ROAD EXTRACTION FROM SATELLITE IMAGES

Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

MEGHANA KANDRU

Under the Guidance of

A.JITENDRA, M.Tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

V.R SIDDHARTHA ENGINEERING COLLEGE

Autonomous and Approved by AICTE

Affiliated to Jawaharlal Nehru Technological University, Kakinada

ABSTRACT

Road extraction refers to identifying and delineating roads from images by analyzing their distinctive features. The analysis of high-resolution satellite images has become an important research topic in urban studies, particularly for automatic road network extraction. Road extraction can be performed using three main methods: manual, automatic, and semi-automatic approaches. In this project, we review various road extraction methods using satellite imagery. Our approach focuses on extracting roads from satellite images by applying mathematical morphology techniques in combination with Canny edge detection.

Keywords - Mathematical Morphology, Canny Edge Detection

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Introduction

Geographical information extraction from remotely sensed images has been an active research subject with a variety of applications in recent years, such as, urban planning, resource management, natural disaster analysis, transportation system modelling . [1]Roads in dynamic cities tend to change very frequently even within a short period of time. Road maps of these areas have to be updated periodically, preferably from current satellite images to meet the urgent need of urban planners. [2] Road information can be extracted from images in three ways: manual extraction, semi-automated extraction and fully automated detection.[3] Manual extraction is subject to the analyst's experience and skills. Roads can be recognized reasonably well even from noisy images that contain incomplete information about roads if s/he is familiar with the study area. However, this manual method is expensive and time consuming. Automatic extraction of roads from satellite images faces several challenges because the image appearance of roads depends upon the spatial resolution of the satellite images. In addition, the extraction is hampered by noise on satellite images. Ground objects such as trees along a street can obstruct the image of roads. Vehicles on the road may cover certain parts of a road and make it difficult to detect on the image.[2] Semi-automatic extraction of roads from satellite images is efficient way of extracting the road.It provide best robustness, quality and completeness. Accuracy is also improved by semi-automatic method. In this method seed point provide by user [4].

1.2 Problem Statement

Most of the states of India and therefore the whole of India need to be digitally mapped. This was manually done about 80 years ago by the British. Most of the existing services of GPS are provided based on the data extracted and inserted manually onto the databases of online map resources like Google Earth and Wikimapia. The extraction and updating of road network databases is crucial to many Geographic Information System (GIS) applications like navigation, urban planning, etc. The development of mature methodologies for automatic road extraction and change detection based on imagery may provide a hopeful solution to this problem. An operational road database extracting and updating system should include the following three main functions:

1) The generation of a road network from the satellite imagery.
2) The detection of roads using digital image processing based methods.
3) The updating of the road database.

1.3 Motivation

With developing space imagery technologies, automatic object detection from satellite images became an essential component for many applications. Likewise, automatic detection of road networks from the satellite or aerial images is a popular research subject.

The automatic road extraction is primarily needed by Geographic Information Systems (GIS) applications, since manual data acquisition for GIS database update is a costly and time consuming task. GIS applications have a wide area of usage including defense industry, transportation systems, emergency management, urban planning and navigation applications.

Despite the existence of high demand on the automatically acquired road information for GIS applications, a robust road extraction algorithm outperforming all other alternative extraction techniques does not exist. Therefore, the topic is needed to be studied further, and new road extraction strategies should be devised.

1.4 Scope

This study is devoted to explore the automatic road extraction problem and to propose a new road extraction algorithm .In fact his dissertation combines the insights gained from two years of research conducted on the subject of road extraction .In the scope of the study, the strengths of Canny Algorithm are analyzed and reported. IKONOS images are utilized for development and experimentation stages. The main purpose of this study is to establish a new road detection scheme exploiting supervised learning and optimization routines in order to construct an accurate classification model.

2. LITERATURE REVIEW

2.1 Survey on Methods of Road Extraction using Satellite Image

Bibo Lu; Yongxia Ku; Hui Wang

Significant research effort is spent for the road extraction problem from aerial or satellite imagery. The common aim is to extract roads represented by binary masks in which true pixels represent road regions and false pixels indicate non-road regions. The first research attempts for this topic have begun in the mid-70s by the study of (Bajcsy & Tavakoli, 1976). For over two decades, automatic road extraction from remote sensing imagery has been an active research area in computer vision, photogrammetry, remote sensing and geographic information system (GIS) researchers. In a detailed literature survey study, approximately 250 state-of-art techniques consisting of several road extraction studies are reviewed by (Mena J. ,). In this study, methods developed for road extraction from satellite images between the years of 1987-2002 are classified into several categories according to their preset objective, extraction technique applied and type of sensor utilized. In a more recent study, (Hauptfleisch, 2010) reviews the major contributions to the topic by grouping them into categories such as automatic seeding, classification, edge detection, Hough transform, multi-resolution analysis and road tracking. Both of these articles introduce the essential methods used by many researchers for years in the perspective of their studies. In this section, the studies conducted by former researchers concerned with the road extraction problem are introduced.

Road extraction methods

Although many researchers have classified the road extraction methods, it is still difficult to classify them in detail due to various applications. In a qualitative survey, it can be found that most of the methods suggested in literature for road extraction consist of one or more types of algorithms: classification-based, knowledge-based, mathematical morphology, active contour model, and dynamic programming, etc.

1. Classification-based methods

Classification-based methods usually use the geometric features, photometric features and texture features of a road.According to the use of labeled training samples, the classification-based methods can be divided into supervised and unsupervised.

1.1 Supervised classification methods

Supervised classification methods are to train the labeled samples. To a large extent, the accuracy of supervised classification methods relies on the selected features and labeled samples. In general, the supervised classification methods are as follows: artificial neural network (ANN), support vector machine (SVM), Markov random fields (MRFs) classifier and maximum likelihood (ML) classifier.

1.2. Unsupervised classification methods

Unsupervised classification methods do not need training samples, which have many advantages in solving classification problems.The most common algorithms are various clustering algorithms, which include K-means, spectral clustering, mean shift and graph theory, etc.

2. Knowledge- based methods

It is difficult to extract road from RS images only using the local spectrum and texture information. Due to the stripe structure of the road making it hard to describe the eigenvector, the data could not be input to the classifier directly. So parameter models such as the energy function can be used to operate on the maximum value of the energy function. The common parameter models usually extract some structural elements according to the relationship among them, and to detect the specific structure so as to realize the object detection finally.

3. Mathematical morphology methods

The mathematical morphology has caused widespread attention in the academic circles, such as image processing, pattern recognition, computer vision and other fields. In the 1980s, researchers started to use mathematical morphology on road extraction, after then all kinds of road extraction methods based on the mathematical morphology have been proposed (Valero et al., 2010 and Zhu et al., 2005). However, mathematical morphology methods usually combine with other methods for image segmentation.

4. Active contour model

Active contour models include parameter active contour model and geometric active contour model, and they are respectively represented by snake and level set. The principle of the models is to use a continuous curve for expressing the object profile, and to define an energy function in order to make the process of image segmentation turn into the minimum value of the energy function. The value can be achieved by solving the Euler's equation. Once the energy reaches to the minimum, the object profile can be achieved.

5.SVM Classification method

The SVM is a supervised learning method, which was firstly proposed for classification and regression analysis. The basic meaning can be described as follows: through the nonlinear transform of the kernel function to transform the low dimensional space to the high dimension space, it constructs the minimal generalization error linear discriminant functions in the high dimensional space so as to realize linear or nonlinear classification in the low dimensional space. Yager and Sowmya (2003) exploited the SVM classifier by using edge-based features such as gradient, intensity, edge length, but the correctness is relatively low as many researchers reported. Melgani and Bruzzone (2004) used the SVM methods to conduct the classification of a high resolution RS image. In many cases, the classification methods of SVM are better than those of the radial basis function neural network and K-nearest neighbor classifier in terms of the accuracy, stability and robustness. The road and building detection using multiclass SVM method was proposed by Simler (2011). The suggested approach includes image segmentation and classification algorithms, which are especially well adapted to multispectral data. And the both spatial and spectral information are used at the object level. The SVM methods have advantages of the structural risk minimization and the good generalization ability, which are widely used in object detection from a RS image. However, there are difficulties to use the SVM methods, such as the estimation of kernel functions, the choice of the dimensional space and training samples, etc.

6.ANN Classification method

The ANN is inspired from a biological neural system. It is also a computational model that is composed of nodes (or neurons), which are connected to each other. In the late 80's, Heermann and Khazenie (1992) proposed the back propagation (BP) algorithm, which makes a rapid development of the road extraction methods based on neural network. The early work was mostly based on the spectral and contextual information of the image pixels using BP neural network and the improved model to classify directly.

Tu-Ko (2003) presented a robust approach of road centerline delineation, in which a neural network was trained with the spectral and edge information. Although the extraction results include many non-road edge segments, the system can achieve good results on the whole. A BP neural network method was used by Mokhtarzade and Valadanzoej (2007).

Through entering different parameters, they were able to get the optimal input vector and test a variety of network structures with iteration time. The optimal network structure and the termination condition in training can be finally established under these conditions, but the process of the input parameters is relatively tedious. The BP neural network was applied to road detection by Kirthika and Mookambiga (2011). At first, the spectral information for road detection was used. Then different texture parameters including the contrast, energy, entropy and homogeneity for each pixel were computed by using the gray level co-occurrence matrix (GLCM) from the source image, and a pre-classified road raster map was produced. To optimize the system functionality and to evaluate the impact of contributing texture parameters on road detection, the extracted texture parameters were integrated with the spectral information. However, the disadvantages of the BP neural network methods include the following aspects: the convergence speed is slow; it needs more training samples; it is more likely to get into the local minima; it declines faster in performance with the increasing categories; and it is easy to become over- fitting, etc. Hence, many novel or improved neural network models have been used for road extraction from RS images. For example, the radial basis function neural network, fuzzy neural network, spiking neural network and hybrid neural network (George et al., 2013; Li and Chen, 2014)

Table 1 – Comparison of different road extraction methods.

Algorithm	Road feature	Sample	Disadvantage	Performance accuracy or correctness (%)
ANN	Intensity, edge, spectrum	1–2	Discontinuity, noisy, over-fitting	95
SVM	Intensity, edge gradient, length, width	3–6	Require more samples, low precious	13–35
MRFs	Mean intensity value, texture	5	Manual intervention	63.5–93.4
Mean shift	Histogram of the HIS image	1–8	Over-segmentation, long literation time	86–90
Knowledge-based methods	Intensity, edge	2–4	Over-segmentation, susceptible to occlusion and shadow	70–79
Mathematical morphology	Geometric feature, direct of line	3–4	Discontinuity, susceptible to structural elements	91.76
Active counter model	Intensity gradient	3	Depend on the seed point selection	95–99
Filtering and group	Direct of line, intensity	2–6	Rely on the prior knowledge	80–98

TABLE 1: COMPARISION OF DIFFERENT EXTRACTION METHODS

2.2 ROADWAY EXTRACTION IN URBAN AREAS USING SATELLITE IMAGERY

Satellite imagery is an important source for the acquisition and update of GIS data. Here the semi-automatic extraction of roads from satellite imagery is proposed. To extract the road from urban areas an effective edge detection algorithm based watershed transform is proposed to yield satisfactory and efficient segmentation of the aerial images for edge detection. The high-resolution images are preferable due to use of width and variance information for road extraction. This algorithm works successfully for straight roads and roads having junctions, minimum obstacles and detects central line of the roads in any orientation by setting proper threshold for satellite images.

There are several approaches for road extraction. In this paper a new algorithm edge detection with the combination of watershed segmentation using distance transform is proposed to extract the roads from aerial image with high resolutions by using semi-automated methods. In order to detect the roads edge detection filters which are the types of high pass filters (Sobel) are conducted to satellite images having different high resolution. In this way spatial frequency of each image was increased so the roads were highlighted and then extracted from each image.

Some pictures may be very noisy and image processing becomes more and more complex due to this the markers selection and extraction are not so easy. In other cases, the objects to be detected may be so complex and so varied in shape, grey level and size that it is very hard to find reliable algorithms enabling their extraction. For that reason segmentation of image is needed.

The watershed transform presents some advantages: The watershed lines always correspond to the most significant edges between the markers. So this technique is not affected by lower-contrast edges, due to noise, that could produce local minima and, thus, erroneous results, in energy minimization methods. Even if there are no strong edges between the markers, the watershed transform always detects a contour in the area.

Roads have different characteristic in urban and rural regions. Proposed methods have high performance for specific region and area. In order to have high road extraction performance for both rural and urban area, it is required to develop a modular approach for road extraction algorithm. Mainly the first step is to apply pre-processing techniques that includes reducing the noise and adjust the image intensity by preserving image information in it.

The noisy images lead to over segmentation and not an accurate segmented image. The first step is the rgb to grey conversion and then the dimensional reduction is done using wavelet transform.

Then the next step is Histogram equalization which is done for contrast adjustment. Then noise is removed from the image by filtering process and pixel values are adjusted so that they will help to obtain the well segmented image. The next step is pre segmentation processes that includes various morphological operations such as finding out regional maxima and mark the foreground objects that help in segmentation process. In the third step the main task is segmentation. After reconstructing the image we superimpose it with the original image, clean the edges of the segmented image and compute background markers. The last step is to apply watershed transformation to the distance transform of the image and then see the result that is segmented image.

The proposed algorithm detects the single, bifurcating and intersection road by setting proper threshold. The median filter is used to remove the noises like shadows and vehicles in the road. The edges of the road is extracted using sobel operator. Current limitations is the algorithm fails on the road cast by different resolution images. The proposed algorithm detects only the roads in urban area and since this is a semi-automatic approach, manual thresholding is to be done for the extraction of road.

2.3 Fast Road Network Extraction in Satellite Images using Mathematical Morphology and Markov Random Fields Thierry G eraud

In the particular field of satellite imagery, many different methods have been proposed to extract roads. Most of these methods fortunately rely on a global optimization process but suffer from drawbacks. Many methods are close to tracking like approaches and cannot take into account features extracted from image regions. Many methods consider that a road is a set of straight lines so road accuracy is quite poor. Last, most of these methods take several minutes to detect roads in rather small satellite images. The author presented a fast method to extract road network in satellite images. A pre-processing stage relies on mathematical morphology to obtain a connected line which encloses road network. Then, a graph is constructed from this line and a Markovian Random Field is defined to perform road extraction.

The watershed transform (WT) is a morphological algorithm usually used for the purpose of segmentation. Considering a grey level image as a topographic map, let us denote by catchment basin associated with a regional minimum of this map, all points whose steepest slope paths reach this minimum.

The watershed line is a closed one-pixel thick crest line which separates every adjacent catchment basins, i.e., regions. Since numerous minima populate images, applying the watershed transform to an image leads to an over-segmentation.

The method proposed is composed of four steps. In the first step pre-processing: From a satellite image compute a grey level image where pixel values denote their potential of belonging to a road. Roads are thus located on crest lines of this “potential” image and mark it as V. The second step filtering consists of computing an area closing of the potential image and then running the watershed transform. The potential image, once closed, has much less minima than the “original” potential image while properly retaining crest lines location. Therefore, the resulting watershed line includes the road network. Multi-scale properties of this morphological filtering can be observed: new curves (features) do not appear when area (scale) increases and a feature which is present at a given scale (a piece of watershed line obtained with a given area) is still present at a lower scale (in the watershed line obtained with a smaller area). This property is very important for us since the only parameter of this filtering step is the area; even with a large value of area, we are guaranteed to have important roads be included in watershed line.

In the third step from the watershed line, a curve adjacency graph (CAG) is built. A node of this graph represents a shed, that is, a connected part of the watershed line separating two adjacent basins. An edge is drawn between two nodes/sheds if one end of the first shed is connected with an end of the second one through the watershed line. For every node they make the distinction between edges coming from connections to one shed end (yellow anchors) and those coming from connections to the other shed end. This distinction, allows to properly handle in the next step some geometrical constraints upon the road network. In the last step Markovian Relaxation, extracting road network now turns out to be a graph labelling problem. Upon the graph structure, they defined Markov random fields.

The authors thus presented a method to extract road network from satellite images. They have transposed the recognition scheme “WT + RAG + MRF”, described is for image segmentation, to the problem of road network recognition. To that aim, they propose a recognition scheme that is, as far as we know, original: “area opening + WT + CAG + MRF”. This recognition scheme is a global optimization process so it provides robust and reproducible results. Moreover, it is general and can easily be adapted to other image processing fields where the recognition of curvilinear structures is involved.

2.3 REVIEW ON CANNY EDGE DETECTION

Ms. Anjum Sheikh ,Prof. R. N Mandavgane,Prof. D. M. Khatri

Qian Xu, Srenivas Varadarajan, Chaitali Chakrabarti, Fellow,IEEE, and Lina J. Karam, Fellow, IEEE "A Distributed Canny Edge Detector: Algorithm and FPGA Implementation" The Canny edge detector is one of the most widely used edge detection algorithms due to its superior performance. It is more intensive as compared to other edge detecting algorithms, but it also has a higher latency because it is based on frame-level statistics. In this paper, we propose a mechanism to implement the Canny algorithm at the block level without any loss in edge detection performance compared with the original frame-level Canny algorithm. Directly applying the original Canny IEEE, and Pierre Manneback "A Multi-Resolution FPGA- Based Architecture for Real-Time Edge and Corner Detection."This work [4] presents a new flexible parameterizable architecture for image and video processing with reduced latency and memory requirements, supporting a variable input resolution.

The proposed architecture is optimized for feature detection, more specifically, the algorithm at the block-level leads to excessive edges in smooth regions and to loss of significant edges in high-detailed regions since the original Canny computes the high and low thresholds based on the frame-level statistics. To solve this problem, we present a distributed Canny edge detection algorithm that adaptively computes the edge detection thresholds based on the block type and the local distribution of the gradients in the image block. It is capable of supporting fast edge detection of images and videos with high resolutions, including full-HD since the latency is now a function of the block size instead of the frame size [2].Paulo Ricardo Possa, Student Member, IEEE, Sidi Ahmed Mahmoudi, Naim Harb,Carlos Valderrama, Senior Member, canny edge detector and the Harris corner detector. The architecture contains neighborhood extractors and threshold operators that can be parameterized at runtime. A performance analysis of the FPGA and the GPU implementations, and an extra CPU reference implementation, shows the competitive throughput of the proposed architecture even at a much lower clock frequency than those of the GPU and the CPU. Maintain a reliable performance with noisy images, low latency and memory requirements [5].

2.3 A MATHEMATICAL MORPHOLOGICAL PERSPECTIVE N THE WORLD OF IMAGES

Rahul Gaurav

This methodology was initiated by G.Matheron and J.Serra for the quantitative analysis of spatial structures, at the Paris School of Mines. Mathematical morphology is a tool for extracting image components that are useful for representation and description. It's mathematical origins stem from set theory, topology, lattice algebra, random functions, stochastic geometry, etc. Hence, we say that it involves set theoretic method of image analysis providing a quantitative description of geometric structures. It is most commonly applied to digital images, but it can also be employed on surface, surface meshes, graphs, solids and many other spatial structures. It characterizes various topological and geometric continuous-space concepts such as shape, size, convexity, connectivity and geodesic distance on continuous and discrete spaces. It is based on shapes in the images not the pixel intensities that are viewed as a general image-processing framework. Generally we use it before and after image segmentation (except the case of watershed segmentation). Two fundamental morphological operations – erosion (shrinking) and dilation (expansion) are based on Minkowski operations. There are two different types of notations for these operations: Serra/Matheron notation and Haralick/Sternberg notation.In this , Haralick/Sternberg notation is used which is the more often used one in case of practical applications.

Need for Mathematical Morphological perspective Advances in image mining and spatial analysis have led to tremendous growth in very large and detailed spatial information retrieval. Such spatial information, if analyzed with proper tool, can reveal much useful information to the human users. Henceforth we need the most efficient technique for processing of such information. Mathematical Morphology can act as a powerful tool for solving several image related queries. The need for morphology has been evident in various fields of science and engineering as well which has been substantiated in recent years like the one in which it acts as one of the best methods to eliminate weak network lines so that there is an emergence of strongly connected subnetworks in such way that one can predict the behaviour of the network. With the help of mathematical morphology, power network images are being decimated for analytical view. It gives a quick view of the strong subnetworks in the power system [5]. Close range photogrammetry is a technology where each measured object is imaged on many photos taken from different camera positions. Mathematical morphology has the potential to make close range photogrammetry technology faster. With the proper configuration of morphological tools, we get exact results of many photos [3]. In the recent years, morphology has shown its need to the researchers in the realization of several topics like:

image enhancement, image segmentation, image restoration edge detection, texture analysis, feature generation, skeletonization, shape analysis, image compression, component analysis, curve filling, general thinning, feature detection and noise reduction.

2.3 Applications

It has shown directions to several research and development works across the globe by providing an easier method in several image related applications. It helps in the teeth detection of a gear using subtraction and labeling, in getting the grid identification from Biochip by detecting the size of parts and analyzing its shape (pattern spectrum) using OTSU and entropy threshold. Another important application lies in the detection of runways in satellite airport imagery, which is a multi step algorithmic process that involves White Top-hat Transformation (segmentation tool that extracts respectively dark Seminar on Spatial Information Retrieval, Analysis, Reasoning and Modelling objects from the uneven background) of the source, image threshold and reconstruction of the detected long features to get the ultimate result. Its easy-to-use mathematical techniques have helped in the medical field too. In the detection of filarial worms, this tool has been proven to be the most efficient one. In such cases, Black Top-hat Transformation of source is firstly done. Hence the reconstruction after eliminating the short structures of the skeleton gives the final result. In advanced medical image processing, morphology helps to segment the vertebra and ribs. The other striking applications of morphology are related to the edge detection, which is based on binary dilation, binary erosion and image subtraction [10]. Reduced noise Morphological Gradient Edge Detector, Erosion Residue Edge Detector, Dilation Residue Edge Detector, and Morphological Gradient Edge Detector are the several notable morphological edge detection techniques. By implementing edge detection using morphology, the edges and parts of the architectural monuments and industrial objects can be possibly detected. Mobile mapping systems is a technology where morphology helps to extract edges and detects characteristic objects in mobile photogrammetry systems to make maps that have been taken from sources like an image taken from a car or moving physical object.

3. PROPOSED SYSTEM

Object detection is a computerized technology related to image processing used in images and videos, to identify instances of real world object. Object detection is usually achieved by edge detection. Edge detection is a process to identify the edges of image by finding the place where intensity changes rapidly. In this experiment an attempt is made to detect the objects by using canny edge detection method in java. The detected objects can be further used for watermarking purpose. The comparison between canny and sobel edge detection is carried out. Morphological filters are used to improve canny edge detection. Contour based learning technique is used to draw contours for detected objects.

Edge detection is a technique in image processing to identify a curve or a path where intensity changes rapidly. There are many type of edge detection technique like Canny, Sobel, Roberts, Prewitt. Canny edge detection is a non maximum suppression technique based on Gaussian filterMorphological filters are collection of non linear operation carried out relatively on the ordering of pixels without affecting their numerical value. Erosion and Dilation are two fundamental operators in morphological filters. Erosion replaces the current pixel value with minimum value found in defined set of pixels. Dilation replaces the current pixel value with maximum value found in defined set of pixels. A watermarking is a process of hiding cryptographic information in a host data to protect its ownership and authenticity. To perform watermarking in an image or video, identifying Region of Interest and Object plays a major role. Object Detection is a technique related with computer vision and image processing. The original canny algorithm relies on frame-level statistics to predict the high and low thresholds and thus has latency proportional to the frame size. In order to reduce the large latency and meet real-time requirements, we presented a novel distributed Canny edge detection algorithm which has the ability to compute edges of multiple blocks at the same time. Thus our proposed project will detect the edges efficiently with reduction in the processing speed and reduced the memory. Time required to detect the edges is less.

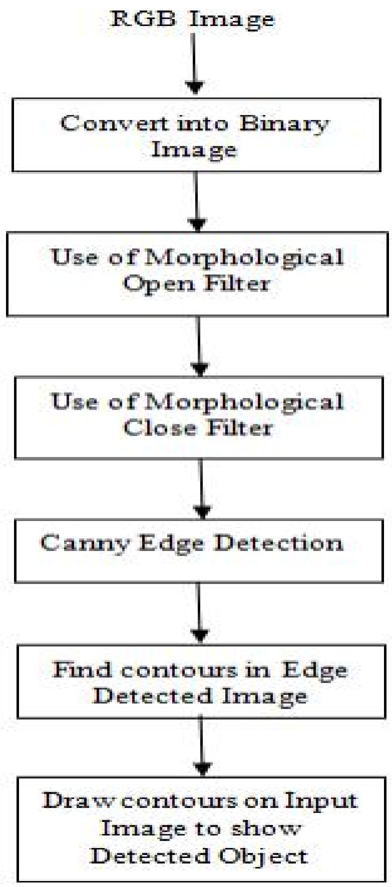


Fig 1: Diagram of Proposed System

The method proposed in this contains following procedure:

1. A RGB image is taken as input and it is converted into gray image.
2. Morphological opening filter is applied to binary image using OpenCV.
3. Close Filter is applied to open filtered image using OpenCv function.
4. On open and close filtered image, Canny Edge detection algorithm is applied.
5. Contours are found from edge detected image and numbers of contours are counted.
6. To indicate the identified object using Canny Edge Detection with less number of edges, Contours are drawn on original image.

Gray Scale Images

. Image formation using sensor and other image acquisition equipment denote the brightness or intensity I of the light of an image as two dimensional continuous function F(x, y) where (x, y) denotes the spatial coordinates when only the brightness of light is considered. Sometimes three-dimensional spatial coordinate are used. Image involving only intensity are called gray scale images. Resolution Similar to one-dimensional time signal, sampling for images is done in the spatial domain, and quantization is done for the brightness values . In the Sampling process , the domain of images is divided into N rows and M columns. The region of interaction of a row and a Coolum is known as pixel.

The value assigned to each pixel is the average brightness of the regions. The position of each pixel was described by a pair of coordinates (xi, xj). The resolution of a digital signal is the number of pixel is the number of pixel presented in the number of columns × number of rows. For example, an image with a resolution of 640×480 means that it display 640 pixels on each of the 480 rows. Some other common resolution used is 800×600 and 1024×728, among other. Resolution is one of most commonly used ways to describe the image quantity of digital camera or other optical equipment. The resolution of a display system or printing equipment is often expressed in number of dots per inch. For example, the resolution of a display system is 72 dots per inch (dpi) or dots per cm.

Gray levels

Gray levels represent the interval number of quantization in gray scale image processing. At present, the most commonly used storage method is 8-bit storage. There are 256 gray levels in an 8 bit gray scale image, and the intensity of each pixel can have from 0 to 255, with 0 being black and 255 being white we. Another commonly used storage method is 1-bit storage. There are two gray levels, with 0 being black and 1 being white a binary image, which, is frequently used in medical images, is being referred to as binary image . As binary images are easy to operate, other storage format images are often converted into binary images when they are used for enhancement or edge detection.

Mathematical Morphology

The mathematical morphology has been widely used in digital image processing and focuses on the area that studies the geometric properties of objects in the images. This allows the extraction of image components that are useful in the representation and description of the shape of a region, such as borders and skeletons (Gonzales; Woods, 2000). The extraction of elements present in an image is accomplished with the help of a suitable structural element. The structural elements are matrices responsible for the removal or addition of labeled pixels in the image, which depend on their size and shape, usually defined by the User, according to the area to be applied. In this paper the mathematical morphology adopted is binary, thus only binary morphological operators were used. The basic operations of morphology are erosion and dilation, at the first the pixels that do not conform to a given pattern are deleted from the image and at the second a small area related to a pixel is changed to a given pattern. These operations are the basis for most of the operations used in mathematical morphology, in other words, they are combinations of these such as opening, closing, skeletonization, among others. The dilation is a morphological operation that combines two sets using the vector addition of elements of sets. Its symbol is $\oplus$, the result as the name suggests is a dilated image, like this the effect of the dilation on an image is the growth or expansion of the object. These objects refer to the pixels whose gray level is greater than zero in relation with the background. The dilation can also be understood as the union of translations of A by elements of B. The Dilation of a set A by B denoted by $A \oplus B$, is given as:

$$A \oplus B = (x \mid (B)\, x \cap A \neq Ø)$$

(1) where A represents the image being operated on and B is called structuring element and its composition defines the dilation, so the dilation expands an image. Therefore, the dilation of A by B is the set of all x displacements such that and A overlap in at least one nonzero element.

Unlike the dilation, the erosion reduces the object present in the image against the background. It is a morphological operation that combines two sets using vector subtraction of elements of sets, its symbol is $\ominus$. The erosion of A by the structuring element B, denoted by $A \ominus B$, is given as

$$A \ominus B = (x \mid (B)\, x \subseteq A)$$

(2) The erosion of A by B is the set of all points x such that B, when translated by x, be contained in A. The result of successive erosions and dilations allow the elimination of specific details of the image, smaller than the structuring element without distortion of the features not deleted. The effect of re-application is no longer to modify the previously transformed result. The opening operator is used to remove parts of objects or even objects smaller than the structuring element. Thus, the opening operator can eliminate noise due to the erosion operator that is applied initially. The opening of a set A by B, denoted by A ° B is given by equation 3.

$$A \circ B = (A\ B) \oplus B$$

where A represents the image being operated on and B is called structuring element. The closing operator tends to join "islands" which the distance between them is less than the structuring element and closing holes smaller than this element. Being the same set A and a structuring element B, the closing of A by B, denoted by A • B, is given by equation 4.

$$A \bullet B = (A \oplus B) \square B$$

Soon, these operated jointly applied enable the formation of the most compact and at the same time, eliminate regions very small or thin.

Mathematical morphology is an image extraction tool based on set theory. It is collection of non linear process used to remove unwanted small details. A configuration of pixels of different shape and size defined with 0 and 1 called structural element plays an important role in mathematical operation. Morphological Filters have two basic functions: Erosion and Dilation.

Open Filter: Opening is a morphological filter where erosion is followed by dilation. It removes noise, round corner from inside and simplifies the image.

Open (src,element)=dilate(erode(src,element))

Closing Filter: Closing is a morphological filter where dilation is followed by erosion. It smoothens the contour and maintains shape and size of object.

Close (src,element)=dilate(erode(src,element))

From the above experiment is proved that canny edge detection is better edge detection technique. But as discussed in [2] it detects very fine edges which are not helpful in identifying individual object which can be further used for watermarking technique. In aim to reduce the number of edges in canny edge detection and to get a prominent object following methodology is carried out in Java with OpenCV.

Canny Edge Detection

Canny developed an approach to derive an optimal edge detector based on three criteria related to the detection performance. The model was based on a step edge corrupted by additive white Gaussian noise. A block diagram of the canny edge detection algorithm is shown in Fig. The original canny algorithm [6] consists of the following steps:

1. Smoothing the input image by Gaussian mask.
2. Calculating the horizontal gradient Gxand vertical gradient Gyat each pixel location by convolving with gradient masks.
3. Computing the gradient magnitude G and direction θGat each pixel location.
4. Applying Non-Maximal Suppression (NMS) to thin edges.
5. Computing high and low thresholds based on the histogram of the gradient magnitude for the entire image.
6. Performing hysteresis Thresholding.
7. Applying morphological thinning on the resulting edge map.

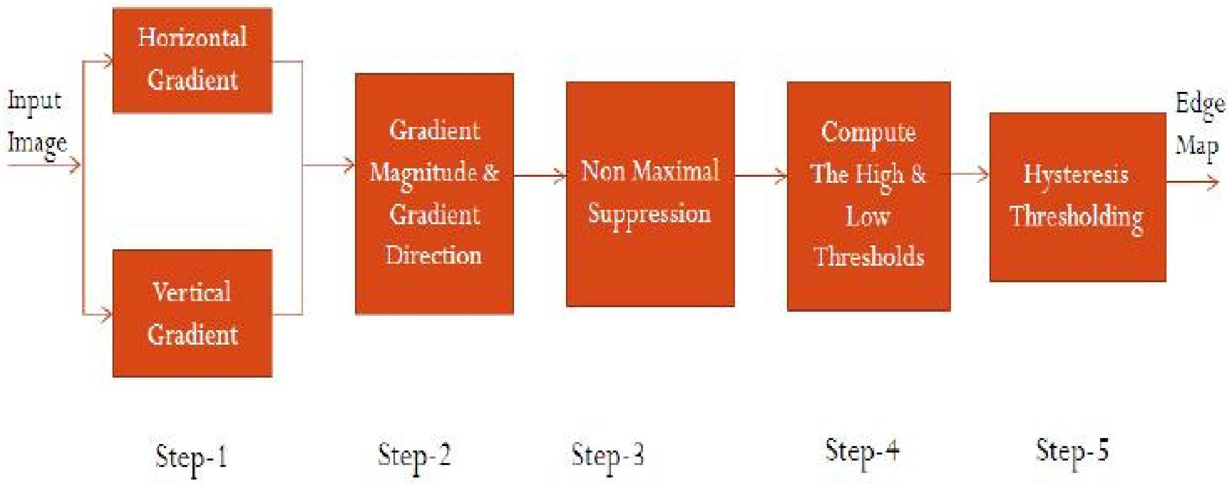


Fig. 2:*Block Diagram of the Canny Edge Detection Algorithm*

First step:

The first step in Canny detector algorithm is to filter out any noise in the original image before trying to locate and detect any edges. Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of a pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increase slightly as the Gaussian width is increased.

Second step:

After smoothing the image and eliminate the noise, next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image for approximately the edge strength at each point that can be found. The Sobel operator uses a pair of 3x3 convolution mask, one estimating the gradient in the x-direction(columns) and the other estimating the gradient in the y-direction.

Third step:

Find the edge direction, which is to ensure the rotation in the x -direction and y-direction. If it is equal to zero, there will be some error.

Fourth step:

Once the direction obtain, the next step is to relate the edge direction to a direction that can be traced in an image. If a pixel in a 3x3 or 5x5 will be shown.

Then, the possibility of the direction can be seen, there only have four possibility direction to the surrounding pixels. 0 degrees will show in the horizontal direction, 45 degrees along the positive diagonal, 90 degrees in the vertical direction or 135 degrees along the negative diagonal. Now the edge orientation must be resolve into one of these four directions depending on which direction it is closest to.

Fifth step:

After the edge direction are known, nonmaximum suppression now has to be applied.Nonmaximum suppression is used to trace along the edge in the edge direction and suppress any pixel value.

Sixth step:

Hysterisis is used as means of eliminating streaking. Streaking is the breaking up an edge contour caused by the operator output fluctuating above and below the threshold.

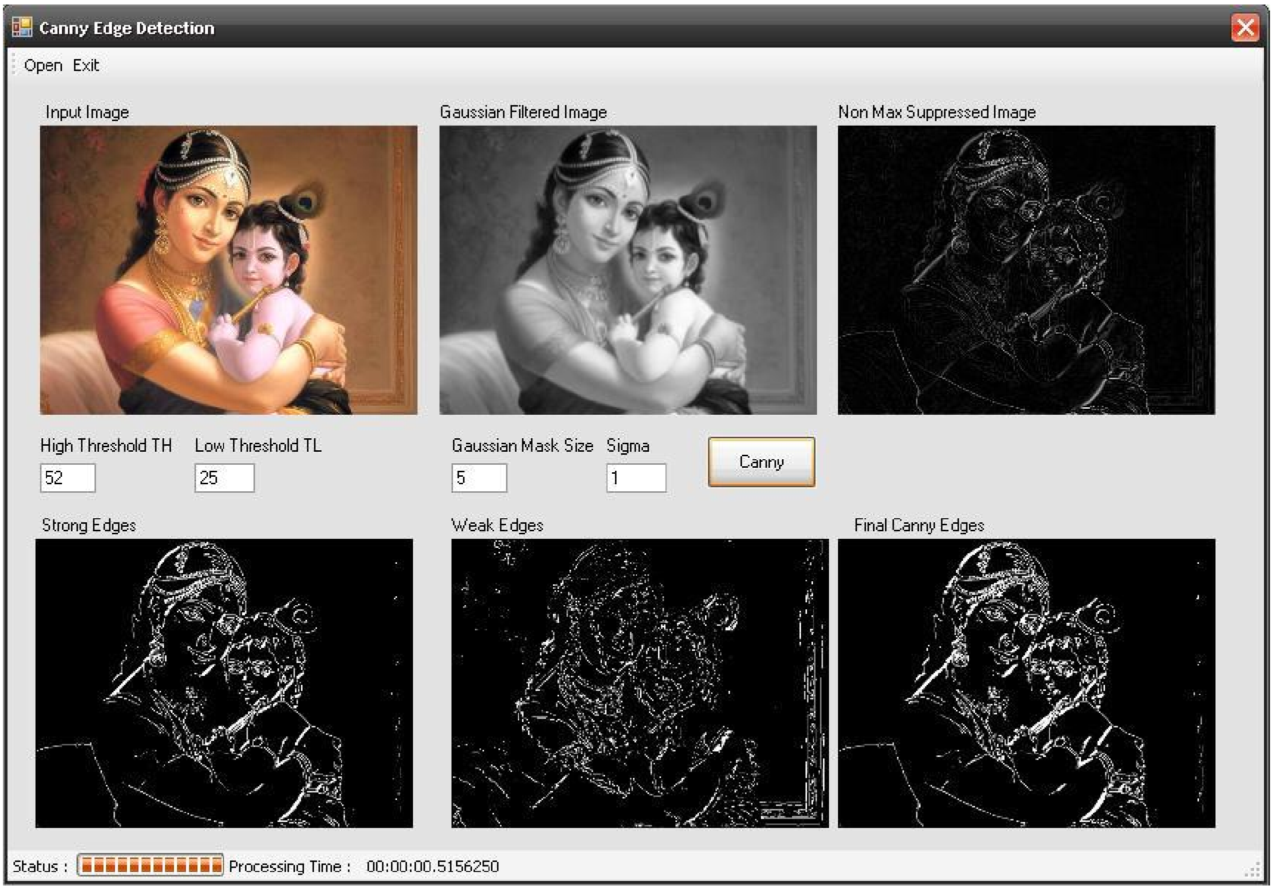


Fig 3: Canny Edge Detection

Comparative Analysis on Edge Detection Techniques

Edge detection is a fundamental tool aimed at identifying transition in host image where brightness changes sharply. It consist set of mathematical methods to extract set of curved line segment called edges. There are many edge detection operators like Canny, Robert, Prewitt, and Sobel. Sobel operator is a differentiation operator which is used in edge detection technique.

It computes approximation of gradient of image intensity function. The resultant image obtained by applying sobel filter will be either norm in vertex or corresponding gradient of input image. Canny edge detection is an operator that detects wide range of objects in a given image using multi stage algorithm. This multi stage algorithm consist mainly four steps: smoothening, finding gradient, non maxima suppression, hysteresis Thresholding.

In [7] after conducting canny and sobel edge detection on a sample image they concluded that canny edge detection holds good for object recognization compared to sobel operator and in order to pick a best edge detection algorithm for object detection following experiment is carried out. The experiment is carried out on two types of image, simple image, with less number of edges and pattern and second one complex image with large number of edges and pattern.

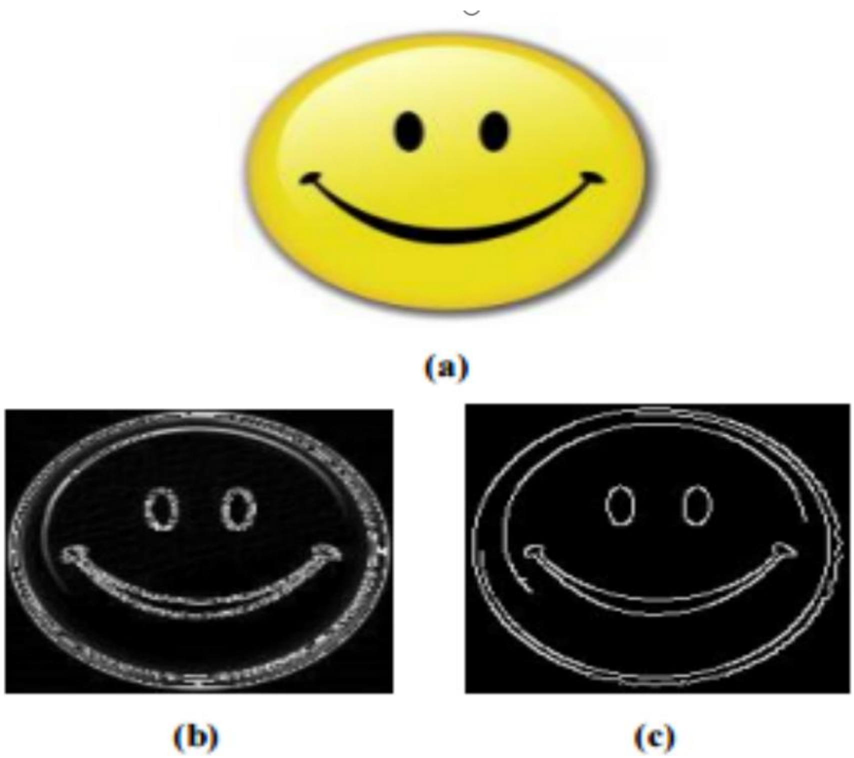


Fig 3.5 - a. Simple Image with less number of edges,

b. Sobel Edge Detection Output, c. Canny Edge Detection Output

Parameter	Sobel	Canny
Computation Complexity[1]	Simple	Complex
Detected Edges	Less no. of degraded edges due to noise increase	Fine edges are detected
Image Pattern	Blur image, no clear pattern detected	Clear image, exact and clear pattern detected
Input Image Complexity	Suitable for simple image with less no. of edges	Suitable for all kind of image

TABLE 2 : COMPARISION OF EDGE DETECTION METHODS

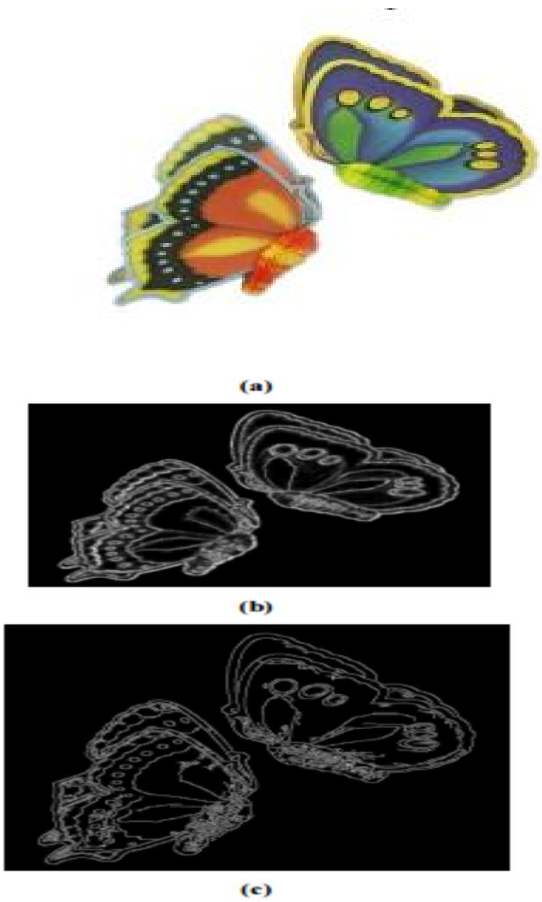


Fig 3.6 - a. Complex Image for pattern and large number of edges,

b. Sobel Edge Detection Output, c. Canny Edge Detection Output

4. SOFTWARE REQUIREMENTS

The software requirements are description of features and functionalities of the target system. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client‘s point of view. The process to gather the software requirements from client, analyse and document them is known as software requirement analysis.

4.1 FUNCTIONAL REQUIREMENTS

Functional requirements define how software behaves to meet user needs. The major functional requirements for the road from satellite images are discussed below:

Input:

The user gives satellite image as input which is taken from the database .

Output:

From the given input image it identifies road with edge detection.

4. 2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors. Non-functional requirements specify the system‘s ‘quality characteristics‘ or ‘quality attributes‘.

Software requirement:

Operating system: Windows

Language: Opencv Python

Hardware requirement:

Processor: Intel®CoreTM

RAM: 64 MB

Hard disk: 2GB or more

System Type: 32-bit Operating System or 64-bit Operating System.

5. DESIGN

We use agile model as a suitable model for the project. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

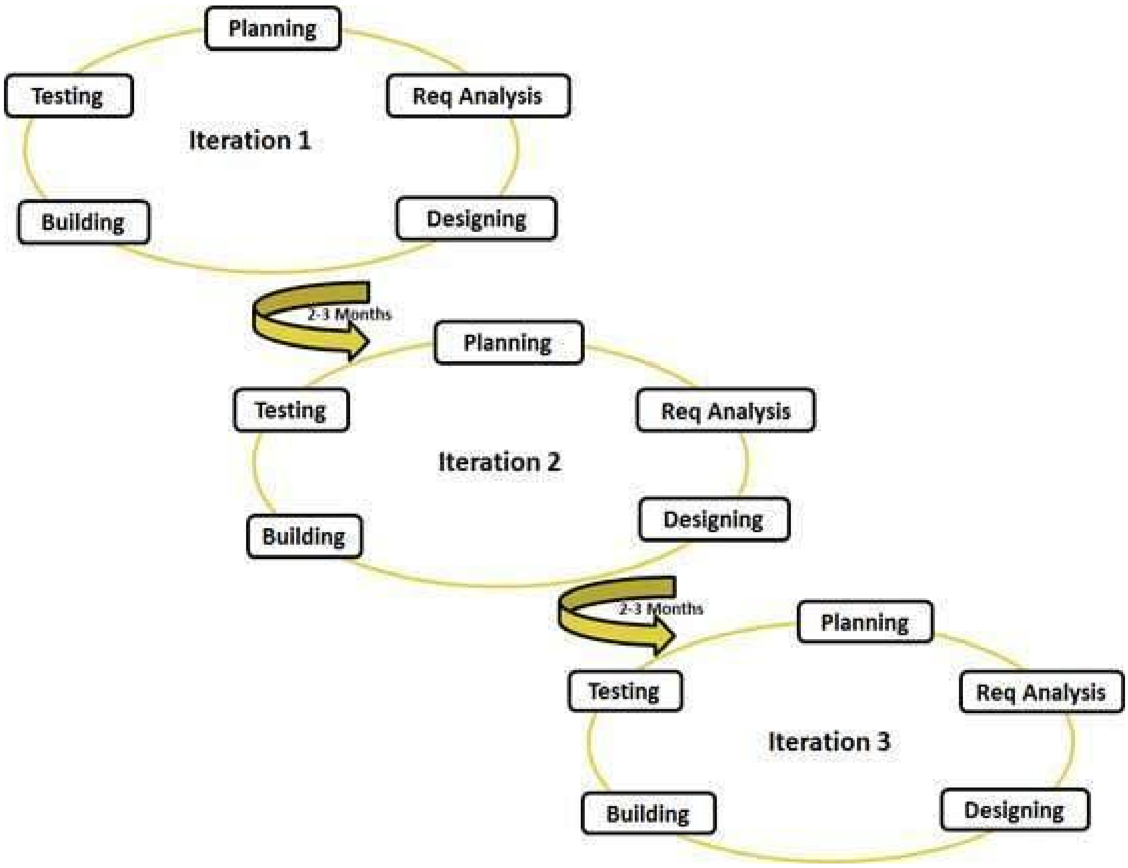
Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

At the end of the iteration a working product is displayed to the customer and important stakeholders. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.

In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer. Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

Agile uses adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer interaction is the backbone of Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.



Software design is the process or art of defining the architecture, components, modules, interfaces and for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of system analysis, system architecture and systems engineering.

5.1 USECASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user/actor and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

- Use cases involved are
    - Takes required image
    - Maintain data base
    - Morphological operations
    - Canny algorithm

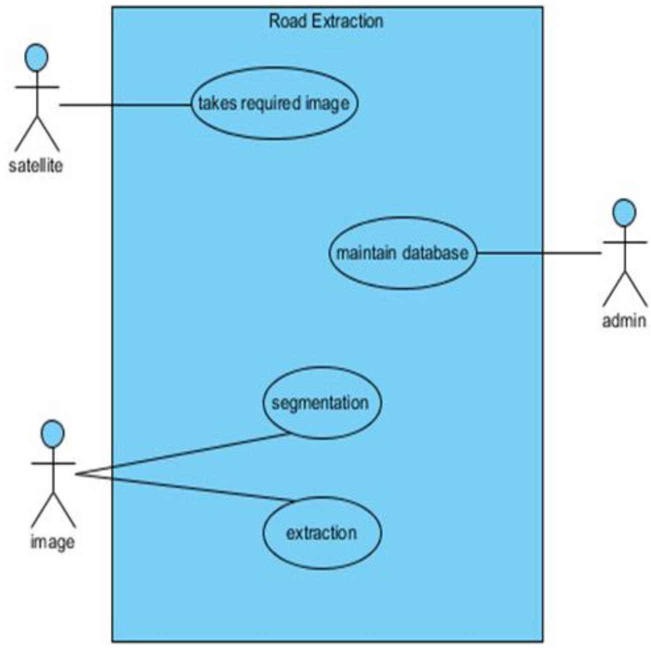


Figure 5.1: Usecase diagram

5.2 INTERACTION DIAGRAMS

The purposes of interaction diagrams are to visualize the interactive behavior of the system. Interaction diagrams are to capture the dynamic aspect of a system. So, to capture the dynamic aspect we need to understand what a dynamic aspect is and how it is visualized.

5.2.1 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

The messages performed are

- sends the original image
- image sent
- process image
- image processed
- detect edge
- edge detected by using Canny algorithm
- extract road
- road is extracted

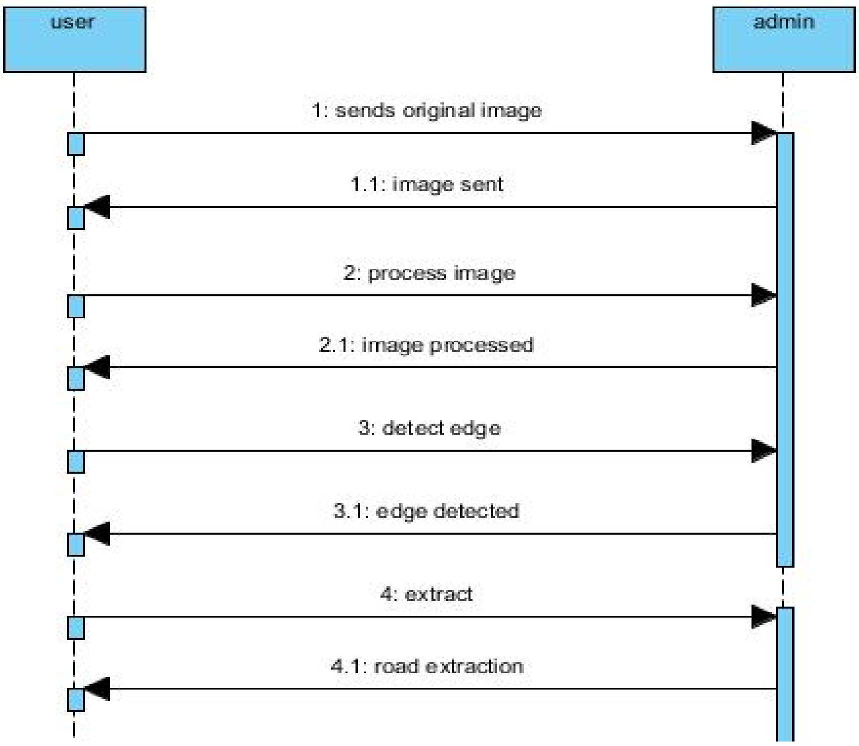


Figure 5.2:Sequence diagram

5.3 FLOW CONTROL DIAGRAM

5.3.1 Activity diagram

Activity diagram is an important diagram in UML (Unified Modeling Language) to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system .Activity diagrams are mainly used as a flow chart consists of activities performed by the system. But activity diagram are not exactly a flow chart as they have some additional capabilities. These additional capabilities include branching, parallel flow, swim lane etc.

It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

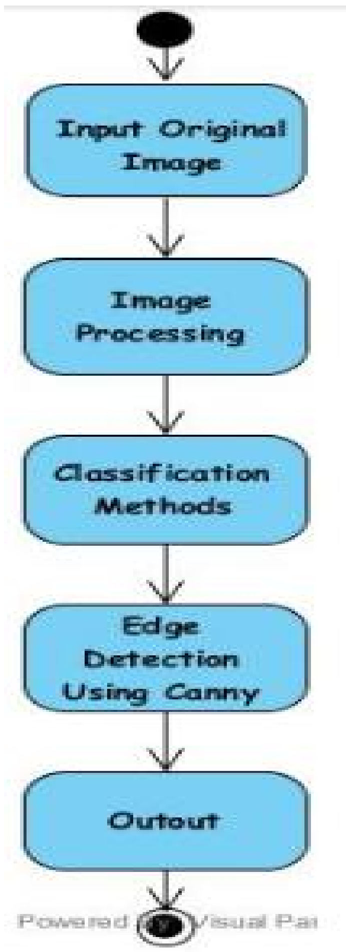


Figure 5.3: Activity diagram

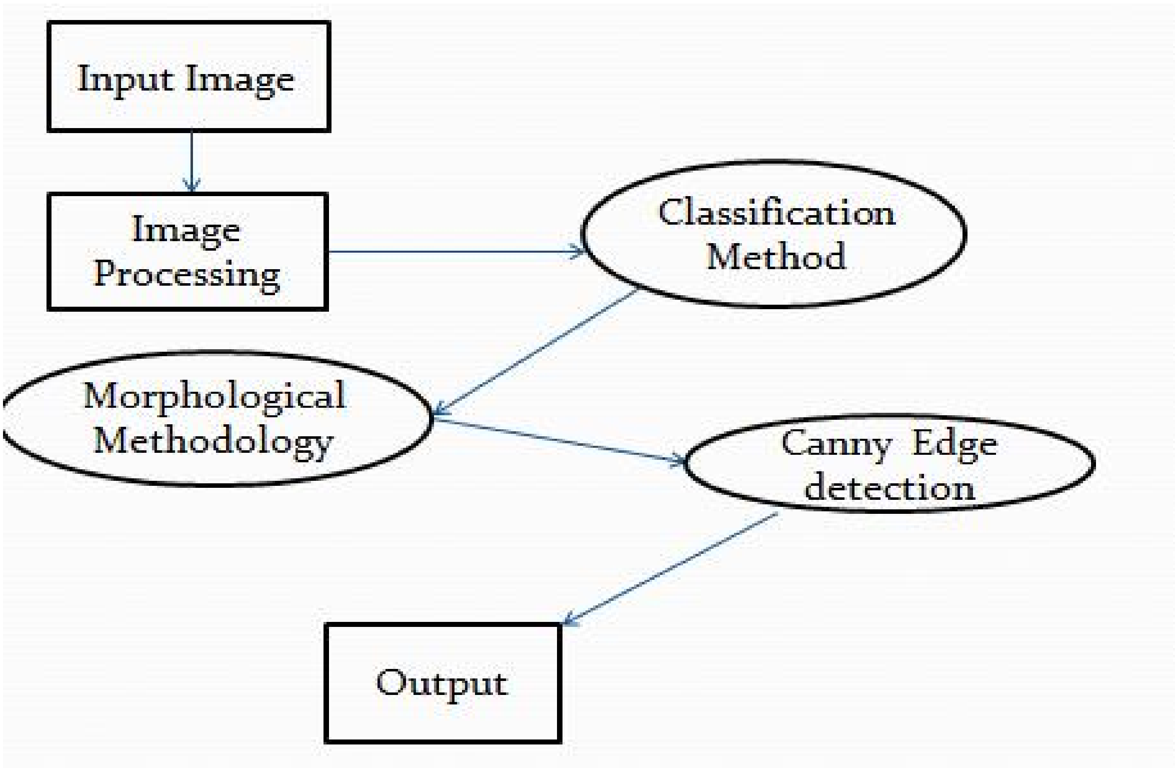
5.4 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a flowchart which also shows this information.

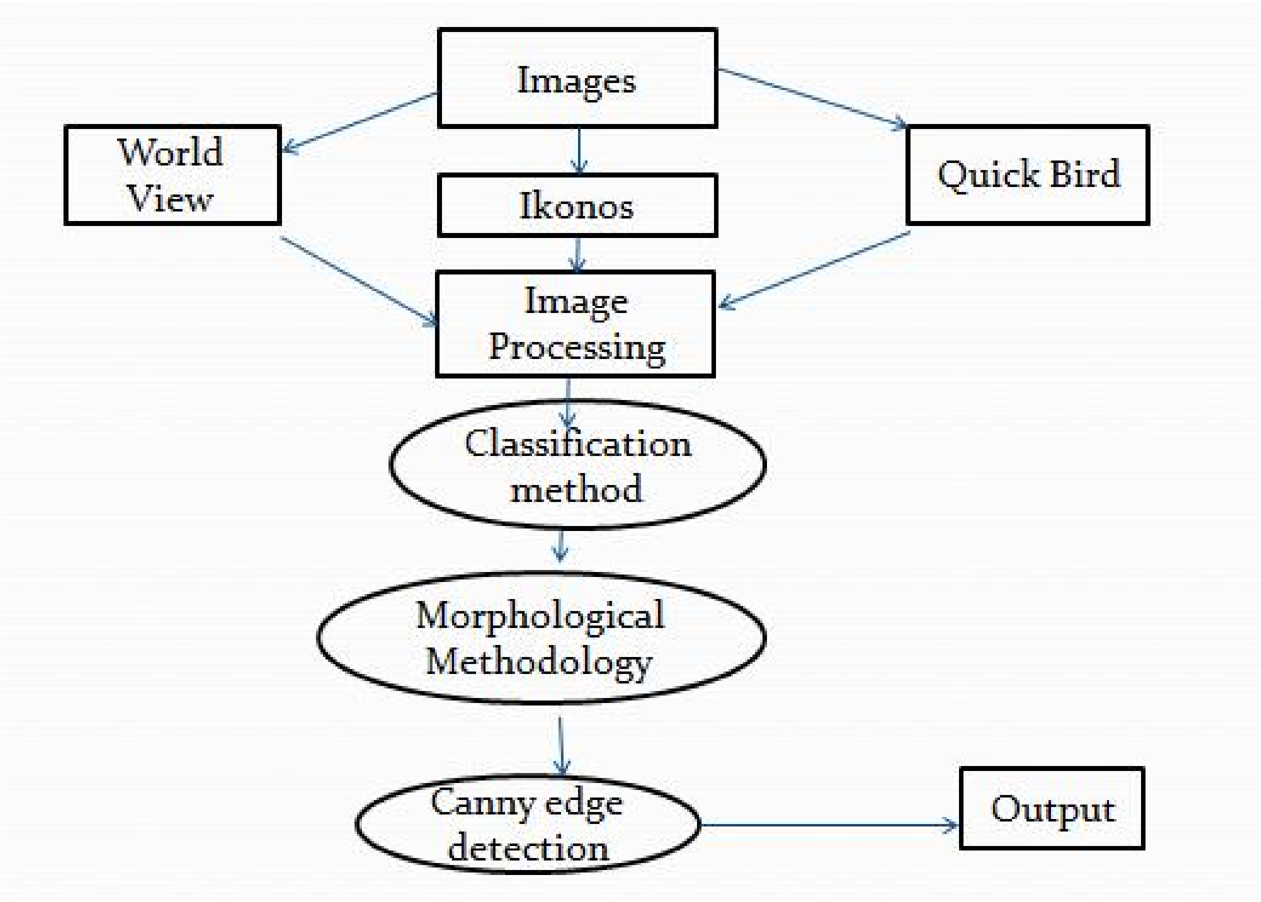
5.4.1 Level 0 Data Flow Diagram

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.



5.4.2 Level 1 Data Flow Diagram

DFD Level 1:



6. CODE

```
import cv2
# Importing opencv library
import numpy as np
#  Importing NumPy,which is the fundamental package for scientific computing with Python
img = cv2.imread('road4.jpg')
# Read the image from disk
cv2.imshow("Original image", img)  # Display image
#kernel = np.ones((5,5), np.uint8)
#img_erosion = cv2.erode(meanshift, kernel, iterations=1)
#cv2.imshow('Erosion', img_erosion)
img_float = np.float32(img)  # Convert image from unsigned 8 bit to 32 bit float
criteria = (cv2.TERM_CRITERIA_EPS+cv2.TERM_CRITERIA_MAX_ITER, 10, 1)
# Defining the criteria ( type, max_iter, epsilon )
# cv2.TERM_CRITERIA_EPS - stop the algorithm iteration if specified accuracy, epsilon, is reached.
# cv2.TERM_CRITERIA_MAX_ITER - stop the algorithm after the specified number of iterations,
max_iter.
# cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER - stop the iteration when any of
the above condition is met.
# max_iter - An integer specifying maximum number of iterations.In this case it is 10
# epsilon - Required accuracy.In this case it is 1
k = 50  # Number of clusters
ret, label, centers = cv2.kmeans(img_float, k, None, criteria, 50, cv2.KMEANS_RANDOM_CENTERS)
# apply kmeans algorithm with random centers approach
center = np.uint8(centers)
# Convert the image from float to unsigned integer
res = center[label.flatten()]
# This will flatten the label
res2 = res.reshape(img.shape)
# Reshape the image
#cv2.imshow("K Means", res2)  # Display image
#cv2.imwrite("1.jpg", res2)  # Write image onto disk

meanshif=cv2.pyrMeanShiftFiltering(img,sp=8,sr=16,maxLevel=1,termcrit=(cv2.TERM_CRITERIA_P
```

```
S+cv2.TERM_CRITERIA_MAX_ITER, 5, 1))
# Apply meanshift algorithm on to image
#cv2.imshow("Output of meanshift", meanshift)
# Display image
#cv2.imwrite("2.jpg", meanshift)
kernel = np.ones((3,3), np.uint8)
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
cv2.imshow('closinggg',closing)
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
cv2.imshow('opening',opening)
#cv2.imshow('Input', img)
# Write image onto disk
gray = cv2.cvtColor(closing , cv2.COLOR_BGR2GRAY)
# Convert image from RGB to GRAY
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
# apply thresholding to convert the image to binary
fg = cv2.erode(thresh, None, iterations=1)
# erode the image
bgt = cv2.dilate(thresh, None, iterations=1)
# Dilate the image
ret, bg = cv2.threshold(bgt, 1, 128, 1)
# Apply thresholding
marker = cv2.add(fg, bg)
# Add foreground and background
canny = cv2.Canny(marker, 110, 150)
# Apply canny edge detector
new,contours,hierarchy= cv2.findContours(canny, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
# Finding the contors in the image using chain approximation
marker32 = np.int32(marker)
# converting the marker to float 32 bit
cv2.watershed(img,marker32)
# Apply watershed algorithm
m = cv2.convertScaleAbs(marker32)
ret, thresh = cv2.threshold(m, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
# Apply thresholding on the image to convert to binary image
thresh_inv = cv2.bitwise_not(thresh)
```

```
# Invert the thresh
res = cv2.bitwise_and(closing ,closing, mask=thresh)
# Bitwise and with the image mask thresh
res3 = cv2.bitwise_and(closing,closing , mask=thresh_inv)
# Bitwise and the image with mask as threshold invert
res4 = cv2.addWeighted(res, 1, res3, 1, 0)
# Take the weighted average
final = cv2.drawContours(res4, contours, -1, (0, 255, 0), 1)
# Draw the contours on the image with green color and pixel width is 1
cv2.imshow("Canny", final)  # Display the image
cv2.imwrite("3.jpg", final)  # Write the image
cv2.waitKey()  # Wait for key stroke
```

7. TESTING

7. 1 INTRODUCTION

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Various types of testing are present: Unit testing, Integration testing, Validation testing, Black Box testing etc. We prefer unit testing for our project as the modules developed independently can be tested successfully using the code snippets and this can help us to easily rectify gaps or errors likely to be occurred during the execution of our program. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. Unit testing is preferred as it independently tests the code for possible errors and it also reuses the code rather than rendering the code to vain. Development is also fast in case of Unit testing

7. 2 TEST CASES

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement. Test Case acts as the starting point for the test execution, and after applying a set of input values; the application has a definitive outcome and leaves the system at some end-point or also known as execution post condition. A typical test case consists of Test Case ID, Test Scenario, Test Case Description, Test Steps, Prerequisite, Test Data, Expected Result, Test Parameters, Actual Result, Environment Information, Comments.

7. 3 TYPES OF TESTS

7. 3. 1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the 31 application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7. 3. 2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7. 3. 3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7. 3. 4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7. 3. 5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

7. 3. 6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot ―see‖ into it.

The test provides inputs and responds to outputs without considering how the software works.

7. 4 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7. 4. 1 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

7. 4. 2 Test objectives

All field entries must work properly. Pages must be activated from the identified link. The entry screen, messages and responses must not be delayed.

7. 4. 3 Features to be tested

Verify that the entries are of the correct format.

No duplicate entries should be allowed.

Test No.	Test Case	Expected Result	Pass
1.	Input road satellite images	Image is displayed with Edge detection.	pass
2.	If the image format is not specified correctly.	Imread error occurs and displays corresponding file doesn't exists	pass
3.	The iterated output is to be load in out picture box.	The out picture box was nicely loaded in the picture box.	pass
4.	If source path of the input files has not been specified correctly.	Displays an error showing, enter the correct input files path.	pass

Table 3: Table showing test cases

7.5 RESULTS

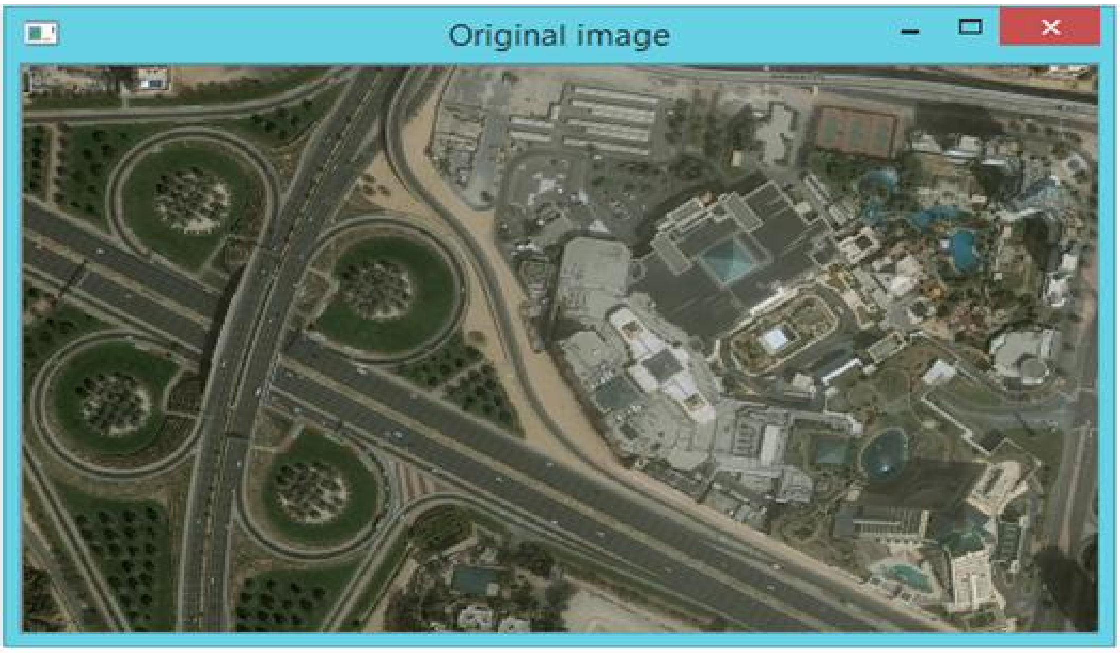


FIGURE 7.5.1 ORIGINAL IMAGE

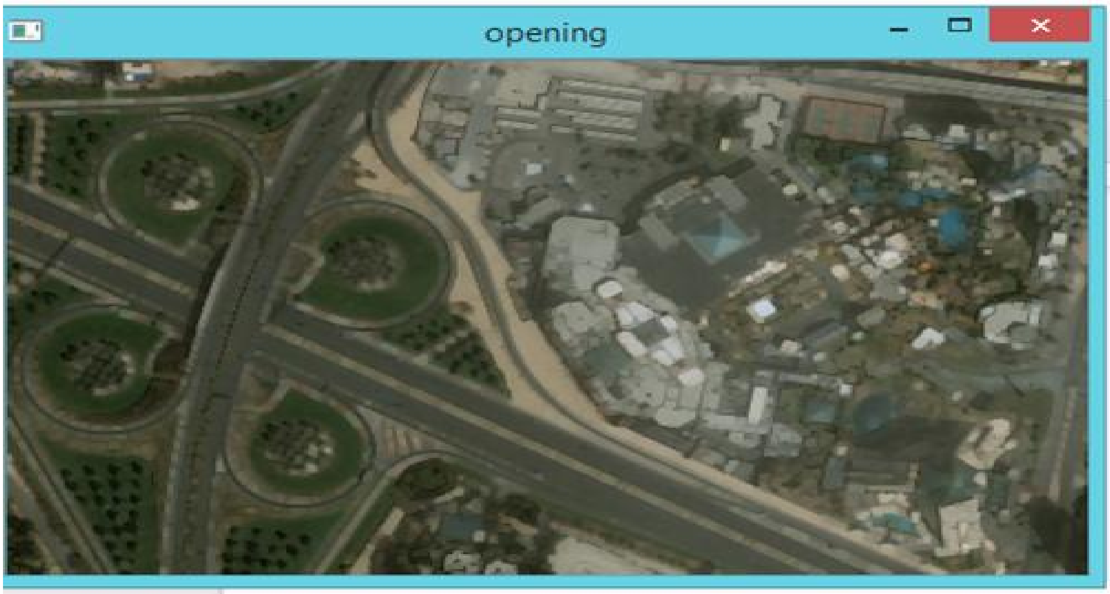


FIGURE 7.5.2 CONVERSION OF OPENING

FIGURE 7.5.3 CONVERSION OF CLOSING

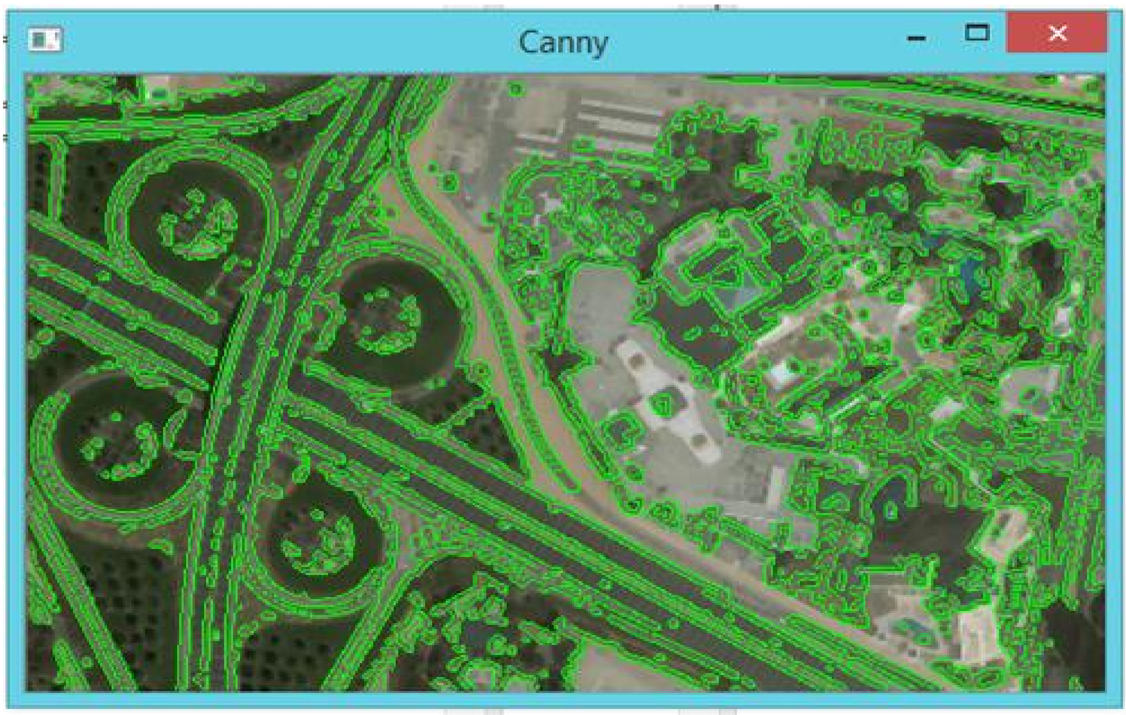


FIGURE 7.5.4 CANNY EDGE DETECTION

FIGURE 7.5.5 ORIGINAL IMAGE

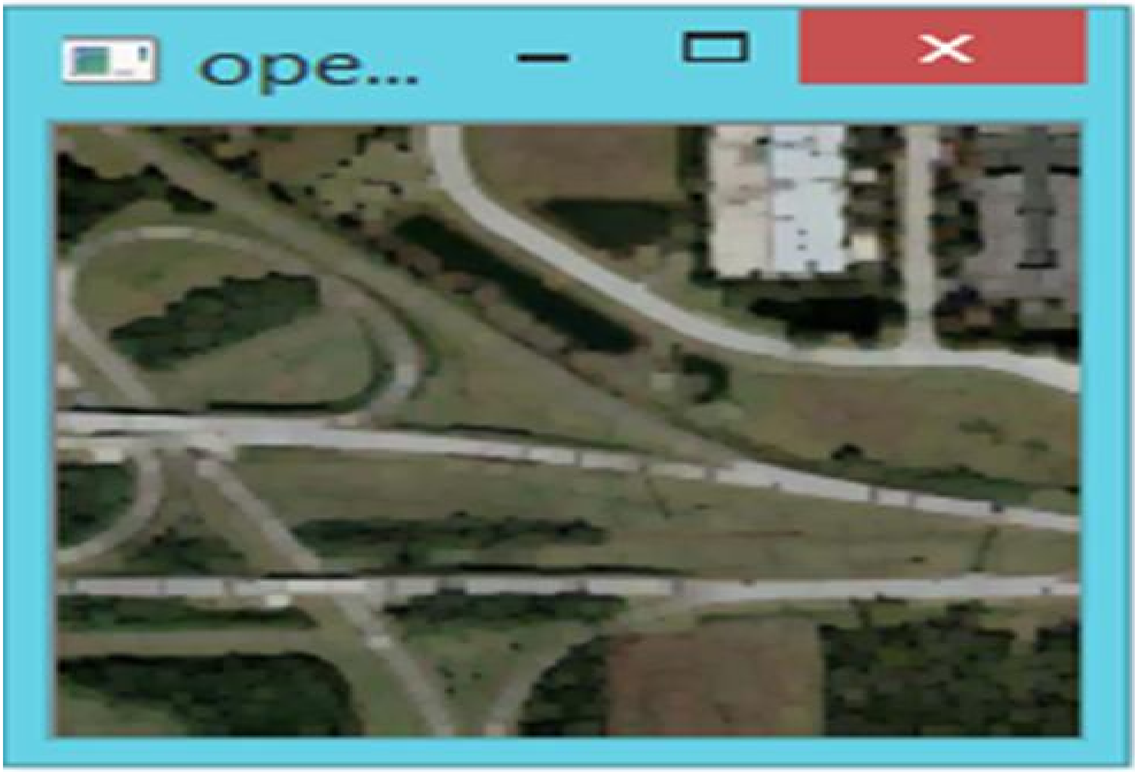


FIGURE 7.5.6 CONVERSION OF OPENING

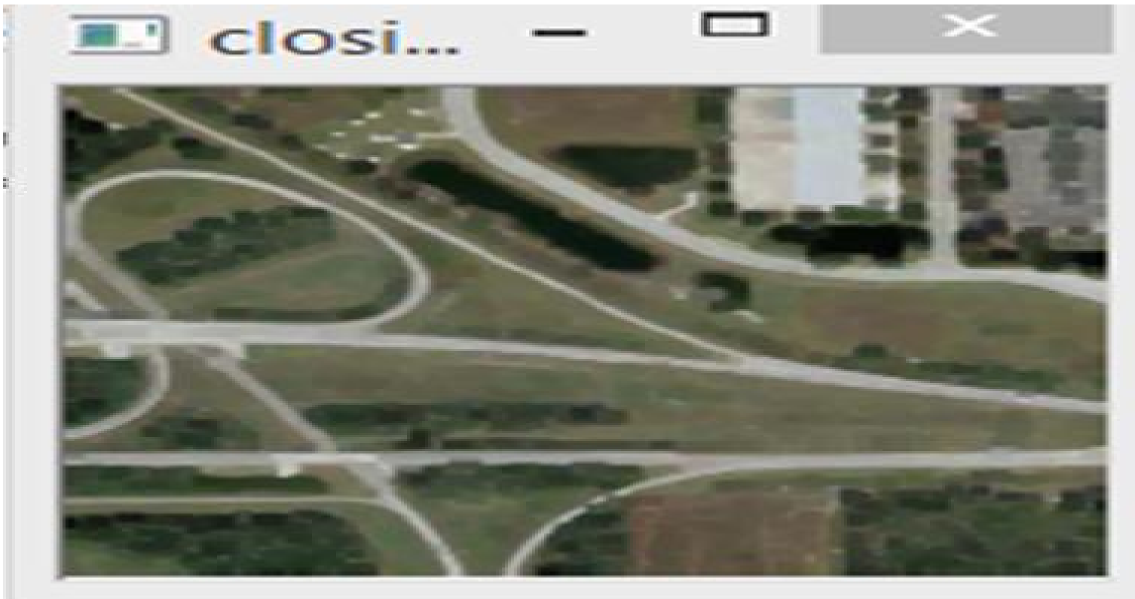


FIGURE 7.5.7 CONVERSION OF CLOSING

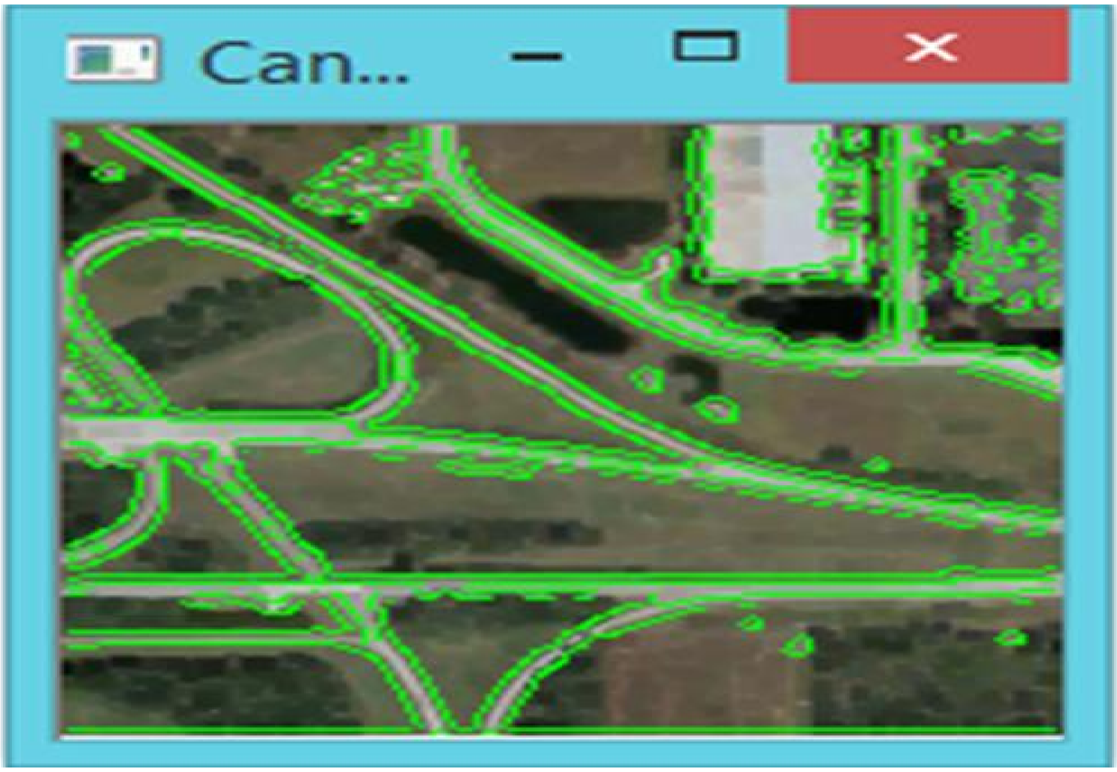


FIGURE 7.5.8 CANNY EDGE DETECTION

8.CONCLUSION

The traditional canny edge detection algorithm has capability of identifying very fine edges which cannot be used for prominent object detection. In this paper, Object identification achieved using Morphological canny edge detection in Open CV. The comparison between sobel edge detection and canny edge detection is carried out. From experimental results canny edge detection is proved as better. Canny edge detection is combined with morphological operation to identify prominent object with less number of edges and it is observed in experimental result that number of edges detected in proposed approach is less compared to simple canny edge detection. Prominent objects are detected and it can be further used for watermarking or any other application that involves object detection.

9.REFERENCES

[1] Weibin Rong, Zhanjing Li, Wei Zhang and Lining Sun, “An Improved Canny Edge Detection Algorithm”, Proceedings of 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China August 3 - 6, 2014.

[2] Cai-Xia Deng, Gui-Bin Wang , Xin-Rui Yang , “Image Edge Detection Algorithm Based on Improved Canny Operator”, Proceedings of the 2013 International Conference on Wavelet Analysis and Pattern Recognition, Tianjin, July, 2013.

[3] J. Canny, "A computational approach to edge detection", IEEE Trans. Pattern Anal. Mach. Intell. 8 (6), 1986.

[4] Road Extraction From Alos Images Using Mathematical Morphology F. S. P. de Castro a, b *, J. A. S. Centeno b

[5] Wang, L.; Qin, Q.; Du, S.; Chen, D.; Tao, J. Road Extraction from Remote Sensing Image Based on Multi-resolution Analysis. In: International Symposium on Remote Sensing of Environment (ISRSE) – Global Monitoring for Sustainability and Security, 31st, 2005, Saint Petersburg. Proceedings… Saint Petersburg: ISRSE, 2005.

[6] Zhang C.; Murai, S.; Baltsavias, E. Road Network Detection by Mathematical Morphology. IAPRS - International Archives of Photogrammetry and Remote Sensing, Paris, p. 185-200, 1999.

[7] Radhika Chandwadkar, Saurabh Dhole, Vaibhav Gadewar, Deepika Raut, Prof. S. A. Tiwaskar “Comparison Of Edge Detection Techniques” Proceedings Of Sixth Iraj International Conference, 6th October 2013, Pune.

[8]Jamie Shotton, Andrew Blake, Roberto Cipolla, “Contour-Based Learning for Object Detection”, Tenth IEEE International Conference on Computer Vision, 2005.

[9]Reza Oji, “An Automatic Algorithm For Object Recognition And Detection Based On Asift Keypoints”, Signal & Image Processing: An International Journal (SIPIJ), Vol. 3, No.5, October 2012.

[10]Wenshuo Gao, Lei Yang, Xiaoguang Zhang, Huizhong Liu “An Improved Sobel Edge Detection”, Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on (Volume: 5)