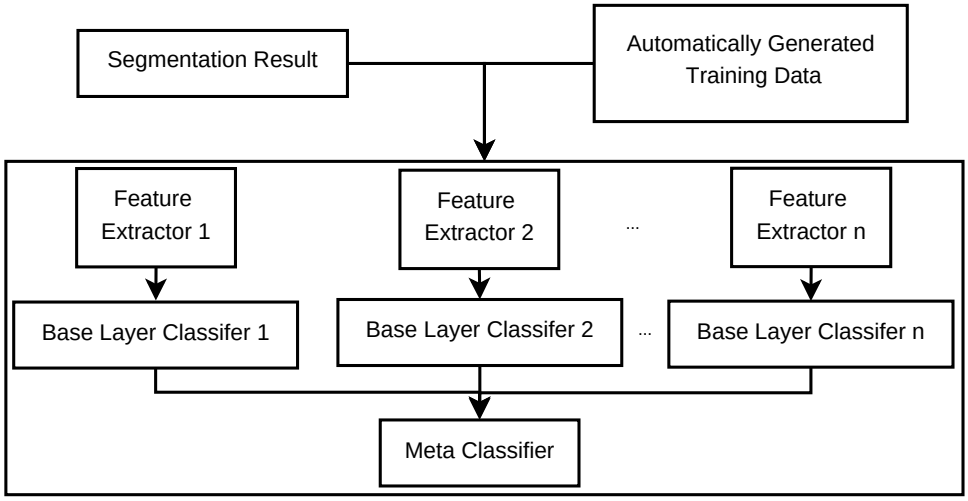


Figure 4.10: Flow Chart of FSG

4.3.1 Feature Extraction in Multiple Feature Spaces

Given a segmented image represented by the dataset $D_{\hat{\theta}} = \{s_i, y_i\}_{i=1}^{N}$ consisting of N segments s_i, a set of features F_k is constructed using the following k^{th} feature extractor τ_k, $\forall k = 1, \ldots, K$;

$$D_{\hat{\theta}} = \{s_i, y_i\}_{i=1}^{N} \xrightarrow{\tau_k} F_k = \{\mathbf{x}_{i,k}\}_{i=1}^{N}, \forall k, \tag{4.18}$$

where $\mathbf{x}_{i,k} \in \mathbb{R}^{d_k}$ is the feature vector extracted from the i^{th} segment in the dataset, and y_i is the label of the corresponding segment. The base-layer consists of K classifiers Υ_k each of which is fed by a set of distinct features extracted from the same segment. A feature set F_k, which is fed to an individual base-layer classifier Υ_k, is selected to represent different physical properties of the segments. Therefore, τ_k are considered as low-level information extractors.

The features employed in this study, are collected from various papers on object detection literature [77] [24]. Since our goal is to show the performance improving effect of the suggested Fuzzy Stacked Generalization architecture, we did not spend an extra effort for feature or sample selection problem. However, one should note that designing the feature space is a very crucial problem in remote sensing applications. It is possible to improve the performances if these problems are worked out.

A diverse set of features, which is used to extract information about color, texture and shape characteristics of segments, is given in the following subsections. Mathematical definitions of the features which are used to construct F_k and the dimensions of the feature vectors are provided in Table 4.1. The dimensions of the feature vectors described in this section, are given in Table 4.1.

4.3.1.1 Color Features

For each segment s_i, we compute the mean color and standard deviation of the intensity values of the pixels in s_i, $mc_i \in \mathbb{R}^d$ and $stdc_i \in \mathbb{R}^d$, where $d = 4$ is the number of color bands.

Table 4.1: Mathematical definitions of the feature extraction algorithms

F_k **Formula**	**Dim.**	**Description**
$mc_i = \left[mc_i^1, mc_i^2, mc_i^3, mc_i^4\right]$	4	$mc_i^d = \frac{1}{M_i}\sum_{j=1}^{M_i} x_{ij}^d$, where d is the selected band and M_i is the number of pixel in s_i and x_{ij} is the intensity value of j^{th} pixel in s_i
$stdc_i = \left[stdc_i^1, stdc_i^2, stdc_i^3, stdc_i^4\right]$	4	$stdc_i^d = \sqrt{\frac{1}{M_i}\sum_{j=1}^{M_i}(x_{ij}^d - mc_i^d)^2}$,
$hist\_pdv_i = \left[pdv_{i1}^1, pdv_{i2}^1, .., pdv_{i8}^4\right]$	32	$pdv_{im}^d = \frac{h_{im}^d}{\sum_{n=1}^{8} h_{in}^d}$, where h_{im}^d is the number of elements in m^{th} bin which is generated from d^{th} band of s_i.
$rectangularity_i = \frac{M_i}{major_i minor_i}$	1	$minor_i$ and $major_i$ are the length of major and minor axes lengths
$axis\_lengths_i = [major_i, minor_i]$	2	$minor_i$ and $major_i$ are the length of major and minor axes lengths
$area_i = M_i$	1	M_i is the number of pixel in s_i
$direction_i = \alpha_i$	1	α_i is the direction of major axis

4.3.1.2 Shape Features

Three shape features are extracted for each segment. The first feature is the $area_i$, which is the number of pixels belonging to s_i. In order to utilize the rectangular shape properties of the buildings, we compute $rectangularity_i$. Moreover, we concatenate the major axis length and minor axis length to generate a two-dimensional shape vector, $axis\_lengths_i$. In addition the direction of the major axis $direction_i$ is used as a shape feature.

4.3.1.3 Texture Feature

The color histogram is used as a texture feature. Each color band is divided into 8 histogram bins for each segment $s_i, \forall i = 1, \ldots, N$. Then, a probability density vector ($hist\_pdv_i$), which describes the ratio of the number of pixels belonging to each bin to the total number of pixels, is computed for each segment s_i.

4.3.2 Decision Fusion with Fuzzy Stacked Generalization

We use an ensemble learning algorithm, called Fuzzy Stacked Generalization (FSG) [32] for decision fusion. Each feature space is separately used to compute the decision of an individual base-layer classifier which is represented by a class membership vector of a segment s_i. Then, the decisions of base-layer classifiers are fused by aggregation to construct a set of meta-layer input feature vectors, which is fed to a meta-layer classifier.

4.3.2.1 Building Detection at the Base-Layer Classifiers

Building detection problem is formulated as a two-class classification problem in which the segments with $y_i = 0$ belong to non-building class, and the segments with $y_i = 1$ belong to building class. The membership values, $\mu_c(\mathbf{x}_{i,k})$ of the features $\mathbf{x}_{i,k}$ for the c^{th} class, $\forall c = 1, 2$, are computed by each classifier Υ_k using a fuzzy k-NN algorithm [78] as

$$\boldsymbol{\mu}_c(\mathbf{x}_{i,k}) = \frac{\sum_{j=1}^{\kappa} l(\eta_j(\mathbf{x}_{i,k}))(\rho_j(\mathbf{x}_{i,k}))^{\frac{-2}{\psi-1}}}{\sum_{j=1}^{\kappa}(\rho_j(\mathbf{x}_{i,k}))^{\frac{-2}{\psi-1}}}, \tag{4.19}$$

where $l(\eta_j(\mathbf{x}_{i,k}))$ is the label of the j^{th} nearest neighbor of $\mathbf{x}_{i,k}$, which is $\eta_j(\mathbf{x}_{i,k})$, and $\rho_j(\mathbf{x}_{i,k})$ is the Euclidean distance between $\mathbf{x}_{i,k}$ and $\eta_j(\mathbf{x}_{i,k})$, $\forall j = 1, ..., \kappa$. ψ is the fuzzification parameter and taken $\psi = 2$, as suggested in [78].

Therefore, a two-dimensional membership vector is obtained for each segment s_i at the output of each base-layer classifier as

$$\boldsymbol{\mu}(\mathbf{x}_{i,k}) = \big[\mu_1(\mathbf{x}_{i,k}), \mu_2(\mathbf{x}_{i,k})\big]. \tag{4.20}$$

The above membership vector of each segment s_i carries information about the decisions of the classifiers $\Upsilon_k, t\forall k$, for identifying the class label of s_i with respect to its input feature vector $\mathbf{x}_{i,k}$. The classification performance of Υ_k is defined as,

$$Perf_k = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{y}_{i,k}}(F_k) \tag{4.21}$$

, where $\hat{y}_{i,k} = \max(\boldsymbol{\mu}(\mathbf{x}_{i,k}))$ is the performance result of each base-layer classifier Υ_k , calculated from the membership vector of each $\mathbf{x}_{i,k}$,

and $\delta_{\hat{y}_{i,k}}(F_k)$ is the Dirac measure defined as;

$$\delta_{\hat{y}_{i,k}}(F_k) = \begin{cases} 1, \hat{y}_{i,k} \in F_k \\ 0, \hat{y}_{i,k} \notin F_k \end{cases} \tag{4.22}$$

Note that the membership vectors in (4.20) satisfy $\sum_{c=1}^{2} \mu_c(\mathbf{x}_{i,k}) = 1$, which map the feature vectors with different sizes and dynamical ranges to a set of fuzzy membership vectors on a line in a two dimensional space, called *decision space*, for each classifier.

4.3.2.2 Decision Fusion at the Meta-Layer

The class membership vectors $\boldsymbol{\mu}(\mathbf{x}_{i,k}), \forall k = 1, \ldots, K$, obtained at the output of each base layer classifier represent the fuzzy decisions of the classifiers for a sample s_i. In the suggested FSG architecture the fuzzy decisions are concatenated to create the feature vector of s_i at the meta-layer as follows:

$$\boldsymbol{\mu}^{meta}(s_i) = \big[\boldsymbol{\mu}(\mathbf{x}_{i,1}), \ldots, \boldsymbol{\mu}(\mathbf{x}_{i,k}), \ldots, \boldsymbol{\mu}(\mathbf{x}_{i,K})\big]. \tag{4.23}$$

Note that the vectors in this feature space satisfy

$$\sum_{k=1}^{K} \sum_{c=1}^{2} \mu_c(\mathbf{x}_{i,k}) = K. \tag{4.24}$$

Therefore, the feature vectors $F_{meta} = \{\boldsymbol{\mu}^{meta}(s_i)\}_{i=1}^{N}$ lie in a $2K$ dimensional feature space, called *fusion space*.

The concatenation operation, which creates the fusion space at the input of the meta-layer classifier, avoids the drawbacks of the classical concatenation of features of different dynamical ranges. Since the output feature spaces of base-layer classifiers consist of membership vectors in the range of $[0, 1]$, it yields a set of compatible feature vectors. Therefore, there is no need for normalization. In addition, the base-layer classifiers transform all the high dimensional feature vectors $\mathbf{x}_{i,k}$ to 2-dimensional decision spaces. This property avoids the curse of dimensionality problem in the fusion space, provided that $2K$ is small compared to the dimensions of the feature vectors $\mathbf{x}_{i,k}$ at the base-layer.

At this point, we convert the learning problem of the meta-layer classifier Υ_{meta}, into seeking the least squares solutions for the linear transformations between the

membership vectors $\boldsymbol{\mu}(\mathbf{x}_{i,k})$ and their corresponding class labels, $y_{i,k}$. Then, the membership matrix $\mathrm{M} \in \mathbb{R}^{N\times 2K}$ is defined as follows:

$$M(s_i) = \left[\boldsymbol{\mu}(\mathbf{x}_{i,1}) \, , ..., \, \boldsymbol{\mu}(\mathbf{x}_{i,k})\right]. \tag{4.25}$$

Similarly, the corresponding class label matrix $Y \in \mathbb{R}^{N\times 2}$ is defined as

$$Y(s_i) = \left[Y_1(s_i), Y_2(s_i)\right],$$

where

$$Y_c(s_i) = \begin{cases} 1, & \text{if} \quad y_i = c \\ 0, & \text{otherwise.} \end{cases} \tag{4.26}$$

Finally, the classification problem can be formalized as the following least-squares minimization problem

$$\min_{Z} \|MZ - Y\|_2, \tag{4.27}$$

where $\|\cdot\|_2$ is ℓ_2 norm, and Z is the linear transformation matrix. At the meta-layer, a *Least-squares Classifier* (LSC) is employed to solve (4.27) using Moore-Penrose pseudoinverse of M which is $M^{\dagger} = (M^T M)^{-1} M^T$ as follows

$$Z = M^{\dagger} Y. \tag{4.28}$$

Note that LSC provides a unique least-squares solution to (4.28) if M has full rank columns. In other words, LSC can correctly classify the segments if the base-layer classifiers provide *indepedent* and *complementary* information about the segments [79].

4.4 Summary

In this chapter, we present an automated self-supervised building detection framework for monocular satellite images, called Self-Supervised Decision Fusion (SSDF). The algorithm includes three main steps, which are information extraction, automatic training sample selection and classification with decision fusion. In the information extraction step, algorithm generates shadow, vegetation, landscape and rectangularity maps. The automatic training sample selection step uses these maps to select positive

and negative samples. Finally, in the last step, algorithm extracts the features from the over-segments and uses them in a multi-layer decision fusion classifier.

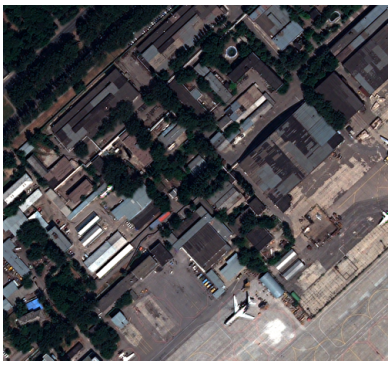
CHAPTER 5

EXPERIMENTS

In this chapter, we analyze the strengths and weaknesses of the developed algorithms. Recall that the suggested SSDF method is based on the idea of self-supervision, which aims to generate training data without any human interaction. For this reason, algorithm uses several important clues, such as shadow evidence and rectangularity during the positive and negative sample selection. In the first part of the experiments, we evaluate our sample selection approach and analyze the relation between the automatically generated training data and classification performances.

The suggested SSDF method, also, borrows a recent decision fusion approach from one of our previous studies [1]. In [1],we analyze the pros and cons of various supervised techniques for building detection problem and compare the performances of single classifiers to that of decision fusion methods of multiple classifiers. In this thesis, we enrich the decision fusion method by the self-supervision approach and provide the experimental analysis under the proposed self-supervised decision fusion framework.

It is also important to compare the proposed SSDF method with the state of the art approaches. Our proposed model is one of the premier studies, which use self-supervision in Remote Sensing. Therefore, we compare our performances with not only an unsupervised approach [2], but also a supervised approach [1], which are previously proposed by us. Finally, we evaluate the robustness of the algorithm to the algorithmic parameters.



(a) Image Sample 1

(b) Image Sample 2

(c) Image Sample 3

(d) Image Sample 4

(e) Image Sample 5

(f) Image Sample 6

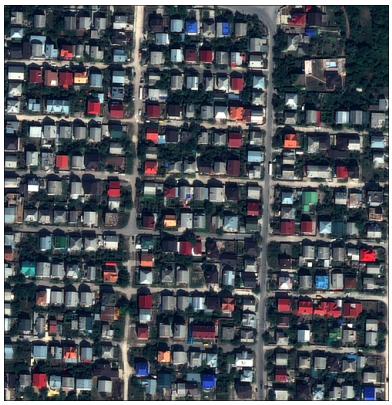
Figure 5.1: The image dataset (1-6) used in the experiments.

(a) Image Sample 7

(b) Image Sample 8

(c) Image Sample 9

(d) Image Sample 10



(e) Image Sample 11

(f) Image Sample 12

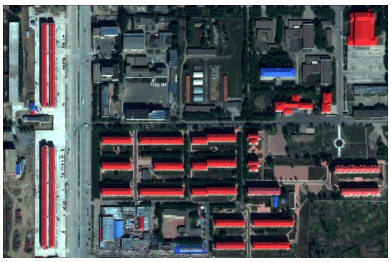
Figure 5.2: The image dataset (7-12) used in the experiments.

5.1 Image Data Set

During the experiments, we select images acquired from two different satellites, namely, QuickBird (61 cm) and Geoeye-1 (50 cm). All imagery used in this study include four

(a) Image Sample 13

(b) Image Sample 14



(c) Image Sample 15

(d) Image Sample 16

(e) Image Sample 17

(f) Image Sample 18

Figure 5.3: The image dataset (12-18) used in the experiments.

multispectral bands (R, G, B, and NIR) with a radiometric resolution of 11 bits per band. The images were corrected for image distortions introduced by the collection geometry, terrain displacement and rectified to a datum and a map projection. We

assessed the building detection performance of the proposed approach over 18 test sites. Among the test images, 4 of them were selected from two different QuickBird images, whereas the rest 14 belong to different Geoeye-1 images. The test images utilized in this study are illustrated in Figure 5.1, Figure 5.2 and Figure 5.3. Since the proposed approach relies on shadow information, the images are carefully selected to extensively cover varying illumination and acquisition conditions that might be encountered during image collection. The solar elevation angle ranges between 33.88 and 78.12, and this fact exposes that a wide range of cast shadow lengths is visible and considered within the selected image data set.

5.2 Evaluation of Automated Training Sample Selection

As mentioned in Chapter 4, the quality of the automatically generated training data is one of the important factors, which effect the performance of the classifiers. Therefore in this section, we evaluate the sample selection approach. For each test image, the positive (building) and negative (non-building) samples are generated according to the methods explained in Section 4.2. After the positive and negative samples are generated, the selected samples are analyzed by using the ground truth data.

The possible decisions of the sample selection algorithm are grouped in six distinct categories as Positive Selected Buildings(B_p), Negative Selected Buildings(B_n), Unselected Buildings(B_p), Positive Selected Non-Buildings(N_p), Negative Selected Non-Buildings(N_n), Unselected Non-Buildings(N_p) as given in Table 5.1.

Table 5.1: The possible decisions of the sample selection algorithm.

Ground Truth \ Algorithm	Selected as positive sample	Selected as negative sample	Unselected
Building segment	B_p	B_n	B_u
Non-building segment	N_p	N_n	N_u

For each test image, we consider the qualitative and quantitative properties of samples on each category and make analysis by using three metrics: correction, completeness

and selection error, as follows:

$$\text{correctness of positive samples} = \frac{B_p}{B_p + N_p} \quad , \tag{5.1}$$

$$\text{correctness of negative samples} = \frac{N_n}{N_n + B_n} \quad , \tag{5.2}$$

$$\text{completeness of positive samples} = \frac{B_p}{B_p + Bn + B_u} \quad , \tag{5.3}$$

$$\text{completeness of negative samples} = \frac{N_n}{N_n + N_b + N_u} \quad , \tag{5.4}$$

$$\text{selection error for buildings} = \frac{B_n}{B_p + B_n} \quad , \tag{5.5}$$

$$\text{selection error for non-buildings} = \frac{N_b}{N_n + N_b} \quad . \tag{5.6}$$

The correctness, basically, defines the precision of the sample selection algorithms. As a result, higher correctness ratio for positive samples means lower mislabeled positive samples in the training set. On the other hand, the completeness of the samples represents the proportion of selected samples of a certain class in the test set. Finally, selection error considers the proportion of mislabeled samples of a certain class in the training set.

It is crucial to analyze the effect of selected training samples on classification task. Since each segment represents a sample in our problem, we calculate segment based performances. For each image, the correctness, completeness and also corresponding segment based precision/recall values are given in Table 5.2 and Table 5.3 for positive and negative samples, respectively.

In these experiments, we realize that, negative sample selection approach is able to select more than 50% of all the negative samples in an image before starting classification. Also, the average correctness of the negative sample is 98.5%, which is relatively high compared to the correctness of the positive samples. However, it is also important to note that the number of total negative samples is much more than

Table 5.2: The correctness and completeness of selected positive samples.

	Correct-ness	B_p	$B_p + N_p$	B_p+Bn+B_u	Complete-ness	Seg. based Recall.	Seg. based Prec.
image 1	69,3%	104	150	287	36,2%	65,2%	64,7%
image 2	62,9%	88	140	227	38,8%	68,7%	66,1%
image 3	78,0%	234	300	679	34,5%	73,3%	64,8%
image 4	92,1%	117	127	158	74,1%	100,0%	72,5%
image 5	96,1%	74	77	112	66,1%	96,4%	91,5%
image 6	94,2%	81	86	158	51,3%	88,6%	89,7%
image 7	97,6%	122	125	222	55,0%	93,2%	88,8%
image 8	67,2%	127	189	213	59,6%	79,3%	54,9%
image 9	64,6%	135	209	221	61,1%	88,2%	54,8%
image 10	86,7%	365	421	457	79,9%	96,3%	65,8%
image 11	77,3%	371	480	680	54,6%	91,9%	60,6%
image 12	92,5%	135	146	304	44,4%	87,5%	89,6%
image 13	79,0%	109	138	157	69,4%	94,3%	61,4%
image 14	72,1%	111	154	171	64,9%	86,5%	55,8%
image 15	71,1%	350	492	614	57,0%	82,2%	69,1%
image 16	70,7%	147	208	207	71,0%	90,8%	53,9%
image 17	87,8%	144	164	299	48,2%	73,2%	78,8%
image 18	72,3%	185	256	374	49,5%	91,2%	68,3%
Average	79,5%	166	214	308	56,4%	85,6%	69,5%

Table 5.3: The correctness and completeness of selected negative samples.

	Correctness	N_n	$N_n + B_n$	$N_p + N_n + N_u$	Completeness
image 1	98,5%	1486	1508	2259	65,8%
image 2	93,4%	538	575	1358	39,6%
image 3	98,9%	2513	2541	3644	69,0%
image 4	100,0%	61	61	389	15,7%
image 5	100,0%	261	261	599	43,6%
image 6	98,5%	389	395	682	57,0%
image 7	96,2%	175	182	347	50,4%
image 8	99,9%	964	964	1772	54,4%
image 9	99,4%	976	982	1676	58,2%
image 10	96,4%	346	359	981	35,3%
image 11	98,6%	430	436	2363	18,2%
image 12	99,1%	1749	1764	2280	76,7%
image 13	99,5%	1405	1412	1976	71,1%
image 14	98,6%	344	349	794	43,3%
image 15	99,2%	1710	1723	3191	53,6%
image 16	97,7%	292	299	699	41,8%
image 17	99,4%	1651	1661	2383	69,3%
image 18	99,6%	2024	2033	3244	62,4%
Average	98,5%	961	972	1702	51,4%

the total positive samples. As a result, 1.5% of the selected negative samples, which are mislabeled, may have a negative effect on the training space.

In order to assess the quality of the training set, we evaluate the effects of the selected samples on classification performances. The performance of the SSDF depends on several important factors. The first one is, the correctness of the positive samples, which has a major effect on the precision of the algorithm. In our test set, we observe that the images with lower correctness values for positive samples (Image 2, Image 9, Image 8 and Image 1) also have lower precision rates (Table 5.2). Moreover, the high number of correctly selected negative samples have positive effect on precision, because compensates for the incorrect positive samples. We analyze the selection error for non-buildings in Table 5.4. In most of the images, the number of correctly selected negative samples is large enough to reduce the effect of the mislabeled ones on the overall performance. However, when the number of correctly selected negative samples is relatively low, the precision value of the classification may be low, depending on the correctness of the positive samples (ex: Image 11 and Image 16).

The recall rates of the proposed SSDF method, depends on several parameters. One of them is the completeness of the positive samples, as described above. Also, the selected negative samples becomes important, when the number of incorrect negative samples is high. In Table 5.5, we analyze the selection error for buildings. The recall values of Image 1, Image 2 and Image 3, which have the highest selection errors, are below the average recall value of our test set.

The statistical properties of the selected samples are very crucial for both precision and recall values. Let's assume that, all of the samples in a class are very similar. In a such case, a generated training data, which has a low completeness value, can classify the rest of the image. On the other hand, if the building class have two distinct types of buildings and all of the generated building samples are extracted from one type of building, no matter what the correctness values are, the detection of the other type of buildings will depend on the dissimilarity of non-building class.

Table 5.4: Selection error for non-building class.

	N_p	N_n	selection error
image 1	46	1486	3,0%
image 2	52	538	8,8%
image 3	66	2513	2,6%
image 4	10	61	14,1%
image 5	3	261	1,1%
image 6	5	389	1,3%
image 7	3	175	1,7%
image 8	62	964	6,0%
image 9	74	976	7,0%
image 10	56	346	13,9%
image 11	109	430	20,2%
image 12	11	1749	0,6%
image 13	29	1405	2,0%
image 14	43	344	11,1%
image 15	142	1710	7,7%
image 16	61	292	17,3%
image 17	20	1651	1,2%
image 18	71	2024	3,4%

Table 5.5: Selection error for building class.

	B_n	B_p	Selection error
image 1	22	104	17,5%
image 2	38	88	30,2%
image 3	29	234	11,0%
image 4	0	117	0,0%
image 5	0	74	0,0%
image 6	6	81	6,9%
image 7	7	122	5,4%
image 8	1	127	0,8%
image 9	6	135	4,3%
image 10	13	365	3,4%
image 11	6	371	1,6%
image 12	15	135	10,0%
image 13	7	109	6,0%
image 14	5	111	4,3%
image 15	13	350	3,6%
image 16	7	147	4,5%
image 17	10	144	6,5%
image 18	9	185	4,6%

5.3 Comparison of the Meta-layer with Base-layer results

The classification approach employed in this study (FSG), is introduced in our recent study [1]. In this study, we analyze the performance of the meta-layer classifier with different descriptors and classifiers. Since our goal is to show the performance improving effect of the suggested self-supervised architecture, we did not spend effort for a detailed evaluation of the base layer classifiers.We analyze the performance results only for the fuzzy k-NN classfiers using individual features and compare them with FSG in Table 5.6. In the base-layer, the highest f-score values are obtained from three classifiers, which are using mean color, color histogram and standard deviation of color. Also, the performance of FSG is higher than the base layers. On the other hand, it is possible to improve the performances if feature selection problem is worked out. The performance improvement of FSG for Image 4 can be visualized in Figure 5.4.

Table 5.6: Pixel Based Performance Results of Base-Layer Fuzzy k-NN Classfiers Using Individual Features.

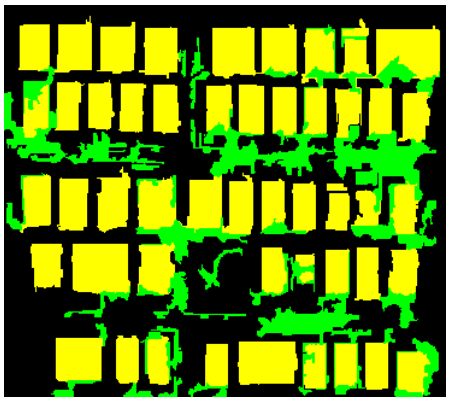
	Precision	Recall	F-score
Mean color	78,7%	81,6%	80,1%
Std color	68,1%	71,8%	69,9%
Axis lengths	71,1%	66,4%	68,7%
Direction	65,2%	63,8%	64,4%
Color Histogram	77,4%	71,6%	74,5%
Rectangularity	56,2%	56,9%	56,5%
Area	66,2%	64,4%	65,3%
FSG	78,9%	86,9%	81,7%

5.4 Comparison of SSDF with Other Studies

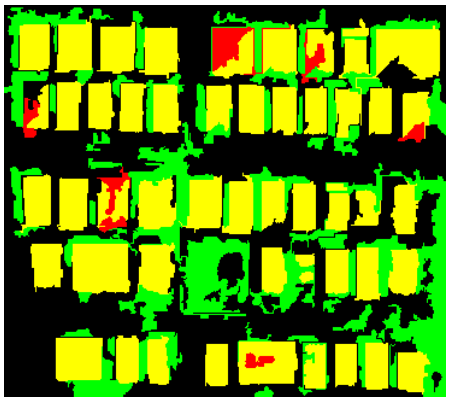
In this section, we compare the suggested SSDF approach with our previous studies [1, 2]. The details of these studies are given in Section 2.1. Note that, these algorithms are based on supervised and unsupervised approaches. Therefore, these experiments also allow us to see the pros and cons of unsupervised, supervised and self-supervised approaches.

Let us start by explaining the experimental set up. It is well known that, comparing

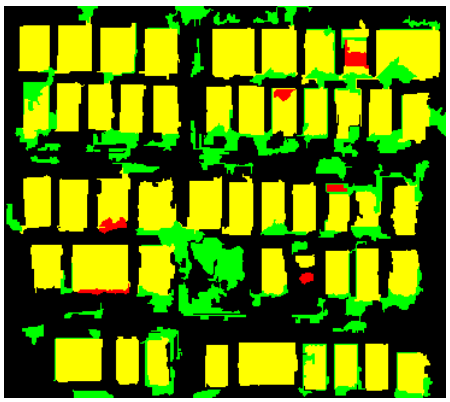
(a) Original Image



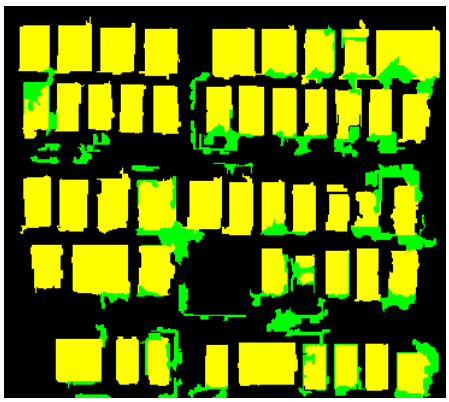
(b) Mean Color (f-score:83,4%)



(c) Std. Dev Color (f-score:75,7%)



(d) Color Hist. (f-score:84,3%)



(e) FSG (f-score:89,4%)

Figure 5.4: Visual Results of 3 base-layer classifier (Mean Color, Std. Color and color histogram) and FSG for Image 4. Yellow, green and red colors represent TP,FP and FN pixels, respectively

a supervised approach with an unsupervised (or self-supervised) one requires some extra analysis during the design of the training set. The performance of a supervised algorithm is related to the training sample size and statistical properties (as

described in Section 3.2). Based upon various training data configurations, the supervised approach has a varying performance. In order to compare the supervised and unsupervised approaches, we plot the f-scores of the supervised method as a function of number of training sample. For each image, the algorithm randomly selects positive and negative samples according to training sample ratio, w_{tr} ,which starts with 0.05 and increases up to 0.6. This procedure is also repeated 10 times for different random subsets. We increase the training size up to 60% of all the samples in an image, which means that 40% of the segments are not selected as training sample. However, in order to make a fair evaluation, three algorithms have to be tested on the same image. Therefore, for each selected training sample, we pull out it from the training set and classify it by using the rest of the training samples.

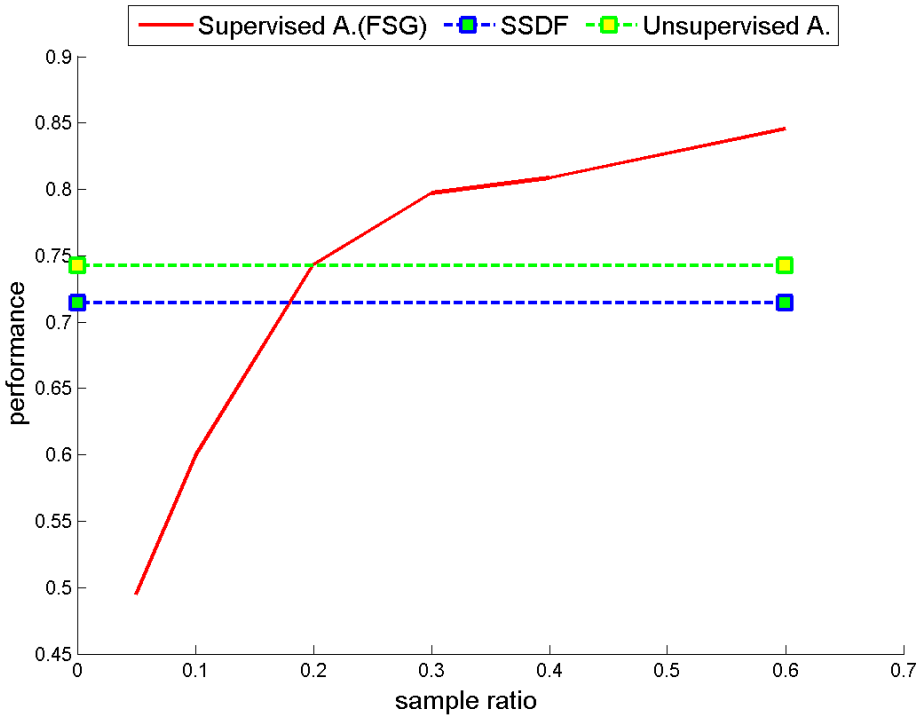
One of the main motivations of these experiments is to observe the effect of automatic sample selection on the classification performance. For this reason, we compare the supervised and SSDF by using the same segmentation parameters and features set, which are described in Section 4.3.1.

In order to evaluate the building detection performance, pixel based scores, which allow us to compare the supervised, unsupervised and SSDF approaches, are preferred instead of segment or object based ones . Moreover, it is crucial to consider both false alarms and missed buildings. Therefore, f-score, which measures the harmonic mean of precision and recall, performances are plotted in the experiments(Figure 5.5 - 5.13).

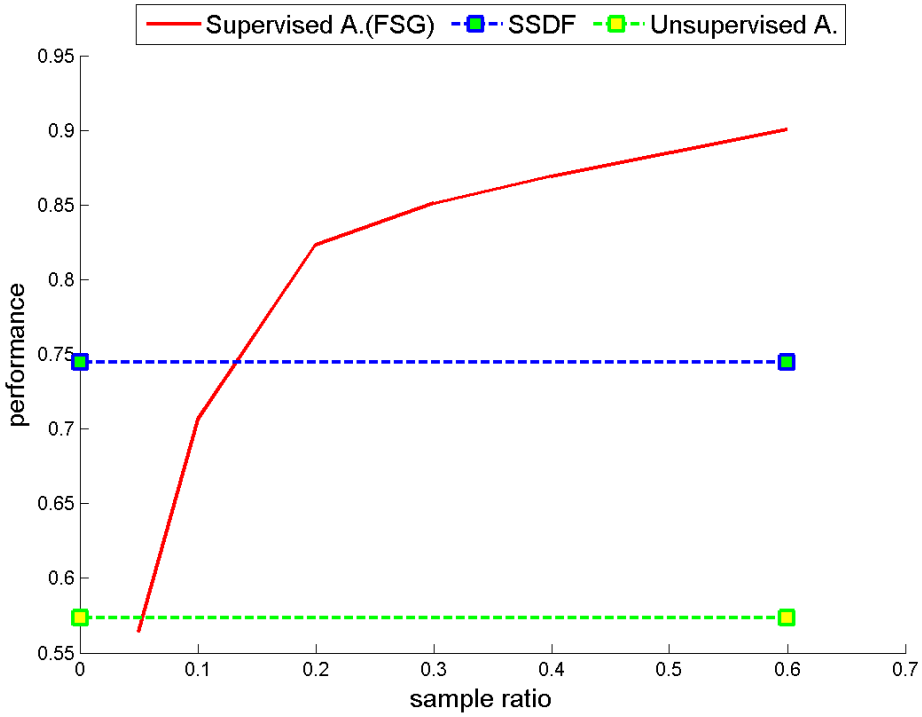
When we analyze the results, given in Table 5.7, we observe that, the average performance of the SSDF is higher than the unsupervised approach. In order to understand the strengths and weaknesses of the SSDF, we inspect some of these results in details. In 10 test images, SSDF shows a higher performance than the unsupervised approach. One of the significant performance differences between SSDF and the unsupervised approach is observed in Image 2 (Figure 5.14). We also analyze the same image for evaluating the training sample selection, in Section 5.2 and we observe that, not only the number of incorrect positive samples, but also the incorrect negative samples is relatively high compared to the average. However, the performance of the SSDF is higher than the unsupervised approach. One of the reasons is that, the shadow detection algorithm misses some of the shadows which belong to building objects.

Table 5.7: Comparision of SSDF and Unsupervised Algorithm

	SSDF			Unsupervised Alg.[2]		
	precision	**recall**	**f-score**	**precision**	**recall**	**f-score**
image 1	82,2%	63,2%	71,5%	77,7%	71,1%	74,3%
image 2	79,1%	70,3%	74,4%	65,9%	50,8%	57,3%
image 3	74,7%	68,3%	71,4%	81,4%	55,8%	66,2%
image 4	80,8%	100,0%	89,4%	83,6%	89,5%	86,5%
image 5	93,5%	96,7%	95,0%	81,3%	93,7%	87,1%
image 6	96,2%	90,6%	93,3%	89,6%	92,8%	91,2%
image 7	91,0%	94,1%	92,5%	93,4%	94,9%	94,2%
image 8	78,7%	87,0%	82,6%	79,6%	82,5%	81,0%
image 9	58,1%	90,2%	70,6%	66,6%	75,3%	70,7%
image 10	60,6%	96,3%	74,3%	76,6%	91,9%	83,6%
image 11	60,9%	90,7%	72,9%	64,8%	78,3%	70,9%
image 12	93,0%	88,8%	90,8%	81,6%	83,8%	82,7%
image 13	86,2%	96,1%	90,9%	77,8%	91,7%	84,2%
image 14	73,1%	89,5%	80,5%	76,5%	91,1%	83,2%
image 15	76,6%	75,1%	75,8%	73,9%	76,3%	75,1%
image 16	51,5%	90,8%	65,7%	72,6%	85,3%	78,4%
image 17	92,4%	80,1%	85,8%	82,1%	90,6%	86,1%
image 18	85,9%	96,3%	90,8%	85,7%	79,6%	82,5%
Average	78,9%	86,9%	81,7%	78,4%	81,9%	79,7%

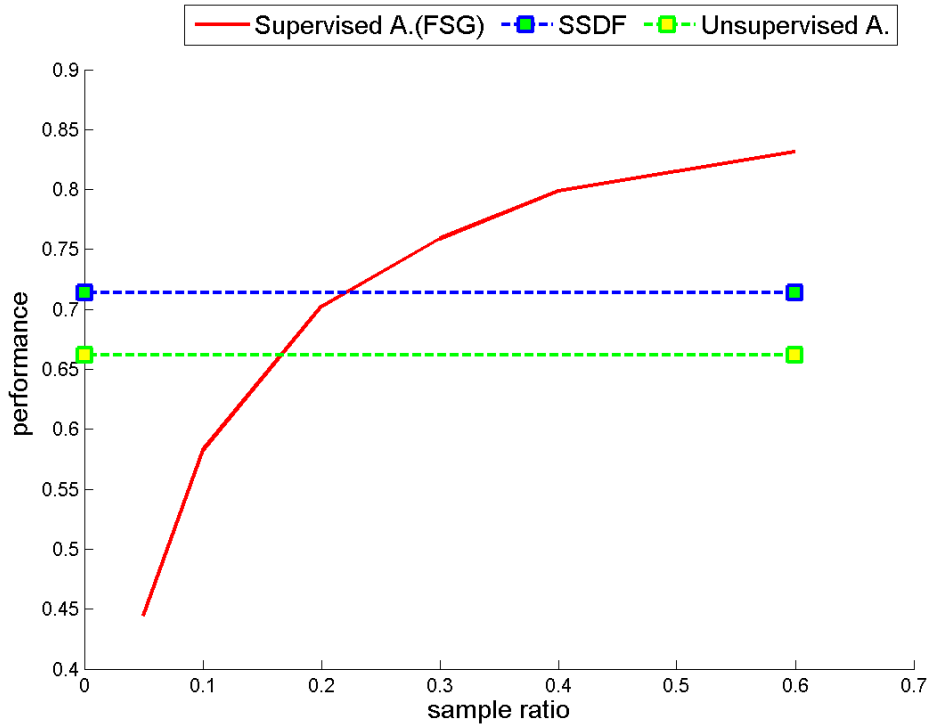


(a) Image 1

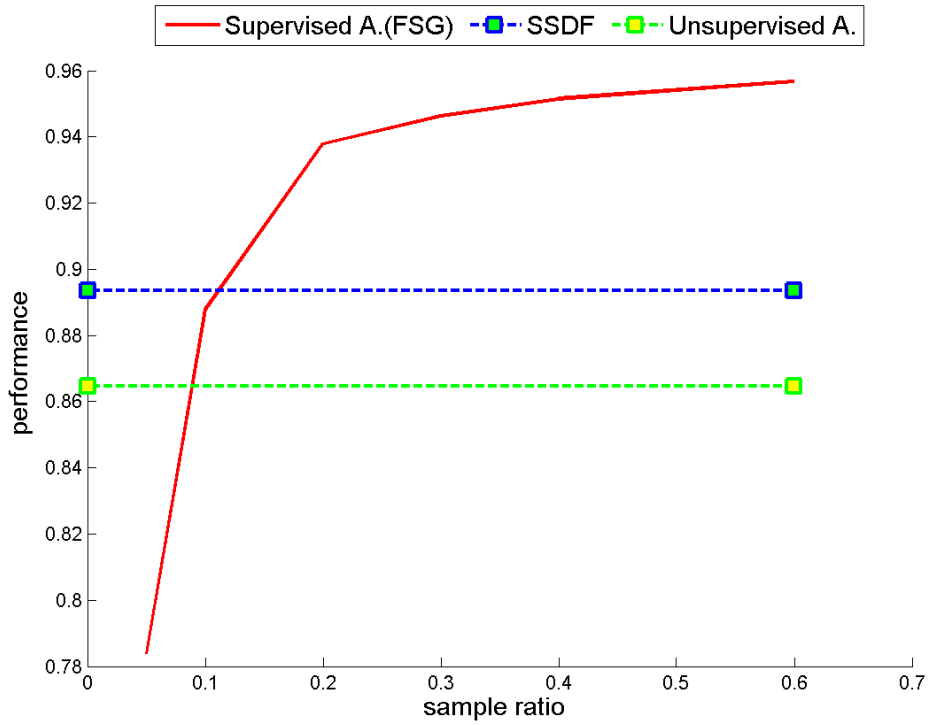


(b) Image 2

Figure 5.5: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 1 and Image 2

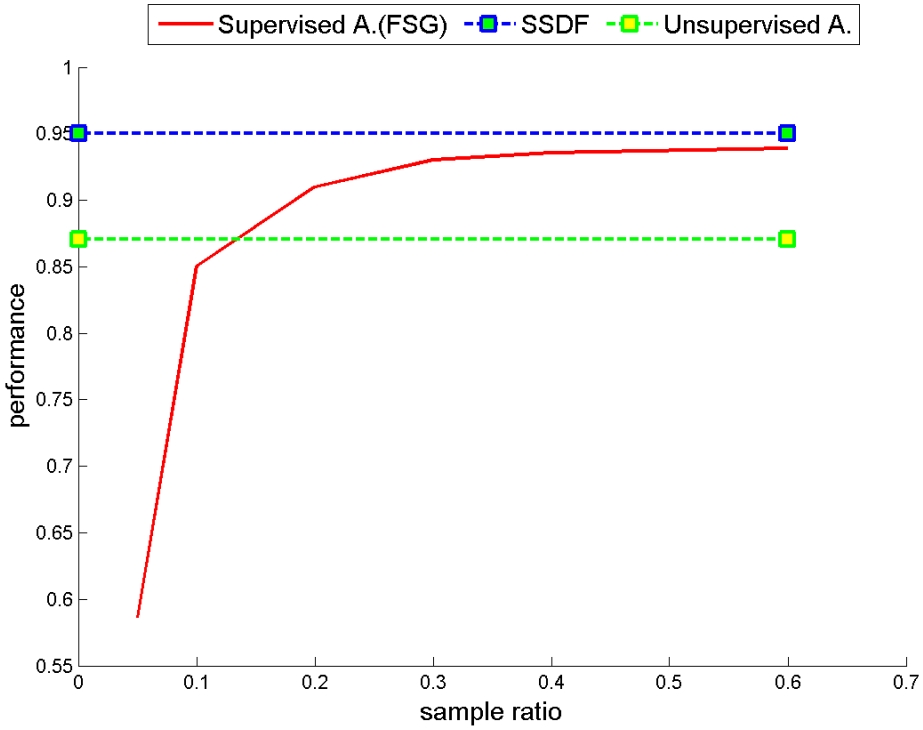


(a) Image 3

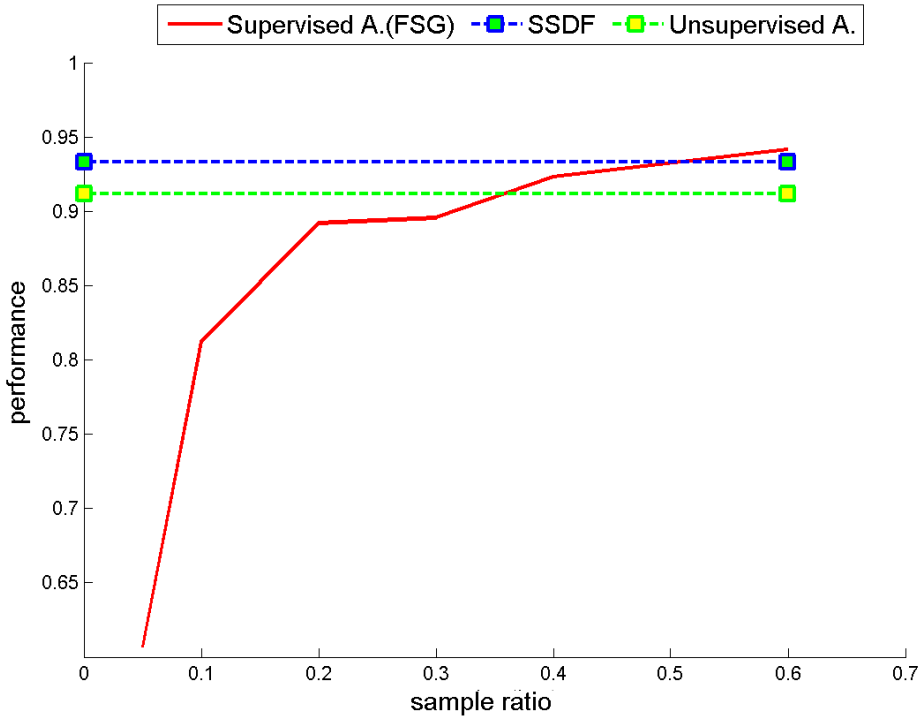


(b) Image 4

Figure 5.6: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 3 and Image 4

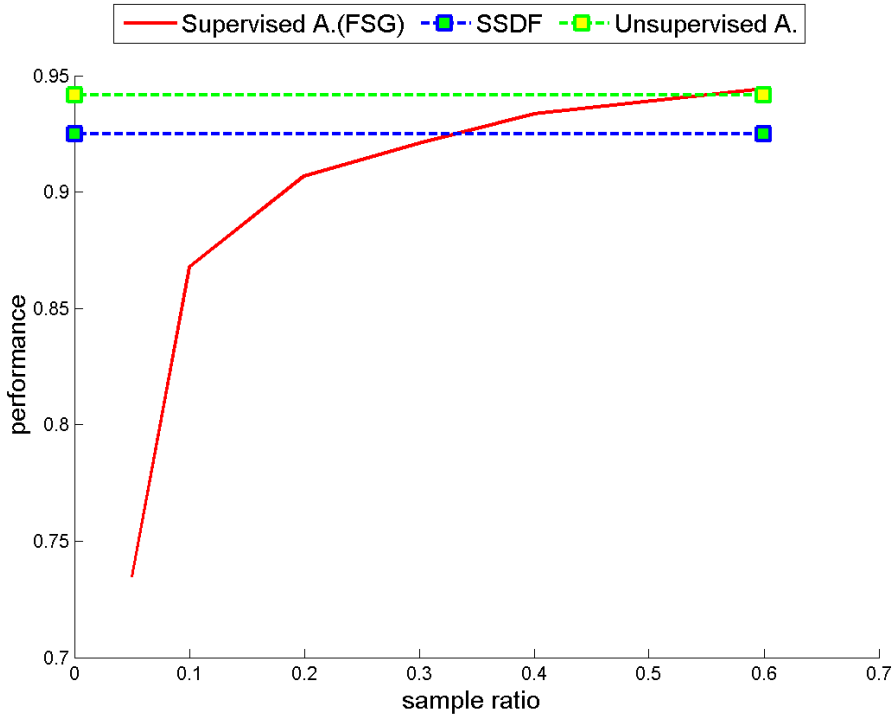


(a) Image 5

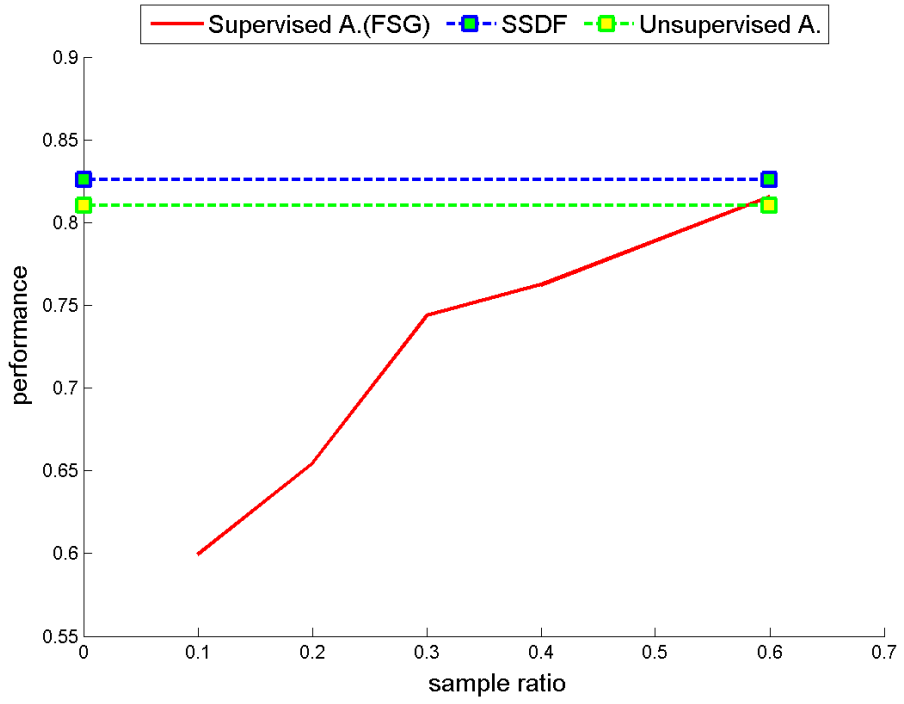


(b) Image 6

Figure 5.7: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 5 and Image 6

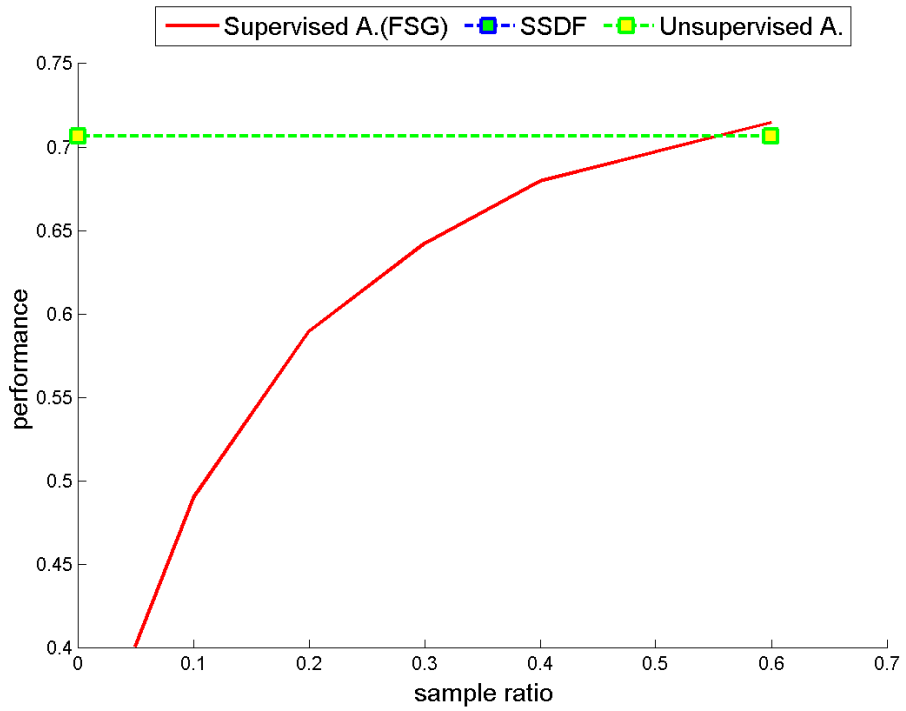


(a) Image 7

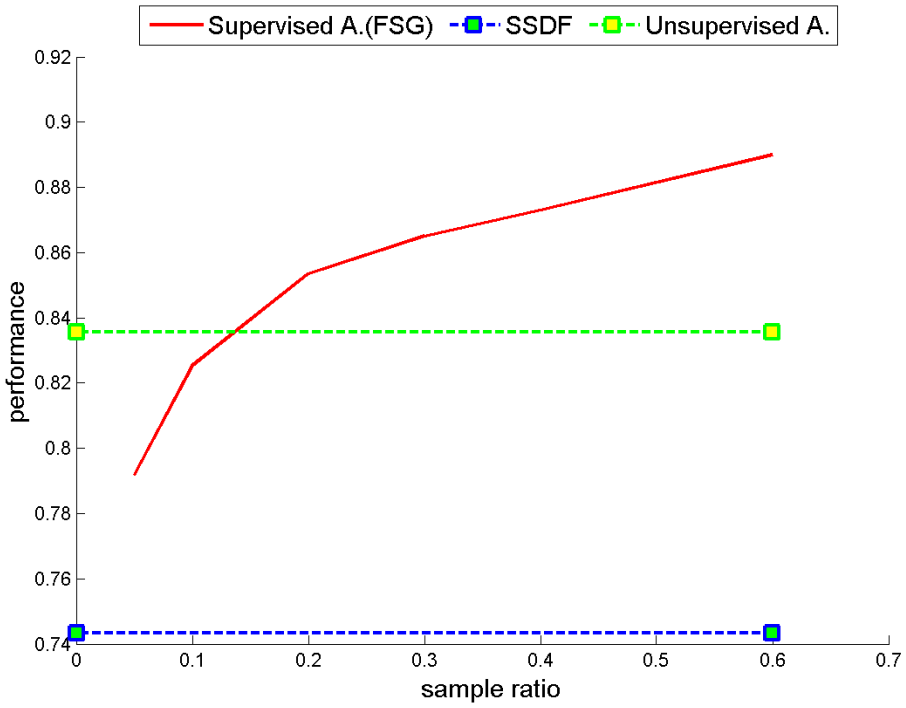


(b) Image 8

Figure 5.8: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 7 and Image 8

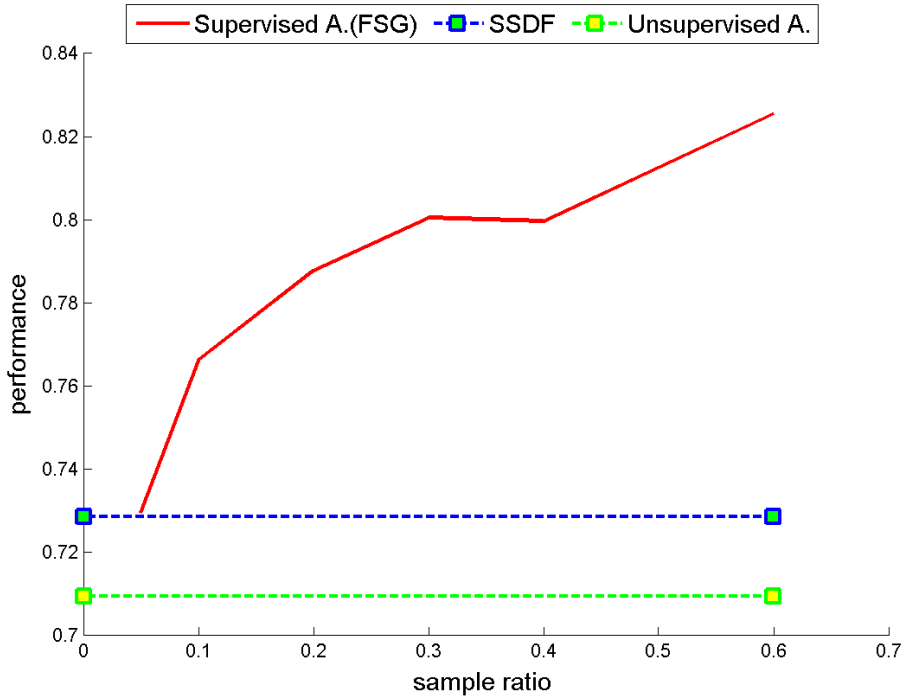


(a) Image 9

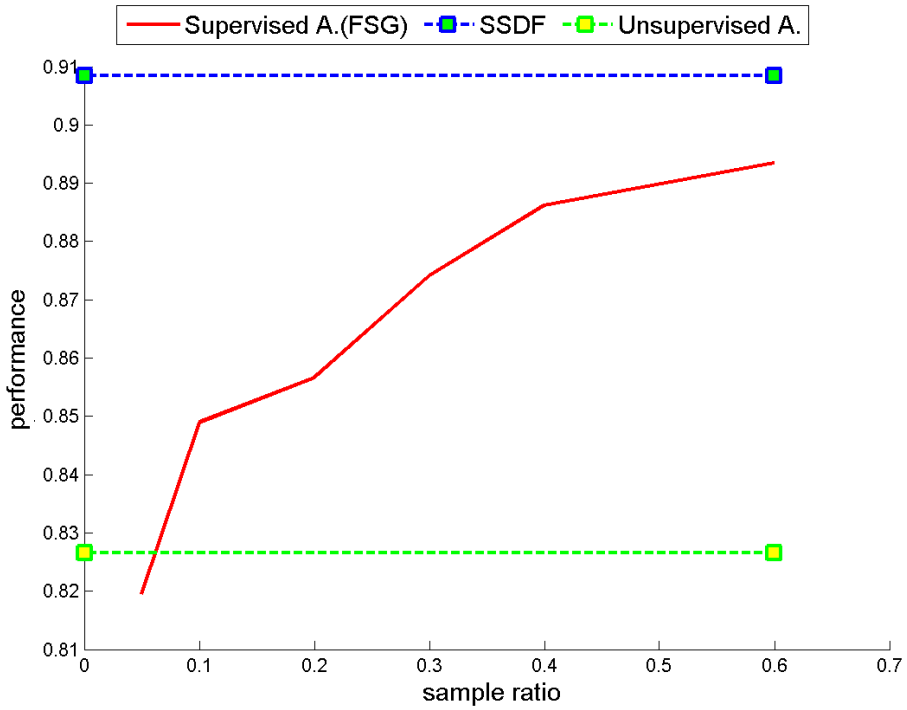


(b) Image 10

Figure 5.9: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 9 and Image 10

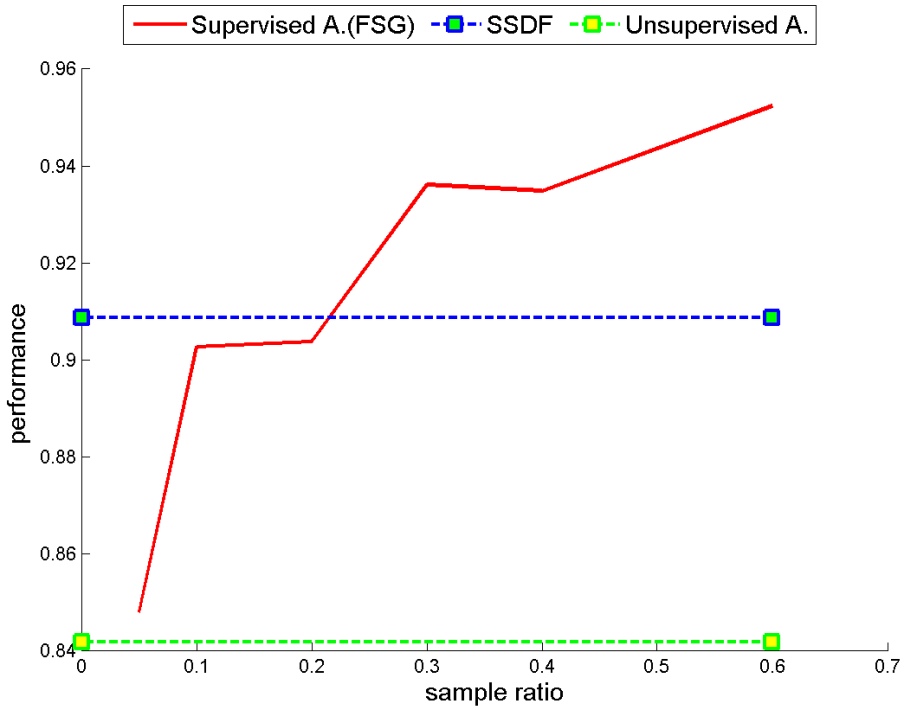


(a) Image 11

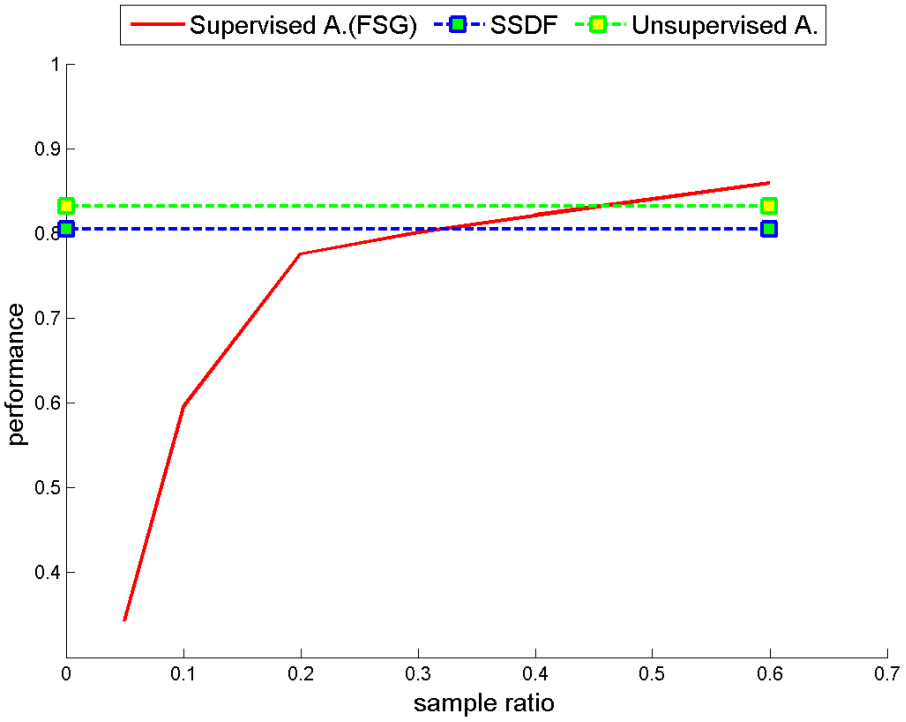


(b) Image 12

Figure 5.10: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 11 and Image 12

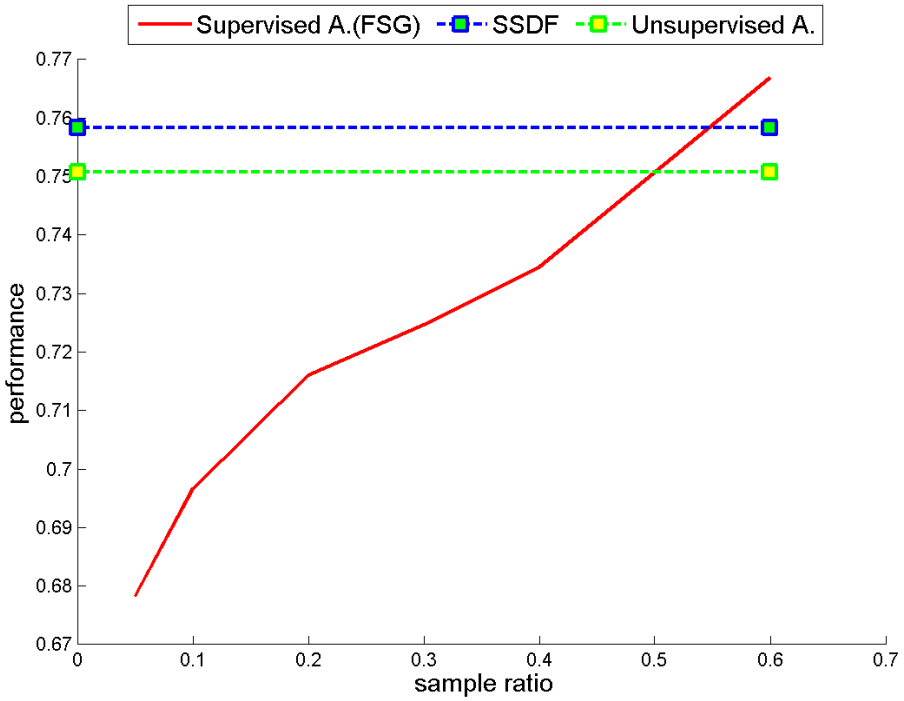


(a) Image 13

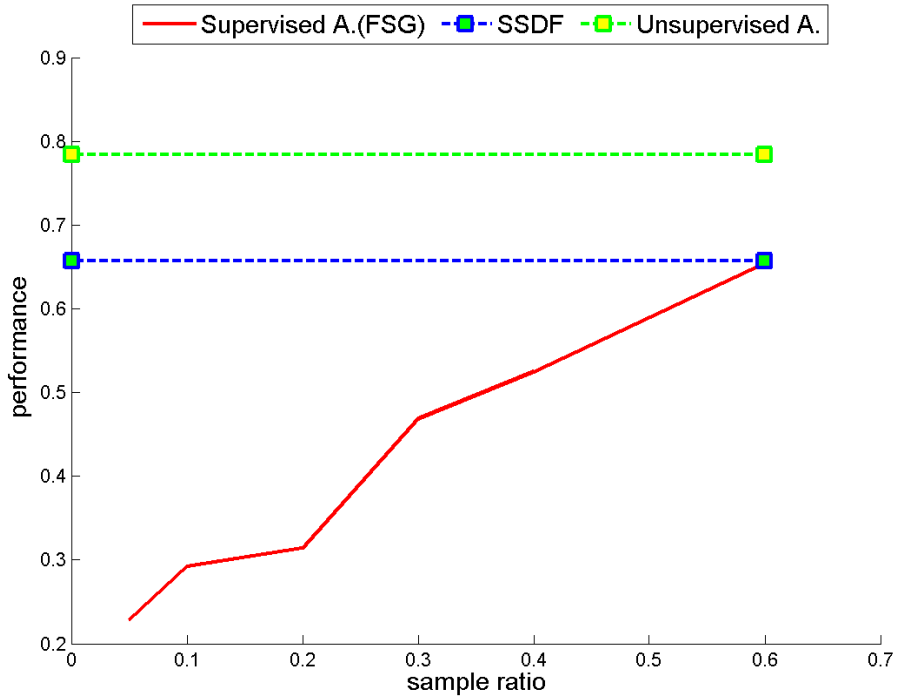


(b) Image 14

Figure 5.11: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 13 and Image 14

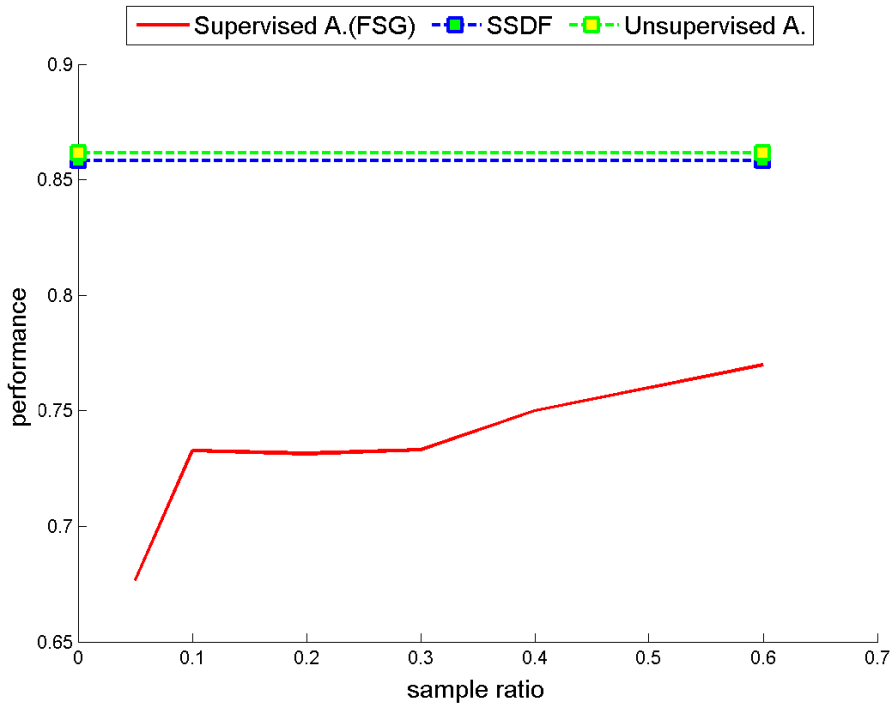


(a) Image 15

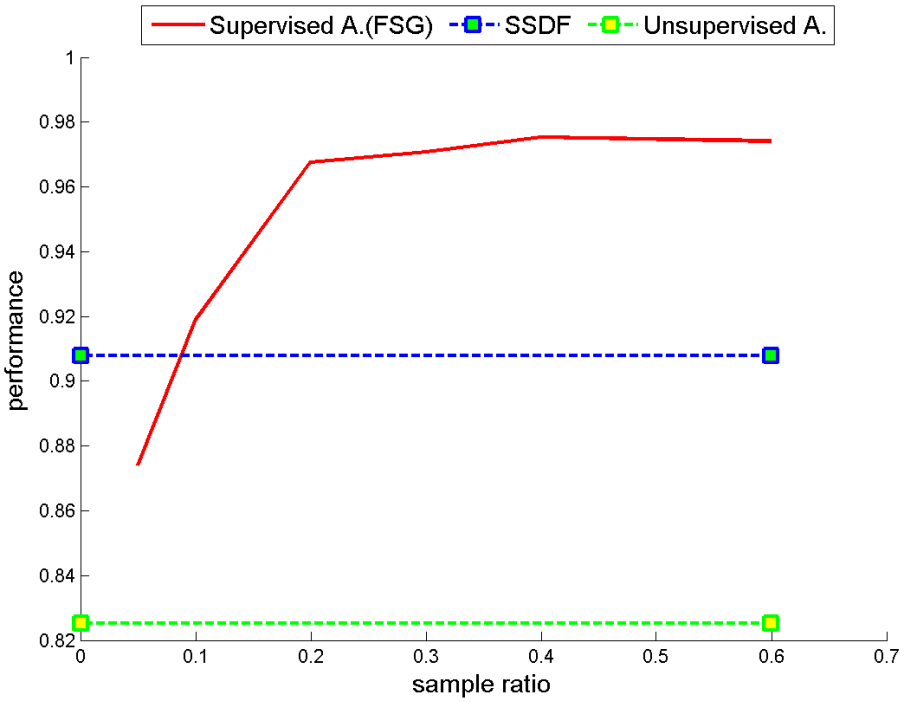


(b) Image 16

Figure 5.12: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 15 and Image 16



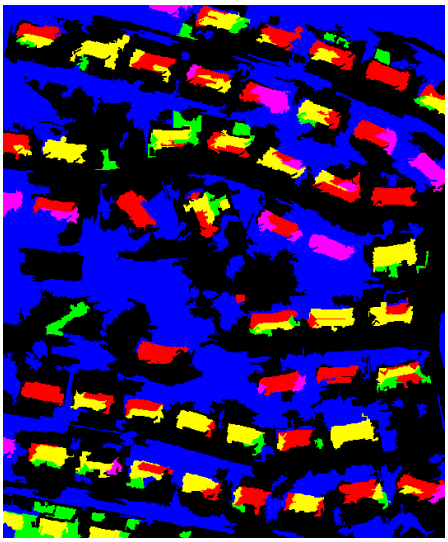
(a) Image 17



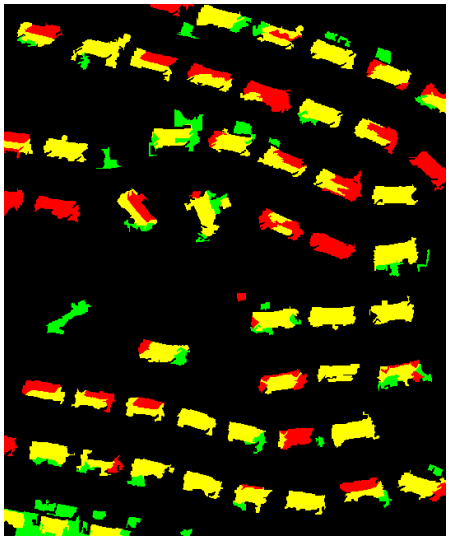
(b) Image 18

Figure 5.13: The comparison of SSDF, Unsupervised Alg.[2], and Supervised FSG[1] using F-score for Image 17 and Image 18

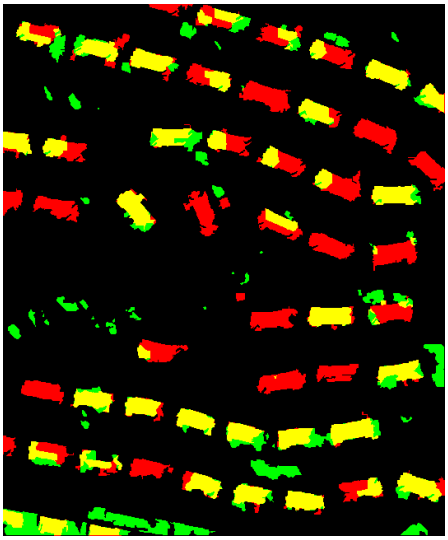
(a) Orginal Image 2



(b) Generated Training Data; Yellow, green, red, blue, pink, black colors represent B_p, N_p, B_u, N_n, B_n, N_u samples, respectively



(c) Result Of SSDF (F-score: 74,4%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

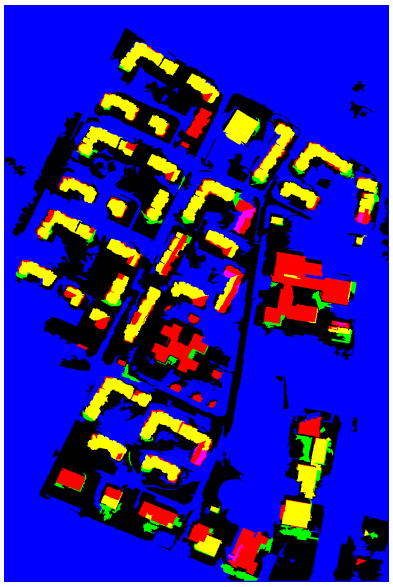


(d) Result Of Unsupervised Algorithm[2] (F-score: 57,3%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

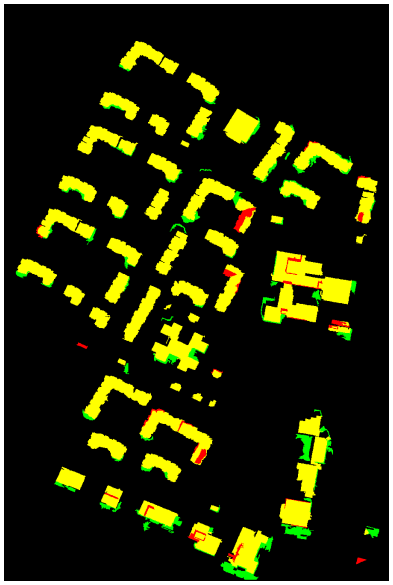
Figure 5.14: The results of SSDF and Unsupervised Building Detection Algorithm [2] for Image 2

Therefore,unsupervised approach has no chance to detect a building if its cast shadow is not correctly detected. On the other hand, in SSDF method, some of these build-

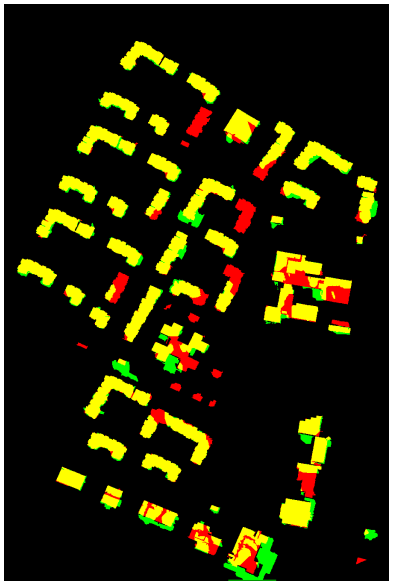
(a) Orginal Image 18



(b) Generated Training Data; Yellow, green, red, blue, pink, black colors represent B_p, N_p, B_u, N_n, B_n, N_u samples, respectively



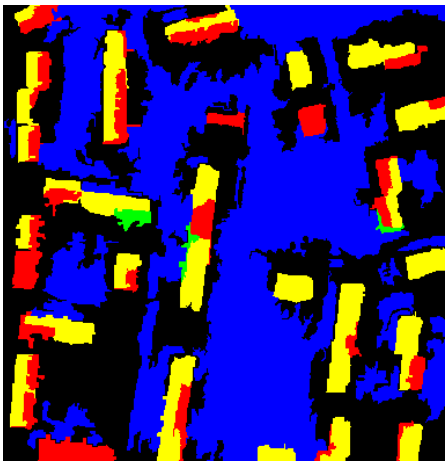
(c) Result Of SSDF (F-score: 92,3%) Yellow, green and red colors represent TP,FP and FN pixels, respectively



(d) Result Of Unsupervised Algorithm [2] (F-score: 82,5%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

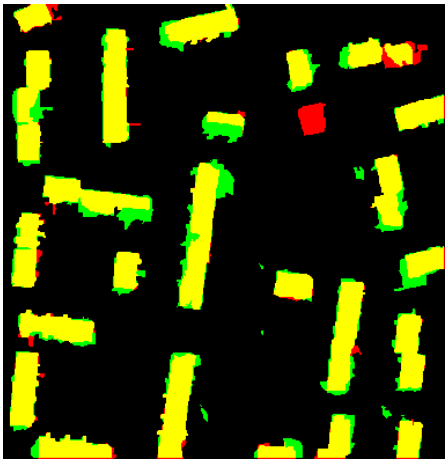
Figure 5.15: The results of SSDF and Unsupervised Building Detection Algorithm [2] for Image 18

(a) Orginal Image 5



(b) Generated Training Data; Yellow, green, red, blue, pink, black colors represent B_p, N_p, B_u, N_n, B_n, N_u samples, respectively

(c) Result Of SSDF (F-score: 95,0%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

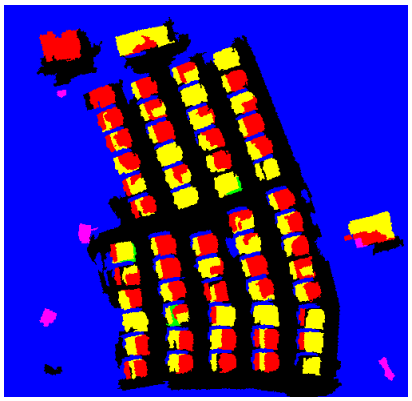


(d) Result Of Unsupervised Algorithm [2] (F-score: 87,0%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

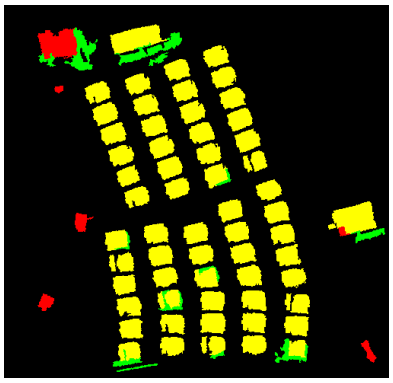
Figure 5.16: The results of SSDF and Unsupervised Building Detection Algorithm [2] for Image 5

ings are selected as negative samples, but the rest of the buildings can be detected during the recognition phase. Moreover, we can also see some false alarms in the result of the unsupervised algorithm, which are generated by the shadow detection algorithm. Fortunately, some of these false alarms are eliminated by using minimum shape threshold in SSDF. A similar situation is observed in image 18 (Figure 5.15).

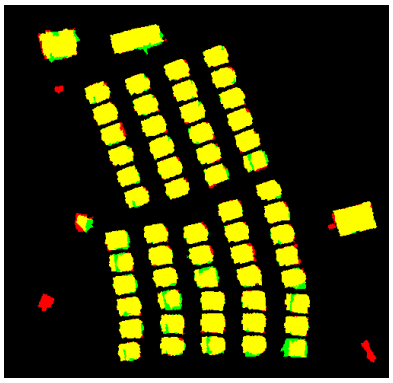
(a) Orginal Image 7



(b) Generated Training Data; Yellow, green, red, blue, pink, black colors represent B_p, N_p, B_u, N_n, B_n, N_u samples, respectively



(c) Result Of SSDF (F-score: 92,5%) Yellow, green and red colors represent TP,FP and FN pixels, respectively



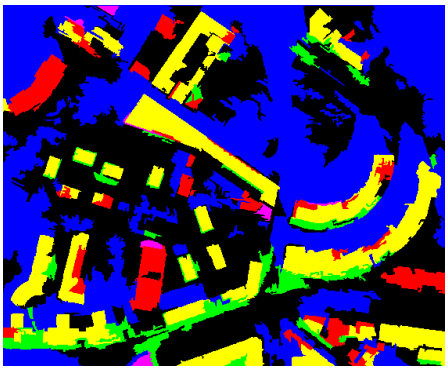
(d) Result Of Unsupervised Algorithm [2] (F-score: 94,2%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

Figure 5.17: The results of SSDF and Unsupervised Building Detection Algorithm [2] for Image 7

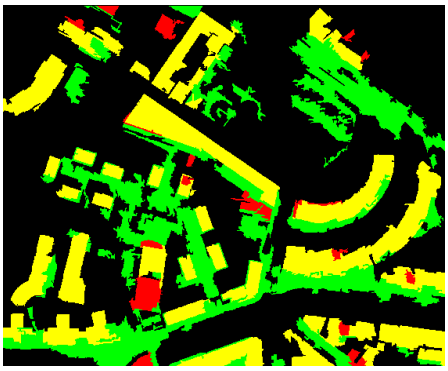
The shadow detection algorithm misses some of the cast shadows of the buildings. However, this time the SSDF algorithm captures rectangular objects and does not use these regions as negative samples. Therefore, our self-supervised algorithm detects these buildings correctly. In the case of all of the shadows are correctly detected, the two approach may have similar results. In three images, the differences of the f-score performances of the SSDF and unsupervised approaches are less than 1% .

During the experiments, we observe some special cases , where the unsupervised al-

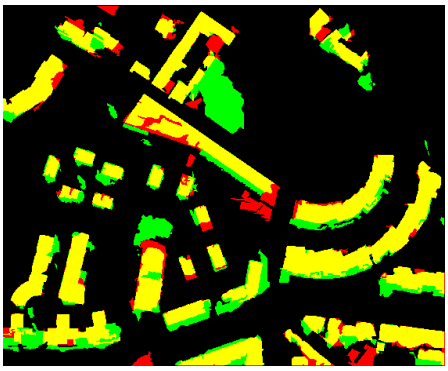
(a) Orginal Image 16



(b) Generated Training Data; Yellow, green, red, blue, pink, black colors represent B_p, N_p, B_u, N_n, B_n, N_u samples, respectively



(c) Result Of SSDF (F-score: 65,7%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

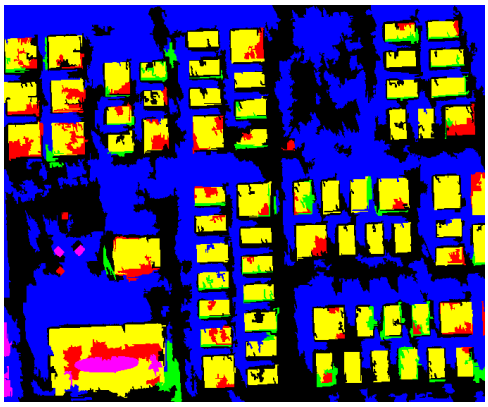


(d) Result Of Unsupervised Algorithm [2] (F-score: 74,8%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

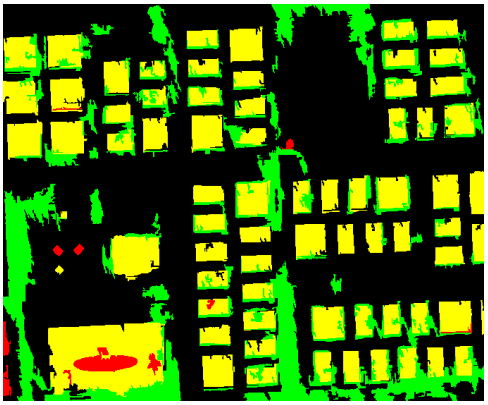
Figure 5.18: The results of SSDF and Unsupervised Building Detection Algorithm [2] for Image 16

gorithm has better performances than the suggested SSDF method. In 5 test images, unsupervised approach has better results than SSDF. As described in Section 5.2, the statistical properties of the selected samples are very crucial for both of the precision and recall values. If the building class has more than one distinct types of buildings and all of the generated building samples are extracted from one type of buildings, the detection of the other type of buildings depends on the dissimilarity of non-building class. We observe typical example for this situation in Figure 5.17. In this test image, there are 37 buildings and 34 of them are typically similar. In the training data

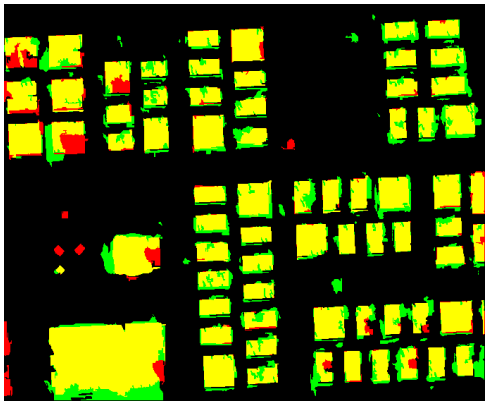
(a) Orginal Image 10



(b) Generated Training Data; Yellow, green, red, blue, pink, black colors represent B_p, N_p, B_u, N_n, B_n, N_u samples, respectively



(c) Result Of SSDF (F-score: 74,3%) Yellow, green and red colors represent TP,FP and FN pixels, respectively



(d) Result Of Unsupervised Algorithm[2] (F-score: 83,6%) Yellow, green and red colors represent TP,FP and FN pixels, respectively

Figure 5.19: The results of SSDF and Unsupervised Building Detection Algorithm [2] for Image 10

generation step, 36 of these buildings are already selected as positive samples and the correctness of the positive samples is 97,6%. However, the building at the upper left is significantly different from the rest of the buildings. Although the shadow detection algorithm finds all of the building cast shadows, sample selection approach eliminates this building during the positive sample selection due to the shape score, R^i_{shp}. Besides, the performance of the SSDF may be affected by incorrect positive samples. Image 16 (Figure 5.18) is one of the examples, which incorrect positive samples create problems. The algorithm adds 353 non-building segments to the training space. How-

ever 61 of them are labeled as positive samples. This means that the selection error for non-building class is 17.3%, which results in a lower precision value. Moreover, when we analyze the positive selected samples from Figure 5.18b, we see that most of the incorrect positive samples are selected from road samples. On the other hand, the amount of negative samples, which are extracted from roads are relatively small. Therefore, the suggested algorithm classify the rest of the roads as buildings.

We observe that, on the average, the supervised algorithm requires more than 30% of the ground truth data for training, in order to catch the performance of the SSDF (Table 5.8). The amount of required training data increases up to 60% for Image 5, Image 8, Image 12, Image 16 and Image 17 in order to catch comparable performances. On the other hand, in Image 10, the performance of the SSDF is significantly lower than both supervised algorithm and unsupervised algorithm. When we analyze the image and output of SSDF (Figure 5.19), we see that most of the buildings and non-buildings are already selected as a training sample. On the other hand, the amount of negative samples , which are selected from the road segments, is rather small. Most of the incorrect positive samples are selected from the road segments. Therefore, the incorrect positive samples create a problem for this image.

Table 5.8: Approximated training ratio for [1] to give similar performance of SSDF.

image 1	20,0%	image 10	1,0%
image 2	15,0%	image 11	5,0%
image 3	22,0%	image 12	60,0%
image 4	12,0%	image 13	25,0%
image 5	60,0%	image 14	35,0%
image 6	50,0%	image 15	55,0%
image 7	30,0%	image 16	60,0%
image 8	60,0%	image 17	60,0%
image 9	55,0%	image 18	10,0%
Average	**35,3%**		

5.5 Sensitivity to Parameters

In this section, we analyze the parameters and their effects on our building detection results. In order to evaluate the effects of the parameters, analogous to the previous section, the pixel based Precision, Recall and F-score metrics are measured. During

the tests, we select minimum, maximum and incrementation steps for the evaluation of each parameter. Precision, Recall and F-score curves are generated from the averaged results of the 18 test images. When the performance curves corresponding each parameter set are analyzed, the following conclusions are derived:

Minimum landscape threshold , T^b_{land}, which is used for positive sample selection, is initialized with 0.08. As shown in Figure 5.21, lower T^b_{land} values result in lower precision rates. When we decrease the values, we allow to use segments, with a small probability of causing a cast shadow, as a positive sample. Therefore, the number of incorrect positive samples increases. On the other hand, it allows us to collect more positive samples and increases the completeness of our training set. As a result, lower T^b_{land} values also result in higher recall rates. When we start to increase the parameter, the precision rates increase incrementally. However, the recall rate is not significantly affected until the T^b_{land} reaches to 0.28 and then begins to decrease. This shows that, for most of the building segment, landscape score, R^i_{land}, is higher than 0.28. We also obtain the best f-score rates for T^b_{land} = 0.28. Thus, we can conclude that, if a good balance between the Precision and Recall metrics is required, the parameter T^b_{land} should be selected around 0.28, for this particular data set.

The rectangularity threshold, T^n_{rect}, significantly affects the performance of SSDF. This parameter is initialized with 0.08 and increased up to 0.60 (Figure 5.20). The algorithm uses this parameter for avoiding to select a segment, which is intersected by a rectangle with a given area ratio, as a negative example. Therefore, for smaller T^n_{rect} values the negative sample selection algorithm becomes more selective. As a result, the number of incorrect negative samples is smaller and recall rates are higher. On the other hand, smaller T^n_{rect} values also reduce the number of correct negative samples, which reduces the precision rates. When we start to increase T^n_{rect}, the number of both correct and incorrect negative sample increases, which causes a decreases in the recall values and increase in precision values. On the other hand, we obtain higher f-score values for higher T^n_{rect}. Therefore, if the recall rate is not very critical, the parameter T^b_{land} could be increased to a value higher than 0.6.

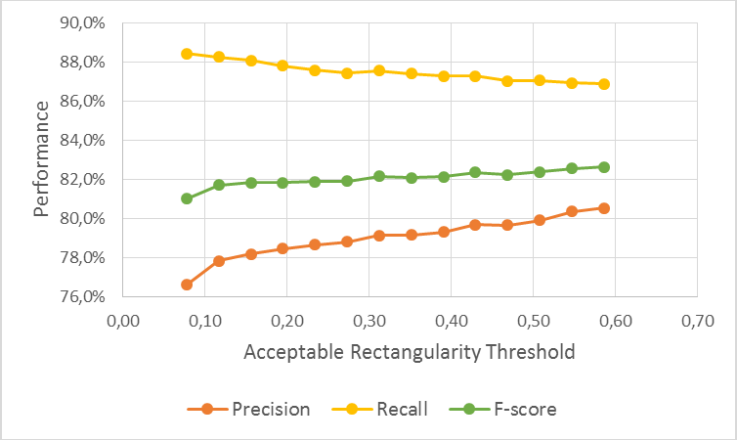


Figure 5.20: Sensitivity to parameter T^n_{rect}

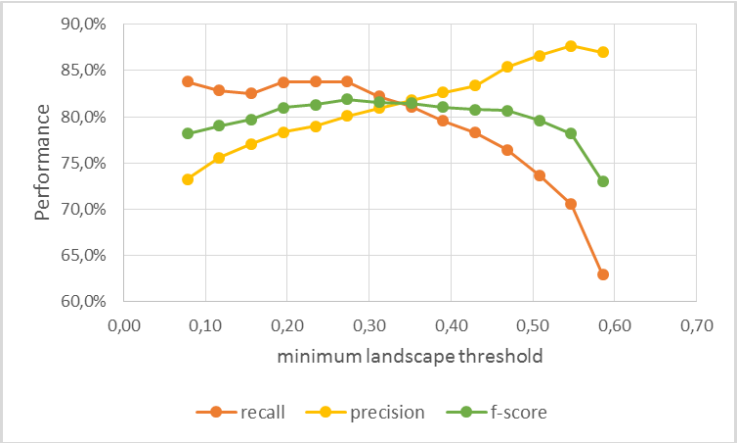


Figure 5.21: Sensitivity to parameter T^b_{land}

5.6 Summary

In this chapter, the performance results of the developed Self-Supervised Decision Fusion (SSDF)system are provided together with some comparisons to the state of the art supervised and unsupervised building detection techniques . In the first set of experiments, the accuracy of the sample selection approach and its effects on classification performances are analyzed. Furthermore, the performance results of the base-layer fuzzy k-NN classfiers using individual features are extracted in order to evaluate the performance of FSG. Finally, the developed approach is compared with two recent studies. We conclude that self-supervision is a successful solution which allow us to use supervised classifiers for new test images without any human interaction.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we combine the unsupervised and supervised learning paradigms to develop a self-trained building detection method for remote sensing images. The proposed model automatically generates the training data from each image layout, independently. This training data is then used to detect the rest of the buildings in the same image. It is quite intuitive that the class conditional densities for building and non-building objects, estimated from a single statistically stable image layout has relatively small within class variances compared to the within class variance of the density functions, obtained from the training set, extracted from a wide range of images. Therefore, rather than extracting the training data from a large variety of images, it is wise to perform the training and testing processes at the same image. This fact allows us to estimate more discriminative class conditional densities of building and non-building objects, compared to the densities which are estimated from large set of images.

Additionally, employing a decision fusion method to the outputs of multiple classifiers enables us to represent a verity of the discriminative properties of the object in different feature spaces. Fusion of the decisions of the multiple classifiers boosts the performance of the individual classifiers, employed in the base layer. The most crucial step in the suggested unsupervised methodology for training set extraction, is to select building and non-building regions with a sufficiently high confidence, so that the training data does not misinform the classifiers. This requires the identification of some invariant properties of the buildings, which remain unchanged across the images

in the dataset. The strongest invariant of the buildings are the cast shadows adjacent to each building. Also, it is well known that most of the buildings appear in a scene in the form of rectangles or union of multiple rectangles with corners of 90 degrees. This information is also successfully employed to extract the training data. Finally, absence of vegetation is used as a supplementary information to increase correctness of the training data.

The proposed method together with its pros and cons can be summarized as follows:

- The power of the suggested method comes from the information extraction step, which enables us to extract the training data from each image layout. This step includes the preprocessing and generating several maps which are used for automated training data generation. This step begins with the fusion of the panchromatic and multispectral image to generate pan-sharped image. After the pan-sharped image is generated, the algorithm extracts the vegetation and shadow masks by using the spectral properties of the pixels. The algorithm uses the sun direction and models the spatial arrangement between the shadow and the related object and prune non-building objects which cause shadows. The algorithm, also, detects the possible rectangular regions. Finally, mean-shift segmentation algorithm is used to obtain over-segmented regions. One should note that the masks obtained at the output of shadow, rectangularity and vegetation detection are not error free. The misses or false alarms in these maps may ruin the extracted training data, resulting in false training.

- In the second step, the algorithm analyzes each segment by using the extracted information and select positive and negative samples. The algorithm uses the detected rectangles to make sure that any non-building sample is selected from these regions. Selecting the number of positive and negative samples requires sufficiently high confidence values. Also, selecting the number of samples from each class needs careful statistical analyses prior to the training process

- The last step of the algorithm extracts the features from the over-segments and trains the two-layer decision fusion classifier (FSG) by using automatically selected samples. At the base-layer of FSG, several classifiers are trained by different color, texture and shape features. Then, at the meta-layer, the clas-

sifier decisions are fused to construct the feature vectors of the fusion space by concatenating the decision vectors of the base-layer classifiers. The aggregated decision vectors are fed to a meta-layer classifier to generate the final decisions on the candidate buildings. Although FSG is a very powerful decision fusion architecture, selection of the feature spaces at the input of the base layer classifiers requires some care. When designing the base layer classifiers, one should pay attention to select the feature spaces, each of which share to extract a specific characteristic of the object class which satisfies the diversity condition of the ensemble learning methodology.

The proposed building detection algorithm is tested in 18 different scenes. In the experiments, we first observe that FSG improves the f-score performance of the individual base-layer classifiers. Also, we compare the proposed approach with our two state of the art algorithms. We also observe that, in average, the supervised algorithm needs to use more than 30% of the ground truth data for training in order to catch our proposed model. Moreover, the experiments show that, the average performance of the SSDF is higher than the unsupervised approach.

We also evaluate the automated training sample selection approach and realize some important factors, which are very critical for a self-supervised approach. The performance of the SSDF method depends on several important factors. First of all, the correctness of the positive samples has a major effect on the precision of the algorithm. In our tests, we see that the images with lower correctness values for positive samples, also have lower precision scores at the output of the SSDF algorithm. Moreover, the amount of correctly selected negative samples has a positive effect on precision, because it tolerates the incorrect positive samples. The recall of the algorithm also depends on several parameters. One of them is the completeness of the positive samples. Also, the number of selected negative samples becomes an important issue, specifically when the number of incorrect negative samples is relatively high compared to the number of correct negative samples.

Our future direction of research is to focus on training sample extraction step to increase the quality of the training data set. We plan to achieve this task by optimizing and validating several parameters of the suggested SSDF method, including the feature

space design at the base layer. Besides, the Multiple Instance Learning [80] paradigm , where the data is assumed to have some ambiguity, can be used for self-supervised building detection problem. Another important issue remaining as the future work, is to generalize the suggested framework to a more generic class of objects in the remote sensing domain.

REFERENCES

[1] Caglar Senaras, Mete Ozay, and Fatos Yarman Vural. Building detection with decision fusion. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 6(3):1295–1304, 2013.

[2] Ali Ozgun Ok, Caglar Senaras, and Baris Yuksel. Automated detection of arbitrarily shaped buildings in complex environments from monocular vhr optical satellite imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(3):1701–1717, 2013.

[3] Swarnajyoti Patra and Lorenzo Bruzzone. A fast cluster-assumption based active-learning technique for classification of remote sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(5):1617–1626, 2011.

[4] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):770–787, 2010.

[5] Helmut Mayer. Automatic object extraction from aerial imagery—a survey focusing on buildings. *Computer Vision and Image Understanding*, 74(2):138–149, May 1999.

[6] Emmanuel Baltsavias. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3-4):129–151, Jan 2004.

[7] Dilek Koc San. *Approaches For Automatic Urban Building Extraction And Updating From High Resolution Satellite Imagery*. PhD thesis, Middle East Technical University, 2009.

[8] Cem Ünsalan and Kim L. Boyer. *Review on Building and Road Detection*, pages 139–144. Advances in Computer Vision and Pattern Recognition. Springer London, 2011.

[9] David L Hall and James Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[10] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

[11] Hui-Ying Li, Sheng-Bo Chen, Zhi Wang, and Wen-Hui Li. Fusion of lidar data and orthoimage for automatic building reconstruction. In *Proc. IEEE IGARSS*, pages 1194–1197, Honolulu, HI, July 2010.

[12] Franz Rottensteiner, John Trinder, Simon Clode, and Kurt Kubik. Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis. *ISPRS J. Photogramm. Remote Sens.*, 62(2):135–149, June 2007.

[13] Liwei Li, Bing Zhang, and Yuanfeng Wu. Fusing spectral and texture information for collapsed buildings detection in airborne image. In *Proc. IEEE IGARSS*, pages 186–189, Munich, Germany, July 2012.

[14] R Hansch and Olaf Hellwich. Random forests for building detection in polarimetric sar data. In *Proc. IEEE IGARSS*, pages 460–463, Honolulu, HI, July 2010.

[15] Claudio Rosito Jung and Rodrigo Schramm. Rectangle detection based on a windowed hough transform. In *Proc. 17th Brazilian Symposium on Computer Graphics and Image Processing*, pages 113–120. IEEE, 2004.

[16] S.D. Mayunga, Y. Zhang, and D.J. Coleman. Semi-automatic building extraction utilizing quickbird imagery. In *Proc. ISPRS Workshop CMRT*, pages 131–136, 2005.

[17] Parvaneh Saeedi and Harold Zwick. Automatic building detection in aerial and satellite images. In *Int. Conf. Control, Automation, Robotics and Vision, (ICARCV 2008)*, volume 1, pages 623–629, Hanoi, Vietnam, 2008. IEEE.

[18] Xiaoying Jin and Curt H Davis. Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information. *EURASIP J. Appl. Signal Process.*, 2005(14):2196–2206, 2005.

[19] Huseyin Gokhan Akcay and Selim Aksoy. Building detection using directional spatial constraints. In *Proc. IEEE IGARSS*, pages 1932 –1935, July 2010.

[20] Orsan Aytekin, Arzu Erener, Ilkay Ulusoy, and Sebnem Duzgun. Unsupervised building detection in complex urban environments from multispectral satellite imagery. *International Journal of Remote Sensing*, 33(7):2152–2177, 2012.

[21] Melissa Cote and Parvaneh Saeedi. Automatic rooftop extraction in nadir aerial imagery of suburban regions using corners and variational level set evolution. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(1):313–328, Jan 2013.

[22] Xin Huang and Liangpei Zhang. Morphological building/shadow index for building extraction from high-resolution imagery over urban areas. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(1):161–172, Feb 2012.

[23] Beril Sirmacek and Cem Unsalan. A probabilistic framework to detect buildings in aerial and satellite images. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(1):211–221, January 2011.

[24] Xian Sun, Kun Fu, Hui Long, Yanfeng Hu, Lun Cai, and Hongqi Wang. Contextual models for automatic building extraction in high resolution remote sensing image using object-based boosting method. In *Proc. 2008 IEEE Int. Geoscience and Remote Sensing Symp. (IGARSS)*, volume 2, pages 437–440, Boston, MA, USA, July 2008.

[25] Keping Chen and Russell Blong. Extracting building features from high resolution aerial imagery for natural hazards risk assessment. In *Proc. IEEE IGARSS*, volume 4, pages 2039–2041. IEEE, Nov. 2002.

[26] Jordi Inglada. Automatic recognition of man-made objects in high resolution optical remote sensing images by svm classification of geometric image features. *ISPRS J. Photogramm. Remote Sens.*, 62(3):236–248, Aug. 2007.

[27] Mathieu Fauvel, Jocelyn Chanussot, and Jón Atli Benediktsson. Decision fusion for the classification of urban remote sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(10):2828–2838, 2006.

[28] M. Turker and D. K. San. Building detection from pan-sharpened ikonos imagery through support vector machines classification. In *Proc. of ISPRS Technical Com. VIII Symposium*, pages 841–846, Kyoto, Japan, 2010.

[29] Ana-Maria Cretu and Pierre Payeur. Building detection in aerial images based on watershed and visual attention feature descriptors. In *Computer and Robot Vision (CRV), 2013 International Conference on*, pages 265–272. IEEE, 2013.

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

[31] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, May 2002.

[32] Mete Ozay and Fatos Yarman Vural. A new fuzzy stacked generalization technique for deep learning and analysis of its performance. *CoRR*, abs/1204.0171, 2012.

[33] Katia Stankov and Dong-Chen He. Building detection in very high spatial resolution multispectral images using the hit-or-miss transform. *IEEE Geoscience and Remote Sensing Letters*, 10(1):86–90, January 2013.

[34] Sébastien Lefèvre and Jonathan Weber. Automatic building extraction in vhr images using advanced morphological operators. In *Urban Remote Sensing Joint Event, 2007*, pages 1–5. IEEE, 2007.

[35] D. Tuia, E. Pasolli, and W.J. Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242, Sep 2011.

[36] Lorenzo Bruzzone and Mattia Marconcini. Toward the automatic updating of land-cover maps by a domain-adaptation SVM classifier and a circular validation strategy. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(4):1108–1122, April 2009.

[37] Suju Rajan, Joydeep Ghosh, and Melba M Crawford. An active learning approach to hyperspectral data classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(4):1231–1242, 2008.

[38] Roman Katz, Juan Nieto, Eduardo Nebot, and Bertrand Douillard. Track-based self-supervised classification of dynamic obstacles. *Autonomous Robots*, 29(2):219–233, 2010.

[39] Christopher A Brooks and Karl D Iagnemma. Self-supervised classification for planetary rover terrain sensing. In *Aerospace Conference, 2007 IEEE*, pages 1–9. IEEE, 2007.

[40] Dirk Schulz. A probabilistic exemplar approach to combine laser and vision for person tracking. In *Robotics: science and systems*, 2006.

[41] Hendrik Dahlkamp, Adrian Kaehler, David Stavens, Sebastian Thrun, and Gary R Bradski. Self-supervised monocular road detection in desert terrain. In *Robotics: science and systems*, volume 38, 2006.

[42] Young-Woo Seo, Nathan D Ratliff, and Chris Urmson. Self-supervised aerial image analysis for extracting parking lot structure. In *IJCAI*, pages 1837–1842, 2009.

[43] Pete Doucette, Peggy Agouris, and Anthony Stefanidis. Automated road extraction from high resolution multispectral imagery. *Photogrammetric Engineering & Remote Sensing*, 70(12):1405–1416, 2004.

[44] Aaron K Shackelford and Curt H Davis. A self-supervised approach for fully automated urban land cover classification of high resolution satellite imagery. In *Proc. 3rd Int. Symp. Remote Sens. Data Fusion Over Urban Areas—Urban*, pages 14–16.

[45] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[46] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[47] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, 149300, 2010.

[48] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):929–944, 2007.

[49] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, Jan. 1975.

[50] R Grompone Von Gioi, Jeremie Jakubowicz, J-M Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):722–732, 2010.

[51] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[52] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[53] Zuyang Kou, Zhenwei Shi, and Liu Liu. Airport detection based on line segment detector. In *Computer Vision in Remote Sensing (CVRS), 2012 International Conference on*, pages 72–77. IEEE, 2012.

[54] Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J Flynn, Horst Bunke, Dmitry B Goldgof, Kevin Bowyer, David W Eggert, Andrew Fitzgibbon, and Robert B Fisher. An experimental comparison of range image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):673–689, 1996.

[55] Xiaoyi Jiang, Cyril Marti, Christophe Irniger, and Horst Bunke. Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing*, 2006:209–209, 2006.

[56] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[57] Selim Aksoy, Ismet Zeki Yalniz, and Kadim Tasdemir. Automatic detection and segmentation of orchards using very high resolution imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(8):3117–3131, 2012.

[58] Encyclopedia Structural engineering. `http://www.statemaster.com/encyclopedia/Structural-engineering/`. [Online; accessed 28-Agust-2013].

[59] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, 8:985–1005, 2007.

[60] Joo Seuk Kim and Clayton Scott. Robust kernel density estimation. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3381–3384. IEEE, 2008.

[61] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification. *New York: John Wiley, Section*, 10:l, 2001.

[62] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1 edition, April 1986.

[63] Peter Hall. Integrated square error properties of kernel estimators of regression functions. *The Annals of Statistics*, 12(1):241–260, 1984.

[64] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*, (4):580–585, 1985.

[65] JG Liu. Smoothing filter-based intensity modulation: a spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18):3461–3472, 2000.

[66] VK Shettigara. A generalized component substitution technique for spatial enhancement of multispectral images using a higher resolution data set. *Photogrammetric Engineering and remote sensing*, 58(5):561–567, 1992.

[67] Te-Ming Tu, Ping S Huang, Chung-Ling Hung, and Chien-Ping Chang. A fast intensity-hue-saturation fusion technique with spectral adjustment for ikonos imagery. *Geoscience and Remote Sensing Letters, IEEE*, 1(4):309–312, 2004.

[68] Selcuk Sumengen, Caglar Senaras, and Ahmet Erdem. Yuksek cozunurluklu uydu goruntuleri icin pankromatik keskinlestirme yontemi. In *SAVTEK, Ulusal Savunma Teknolojileri Kongresi*, page 80, 2012.

[69] Israa Amro, Javier Mateos, Miguel Vega, Rafael Molina, and Aggelos K Katsaggelos. A survey of classical methods and new trends in pansharpening of multispectral images. *EURASIP Journal on Advances in Signal Processing*, 2011(1):1–22, 2011.

[70] René R Colditz, Thilo Wehrmann, Martin Bachmann, Klaus Steinnocher, Michael Schmidt, Günter Strunz, and Stefan Dech. Influence of image fusion approaches on classification accuracy: a case study. *International Journal of Remote Sensing*, 27(15):3311–3335, 2006.

[71] David M Gates. *Biophysical ecology*. DoverPublications. com, 2003.

[72] Compton J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sens. Environ.*, 8(2):127–150, May 1979.

[73] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[74] Mustafa Teke, Emre Baseski, Ali Ozgun Ok, Baris Yuksel, and Caglar Senaras. Multi-spectral false color shadow detection. In *Proc. 2011 ISPRS conference on Photogrammetric image analysis (PIA'11)*, pages 109–119, Berlin, Heidelberg, 2011. Springer-Verlag.

[75] Isabelle Bloch. Fuzzy relative position between objects in image processing: a morphological approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(7):657–664, 1999.

[76] Jack E Bresenham. Incremental line compaction. *The Computer Journal*, 25(1):116–120, 1982.

[77] Ping Zhong and Runsheng Wang. A multiple conditional random fields ensemble model for urban area detection in remote sensing optical images. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(12):3978 –3988, Dec. 2007.

[78] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-15(4):580–585, July/Aug. 1985.

[79] Mete Ozay and Fatos Yarman Vural. A new decision fusion technique for image classification. In *Proc. IEEE the International Conference on Image Processing, (ICIP 2009)*, pages 2189–2192, Cairo, Egypt, Nov. 2009.

[80] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Şenaras, Çağlar
Nationality: Turkish (TC)
Date and Place of Birth: 02.08.1980, Denizli
Marital Status: Married

EDUCATION

Degree	Institution	Year
M.S.	Department of Civil Engineering, METU, Ankara, Turkey	2004-2007
B.S.	Department of Civil Engineering, METU, Ankara, Turkey	1999-2004

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2007-2013	HAVELSAN AŞ.	Software Engineer
2004-2007	Informatics Institute, METU, Ankara, Turkey	Research Assistant

PUBLICATIONS

1. Çağlar Şenaras, Mete Ozay, Fatoş T. Yarman Vural, Building Detection with Decision Fusion, in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, no.99, pp. 1-10, March 2013.

2. Ali Özgün Ok, Çağlar Şenaras, Barış Yüksel, Automated detection of arbitrarily shaped buildings in complex environments from monocular VHR optical satellite

imagery, in IEEE Transactions on Geoscience and Remote Sensing, 2013.

3. Çağlar Şenaras, Baris Yuksel, Mete Ozay, Fatos T. Yarman Vural, Automatic Building Detection with Feature Space Fusion using Ensemble Learning, IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2012.

4. Ali Özgün Ok, Çağlar Şenaras, Barış Yüksel, Exploiting shadow evidence and iterative graph-cuts for efficient detection of buildings in complex environments, ISPRS Hannover Workshop, 2013

5. Mustafa Teke, Emre Başeski, Ali Özgün Ok, Barış Yüksel, Çağlar Şenaras ,Multi-spectral false color shadow detection, Photogrammetric Image Analysis, 2011

6. Emre Başeski, Çağlar Şenaras, Cloud Detection in High-Resolution Satellite Images, IEEE Signal Processing, Communication and Applications, 2013.

7. Baris Yuksel, Çağlar Şenaras, Mete Ozay, Fatos T. Yarman Vural, Automatic Building Detection From Satellite Images Using Stacked Generalization Architecture, IEEE Signal Processing, Communication and Applications, 2011.

8. Çağlar Şenaras, Selçuk Sümengen, Ahmet Erdem, Uzaktan Algılama Uygulamaları için Uzamsal Veri Yönetim Sistemi, SAVTEK, Ulusal Savunma Teknolojileri Kongresi, 2012

9. Selçuk Sümengen, Çağlar Şenaras, Ahmet Erdem, Yüksek Çözünürlüklü Uydu Görüntüleri için Pankromatik Keskinleştirme Yöntemi, SAVTEK, Ulusal Savunma Teknolojileri Kongresi, 2012