N° d'ordre : ..../..../....

الجمهـــــورية الجـــزائرية الديمـــقراطية الشعـــــــبية

République Algérienne Démocratique et Populaire

وزَارة التـــعليم العالي وَ البـــحث العـلمي

Ministere de l'Enseignement Superieur et de la Recherche Scientifique

جـامعة الشهيد حمة لخضر الوادي

Universite Chahid Hamma Lakhdar d'El-Oued

كليـة العلـوم الدّقيقـة

Faculte des Sciences Exacts

قسـم الإعـلام الآلـي

Departement D'Informatique

THÈSE

Présentée en vue de l'obtention du diplôme de **Doctorat**

3ème cycle (LMD) en Informatique

Option : système d'informations interopérables

Par : **Ayoub Benchabana**

Titre

Conception et mise en œuvre d'une plateforme de traitement d'images satellitaires et aériennes pour la mise à jour des objets spatiaux d'un système d'information géographique (SIG) destiné au suivi des changements urbains

Soutenue publiquement le : 13/01/2024

devant le jury composé de:

Président	*Pr. Mohammed Charef Eddine MEFTAH*	*Université de Eloued*
Directrice de la Thèse	*Pr. Mohamed-Khireddine KHOLLADI*	*Université de Eloued*
Examinateur	*Pr. Mohamed Rédha LAOUAR*	*Université de Tébessa*
Examinateur	*Dr. Oussama AIADI.*	*Université de Ouargla*
Examinateur	*Dr. Soltane MERZOUG*	*Université de Tébessa*
Examinateur	*Dr Abdelkamel BENALI*	*Université de Eloued*

Année Universitaire: 2023-2024

N° d'ordre : ..../..../....

الجمهـــــورية الجـــزائرية الديمـــقراطية الشعـــــــبية

Democratic And Popular Republic Of Algeria

وزَارة التـــعليم العالي وَ البـــحث العـلمي

Ministry Of Higher Education And Scientific Research

جـامعة الشهيد حمة لخضر الوادي

University of Chahid Hamma Lakhdar El-Oued

كليـة العلـوم الدّقيقـة

FACULTY OF EXACT SCIENCES

قسـم الإعـلام الآلـي

COMPUTER SCIENCE DEPARTMENT

THESIS

Presented with a view to obtaining the Doctoral degree 3rd cycle (LMD) in Computer Science

Option : interoperable information system

By : Ayoub Benchabana

Title

Design and implementation of a satellite and aerial image processing platform to update spatial objects of geographic information system (GIS) for monitoring urban changes

Publicly defended on : 13/01/2024

in front of the jury composed of :

President	*Pr. Mohammed Charef Eddine MEFTAH*	*University of Eloued*
Supervisor	*Pr. Mohamed-Khireddine KHOLLADI*	*University of Eloued*
Examiner	*Pr. Mohamed Rédha LAOUAR*	*University of Tébessa*
Examiner	*Dr. Oussama AIADI.*	*University of Ouargla*
Examiner	*Dr. Soltane MERZOUG.*	*University of Tébessa*
Examiner	*Dr Abdelkamel BENALI*	*University of Eloued*

Academic year: 2023-2024


بسم الله الرحمن الرحيم
وقل رب زدني علما
صدق الله العظيم

I dedicate this work to…

My self,

To all those who were giving me any kind of support,

Especially, to my Mother,

To my Wife,

To all my family,

Ayoub Benchabana

Acknowledgment

First, I thank ALLAH the Almighty for allowing me to reach this modest scientific level and for giving me the courage and patience to carry out the work in this thesis.

It has been an auspicious journey for me to arrive at this point. I am really short of words now, but I am taking this opportunity to express my sincere thanks to my supervisor, ***Pr. Mohamed Khireddine Kholladi,*** *for accepting to be in charge of my thesis and for his incontestable instructions, patience, skills, and remarks that have benefited me.*

I would like to sincerely thank ***Dr. Ramla Bensaci and Dr. Belal Khaldi****, who had a great virtue in accomplishing this thesis. Thank you, Dr. Ramla Bensaci, for your support, patience, and encouragement while realizing this work.*

I want to thank all the jury members who have agreed to preside and read this work. ***Pr. Mohammed Charef Eddine MEFTAH*** *from the University of Eloued as a jury president,* ***Pr. Mohamed Rédha LAOUAR*** *from the University of Tébessa,* ***Dr. Oussama AIADI*** *from the University of Ouargla,* ***Dr. Soltane MERZOUG*** *from the University of Tébessa and* ***Dr. Abdelkamel BENALI*** *from the University of Eloued as a jury Examiners.*

Finally, my thanks go to all those who contributed in any way to the outcome of this work.

Abstract

Object detection and identification from remotely sensed data, especially Buildings, can be considered the foundation of updating Geographic Information system data for improving and monitoring the infrastructure of cities. Small-scale objects like buildings may now be recognized because of the development of extremely high-resolution remote-sensing images. However, manually separating the buildings from the images requires substantial processing time. Comprehensive research has been conducted on various approaches, including traditional image processing techniques, supervised and unsupervised machine learning methods, and deep learning architectures. The literature survey explores the evolution of feature extraction algorithms, classification models, and their applications in urban environments. Notable studies on semantic segmentation, object-based image analysis, and multi-sensor data integration are discussed. Additionally, insights are provided into the challenges and limitations faced by current techniques, paving the way for the development of novel strategies proposed in this thesis. The synthesis of this extensive review establishes a foundation for the research, highlighting gaps in the current state-of-the-art and setting the context for the proposed advancements in automated building detection.

Therefore, a robust building detection methodology is necessary. We present two approaches for extracting buildings from high-resolution images. The first approach is based on a supervised machine learning technique, and for the second, we use deep learning methods.

In the supervised approach, the image is firstly divided into superpixel patches, from which the colors and texture features are retrieved. Buildings, roads, trees, and shadows are then separated into four groups using the Support Vector Machines technique (SVM). The approximate location of the building has been determined using a seed point start and an adaptive regional growth approach based on the previously known position of the shadows. A contouring procedure involving an open morphological operation was applied to extract the final shape of buildings.

The second method involves four main steps: homogeneous superpixel image segmentation through an altered Simple Linear Iterative Clustering (SLIC), extensive feature extraction via a variational auto-encoder (VAE) adjust on the superpixels for training and testing data collection, classification of four classes (buildings, roads, trees, and shadows) utilizing extracted feature data as feedback to a Convolutional Neural Network (CNN), and extraction of building forms by morphological processes and regional growth.

Our innovative methods have excellent accuracy rates for identifying building units. Accuracy assessment over different study areas shows the advantage of our novel approach, making it a robust, realistic, and accurate tool.

Keywords: *Building detection, Image processing, Feature extraction, Machine Learning, Deep Learning, Remote sensing, Geographic Information System.*

Résumé

La détection et l'identification des objets à partir des données de télédétection, particulièrement les constructions, peuvent être considérées comme le fondement de la mise à jour du système d'information des données géographique pour l'amélioration et la surveillance des infrastructures des villes.

Les objets à petite échelle comme les bâtiments, désormais grâce au développement d'images de télédétection, peuvent être identifiés à très haute résolution. Cependant, la séparation manuelle des constructions des images nécessite un temps de traitement très conséquent. Des recherches approfondies ont été menées sur diverses approches, notamment les techniques traditionnelles de traitement d'images, les méthodes d'apprentissage automatique supervisé et non supervisé et les architectures d'apprentissage profond. L'étude de la littérature explore l'évolution des algorithmes d'extraction de caractéristiques, des modèles de classification et de leurs applications en environnements urbains. Des études notables sur la segmentation sémantique, l'analyse d'images basée sur les objets et l'intégration de données multi-capteurs sont discutées. De plus, des informations sont fournies sur les défis et les limites rencontrés par les techniques actuelles, ouvrant la voie au développement de nouvelles stratégies proposées dans cette thèse. La synthèse de cette étude approfondie établit une base pour la recherche, mettant en évidence les lacunes de l'état de l'art actuel et définissant le contexte des avancées proposées dans la détection automatisée des bâtiments.

Pour cette raison, l'élaboration d'une nouvelle méthodologie robuste pour la détection des constructions est nécessaire. Nous présentons deux approches pour extraire des bâtiments à partir d'images haute résolution. La première approche est basée sur une supervisée technique de l'apprentissage automatique, et pour la seconde, nous utilisons des méthodes d'apprentissage profond.

Dans l'approche supervisée, l'image est d'abord divisée en patchs de superpixels, à partir desquels les couleurs et les caractéristiques de texture sont récupérées. Les bâtiments, les routes, les arbres et les ombres sont ensuite séparés en quatre groupes à l'aide de la technique des machines à vecteurs de support (SVM). L'emplacement approximatif du bâtiment a été déterminé à l'aide d'un point de départ et d'une approche de croissance régionale adaptative basée sur la position précédemment connue des ombres. Une procédure de contournage a été appliquée pour extraire la forme finale des bâtiments impliquant une opération morphologique ouverte..

La deuxième approche consiste en quatre étapes principales : segmentation de l'image en superpixels homogènes à l'aide d'un clustering itératif linéaire simple (SLIC) modifié, extraction approfondie des caractéristiques à l'aide d'une échelle d'auto-encodeur variationnel (VAE) sur les superpixels pour la formation et le test de la collecte de données,

identification de quatre classes (bâtiments, routes, arbres et ombres) à l'aide de données d'entités extraites en entrée d'un réseau neuronal convolutif (CNN), et extraction de formes de bâtiments par croissance régionale et opérations morphologiques.

Notre méthodes innovante donne d'excellents taux de précision dans le cadre d'identification des unités de construction. L'évaluation de la précision sur différents domaines d'étude démontre les avantages qu'on obtient avec cette nouvelle approche, ce qui permet de la positionner comme un outil robuste, réaliste et précis.

Mots clés: *Détection de bâtiments, Traitement d'images, Extraction de caractéristiques, Apprentissage automatique, Apprentissage profond, Télédétection, Système d'information géographique..*

الملخص

يمكن اعتبار اكتشاف العناصر وتحديدها من البيانات المستشعرة عن بعد وبالأخص المباني، من بين الخطوات الأساسية لتحديث بيانات أنظمة المعلومات الجغرافية لتحسين ومراقبة البنية التحتية للمدن وكيفية توسعها. في الوقت الراهن وبسبب التطور السريع في الصور الملتقطة عن بعد إلى صور بدقة عالية جداً، يمكننا التعرف على العناصر ذات أحجام صغيرة مثل المباني، ومع ذلك يستغرق تحديد المباني بشكل يدوي وتمييزها من الصور الملتقطة وقتًا طويلاً، وعليه تم إجراء بحث شامل حول أساليب مختلفة، بما في ذلك تقنيات معالجة الصور التقليدية، وطرق التعلم الآلي الخاضعة للإشراف وغير الخاضعة للإشراف، وبنيات التعلم العميق. بدراسة للأبحاث السابقة تظهر تطور خوارزميات استخراج الميزات ونماذج التصنيف وتطبيقاتها في البيئات الحضرية. وتناقش الدراسات البارزة حول التجزئة الدلالية، وتحليل الصور المستندة إلى الكائنات، وتكامل بيانات أجهزة الاستشعار المتعددة. بالإضافة إلى ذلك، يتم تقديم رؤى حول التحديات والقيود التي تواجهها التقنيات الحالية، مما يمهد الطريق لتطوير استراتيجيات جديدة مقترحة في هذه الأطروحة. يضع توليف هذه المراجعة الشاملة أساسًا للبحث، حيث يسلط الضوء على الثغرات في الوضع الحالي ويحدد سياق التطورات المقترحة في الكشف الآلي عن المباني.

لذلك هناك حاجة إلى نظام فعال وتلقائي لتحديد المباني من بين بقية العناصر. نقدم في هذا العمل مقاربتين جديدتين لاستخراج المباني من الصور عالية بالإعتماد في المنهج الأول على تقنية التعلم الآلي تحت الإشراف، أما في الثاني فيتم الإعتماد على التعلم العميق.

في تقنية التعلم تحت الإشراف يتم فيها أولاً تقسيم الصورة إلى قطع تحتوي على مجموعة من البكسلز ويتم من خلالها استخراج مميزات الألوان والملمس. بعد ذلك تصنف إلى أربع مجموعات وهي: المباني، الطرق، الأشجار والظلال باستخدام تقنية التعلم الآلي *(SVM)*. تحدد المواقع التقريبية للمباني باستخدام نقط بداية ونهج النمو الإقليمي التكيفي بناءً على المعرفة المسبق للإتجاه للظلال. بعدها تطبق إجراء تحديد الحدود لاستخراج الشكل النهائي للمباني التي يتم ضمنها سد الفراغات والشوائب في أشكال المبان.

أما في التعلم العميق فهو مكون من أربع خطوات رئيسية: تجزئة الصورة إلى وحدات بكسل فائقة ومتجانسة باستخدام مجموعة تكرارية خطية بسيطة معدلة (*SLIC*)، استخراج ميز معمقة باستخدام مقياس التشفير التلقائي المتغير (*VAE*) على وحدات البكسل الفائقة من أجل التدريب واختبار البيانات، بعدها يتم تحديد أربع فئات (المباني والطرق والأشجار والظلال) باستخدام بيانات المعالم المستخرجة من الخطوة السابقة كمدخلات لشبكة عصبية تلافيفية (*CNN*)، وأخيراً استخراج أشكال المباني من خلال النمو الإقليمي والعمليات المورفولوجية.

تتميز طريقتنا المبتكرة بمعدلات دقة ممتازة لتحديد الأبنية ويظهر ذلك من خلال تقييم الدقة المتحصل عليها في مناطق دراسة مختلفة مما يجعله نظام قوي، واقعي وفعال.

الكلمات المفتاحية: اكتشاف المباني ، معالجة الصور، استخراج الميزات، التعلم الآلي، التعلم العميق، الاستشعار عن بعد، نظام المعلومات الجغرافية.

Table of Contents

List of Figures

List of Tables

Abbreviations

AE : Auto-Encoder

ANNs : Artificial Neural Networks

AUT : Authenticity Of Detection

BDSV : Building Detection with Shadow Verification

CCM : Color Co-Occurrence Matrix

CIELAB: Commission Internationale Eclairage l*a*b

CMYK : Cyan, Magenta, Yellow, And Key (Black)

CNN: Convolutional Neural Networks

CRF : Conditional Random field

DBMS : Database Management System

DCT : Discrete Cosine Transform

DEM : Digital Elevation Model

DL : Deep Learning

DNNs : Deep Neural Networks

DSM : Digital Surface Model

FD: Fractal Dimension

FN : False Negative

FNR : False-Negative Rate

FP :False Positive

FT : Fourier Transform

GIS : Geographic Information System

GUI : graphical user interface

HSL : Hue-Saturation-Lightness color model

HSV : Hue, Saturation, and Value color model

ICICM :Integrative Color Intensity Co-occurrence Matrix

k-NN : k-Nearest Neighbor

LDA : Linear discriminate analysis

LIDAR : Light Detection and Ranging

LoG : Laplacian of Gaussian

ML : Machine learning

MRF : Markov Random Field

NASA : National Aeronautics and Space Administration

P: Precision

R:Recall

RDBMS : Relational Database Management System

RGB : Red, Green, And Blue Colors

RS : Remote sensing

SAR : Simultaneous Autoregressive

SLIC : Simple Linear Iterative Clustering

SQL : structured query language

SVM : Support Vector Machine

TIN : Triangulated Irregular Networks

TP : True Positive

VAE : Variational AutoEncoder

YIQ or YUV : (Y) luma, or brightness, (U) blue projection and (V) red projection

General introduction

General introduction

Urban growth is accompanied by the proliferation of information describing the urban territory and those who inhabit it: location of activities, means of transport, equipment, land and heritage management, etc. All this information carries, in one way or another, a location attribute, whether it is accessible in the form of precise identification or the form of aggregation available according to predefined divisions. Their visualization in the form of plans or maps remains one of the simplest ways of apprehending them. A process of rationalizing storage and access to localized data must be implemented to handle information that is becoming more complicated and to see its dynamics and extension on a territorial scale. Currently, one of the best possible choices for organizing the city's technical and social information is called the Geographic Information System (GIS).

A Geographic Information System is a computer system that captures, stores, manipulates, analyzes, manages, and presents all spatial or geographical data types (Blakemore and Chorley, 1988). GIS may assist people and organizations in better understanding geographical patterns and relationships by connecting seemingly unconnected data.

Though many of the geographic techniques and concepts that GIS automates have existed since the 19th century, the first to originate the term "geographic information system" was Roger Tomlinson in 1969 (Tomlinson, 1969). Afterward, at the end of the 20th century, Users started to consider accessing GIS data over the Internet due to the fast rise in multiple systems being integrated and standardized on a few platforms, requesting data format and transfer standards. On the other hand, remote sensing provides another tool that can be integrated into a GIS. Remote sensing includes imagery and other data collected from satellites, aircraft, and drones. Ultimately, it was also rapidly developing, introducing a revolution in the description of the earth and updating its occupation. The evolution of sensors towards very high resolution, the diversification of image types (panchromatic, color, infrared, radar), and the increase in their number are progressively multiplying their fields of application, such as earth science, environmental protection,

cartography, and urban planning. In this thesis, we are interested in high-resolution spatial images of urban areas. These visually provide access to objects (buildings, trees, cars, parking lots, etc.) that are perceived individually.

Remote sensing data has helped track and spot urban change and provides crucial knowledge for upcoming development. In the past, scientists and planners produced land use maps manually by interpreting aerial pictures. Examining large amounts of aerial imagery by hand is expensive and time-consuming. With a resolution of up to 100 pixels per square meter, such imaging has significantly expanded the range of potential applications, but at the price of increasing the required manual processing. However, with the advancement of remote sensing technology and the accessibility of modern high-resolution digital images, it is now feasible to construct land use maps in a more timely, less expensive, and more accurate manner. While significant progress has been made in the past years, only a few semi-automated systems that work in limited domains are in use today.

Recent large-scale machine learning applications to such high-resolution imagery have generated object detectors with unprecedented precision levels. The analysis of aerial images in machine learning applications is typically designed as a pixel labeling problem. The objective is to either construct a complete semantic segmentation of the image into classes like buildings, roads, trees, grass, and water or a binary image classification for a specific object class.

Although image labeling or parsing of general scenes has been widely researched, aerial images have unique features that make them easier to classify. Initially, we can assume that objects' perspective and scale are constant by limiting our use of aerial images with a known ground resolution. Moreover, compared to the datasets available for general image labeling tasks, the quantity of labeled and unlabeled aerial imagery is enormous. Approaches that can effectively learn from large volumes of labeled data should have a clear advantage over methods that cannot be handled when labeling aerial images.

Building detection lies in addressing crucial challenges and contributing to various fields. It aids urban planners in understanding the spatial distribution of structures, allowing for more informed decision-making in city development and infrastructure planning. Accurate building detection is crucial for efficient emergency response in

disaster-prone areas. It facilitates quick identification of affected areas and helps allocate resources effectively. Monitoring and analyzing building patterns contribute to assessing environmental impacts, such as deforestation, urban sprawl, and changes in land use, assisting in sustainable development efforts. With advancements in remote sensing technologies, including satellite imagery and aerial photography, there is an opportunity to explore and implement sophisticated algorithms for improved building detection.

The study's objective is to develop a new method that is highly successful in labeling aerial or satellite images. Furthermore, This thesis mainly focuses on Two primary problems we face when using image labeling approaches with aerial images:

- Background and Features: Because local color indications fail to effectively distinguish between pairs of object classes like roads and buildings or trees and grass, identifying aerial image labels requires using the background context. Additionally, shadows caused by trees and buildings can assist in determining their locations by knowing the direction of the sunlight in advance. Since the number of pixels increases in a high-resolution input image, the number of parameters and the amount of computation required also increase. That being the case, innovative approaches for extracting discriminative features from a broad image are required for aerial image labeling.

- Results outputs: Strong correlations exist between the labels of neighboring pixels in an image, and effectively utilizing this structure can considerably increase labeling accuracy. However, noisy labels still appear due to several factors; one of the causes is the difference in color shades on the building's rooftops.

The main contribution of this thesis is a systematic structure of two novel approaches to extracting buildings in high-resolution images. The first is characterized by its speed in calculations and implementation without needing high-performance equipment and can be adapted to address specific problems. The second has a much higher accuracy and identification ratio, even in highly complex areas. Moreover, the accuracy assessment of both approaches over different study areas shows the advantage and superiority in performance, primarily due to the advantage of using full superpixels instead of single normal pixels without losing information; in addition to the structure of the system we proposed for data classification with the use of an adaptive regional growth method for defining final building shapes, making both approaches robust, realistic, and accurate tools.

The rest of the thesis is organized as follows:

- Chapter 1 briefly describes the geographic information system, components, function, and data representation.
- Chapter 2 explains general concepts of remote sensing and high-resolution earth observation technologies.
- Chapter 3 provides a basic understanding of the image processing methods employed throughout this work.
- Chapter 4 exhibits a brief description of the machine learning field.
- Chapter 5 illustrates our novel building detection shadow-based approach methodology and implementation with analysis and results.
- Finally, a general conclusion summarizes the results obtained by our approach with some unresolved problems and future research projects.

Chapter 1

Geographic Information System

Chapitre 1

Geographic Information System

1.1 Introduction

The proliferation of approaches to constituting and processing geographic information is accompanied by a proliferation of tools capable of structuring and processing it; they are grouped under the term GIS. As stated before, a geographic information system is a computer system designed to capture, store, manipulate, analyze, manage, and present all spatial or geographical data types.

A geographic information system is a type of information system made to operate with data referred to by coordinates. In other words, a GIS is both a database system with specialized capabilities for spatially-referenced data and a collection of procedures for dealing with data. A GIS might be seen as a more advanced map in certain ways (‘Book Review’, 1991).

Applications for GIS use both software and hardware. These applications might use digital, photographic, spreadsheet, or geographic data. Moreover, Remote sensing provides another tool that can be integrated into a GIS. Images and other data from satellites, balloons, and drones are included in remote sensing. GIS technologies allow the merging these many sorts of information on a single map, regardless of their origin or initial format. GIS's primary index variable to connect these disparate data points is location.

GIS allows for comparing and analyzing a wide range of information types. The system may contain information on individuals, such as population, income, or education degree. It can also show data details on the topography, such as the locations of streams, various plant species, and soil types—information on the locations of schools, hospitals, farms, roads, and electric power lines.

A GIS file format can be either raster or vector as the primary forms. Cell or pixel grids make up raster formats. Raster formats are practical for storing GIS data that changes, such as altitude or satellite imagery. Polygons created using nodes and lines are known as vector formats. The storage of GIS data with distinct borders, such as school districts or streets, is made possible via vector formats.

1.2 The elements of GIS

Five essential elements must be integrated for a GIS to function: hardware, software, data, methods, and users that establish a fundamental framework upon which every type of geographically referenced information may be assembled (Ali, no date). The five elements are more thoroughly discussed in the following section.

Figure 1.1: the five elements of GIS (Mierzejowska and Pomykoł, 2019).

1.2.1 Hardware

The first component of the GIS is the computer hardware that stores and allows access to the GIS data and software programs. Depending on organizational demands, hardware specifications might change significantly. Centralized servers, laptops, desktop computers, scanners, printers, and other devices may be connected over the intranet in secure facilities. Other businesses may use a GIS via high-speed internet-connected devices to improve communication and collaboration between headquarters and remote sites. GIS technologies are increasingly used on smartphones, tablets, and other mobile computing devices.

1.2.2 Software

Another crucial element of geographic information systems is computer software. The capacity to store, analyze, and visualize GIS data would not be feasible without GIS software applications or apps. The database management system and a graphical user interface (GUI) or dashboard with menu choices are important GIS software programs. It lets users digitize, store, manage, and query GIS data, carry out complicated analyses, and create reports, charts, maps, globes, and other eye-catching data-driven presentations.

1.2.3 Data

An equally important component of GIS is data, containing geographical information and an attribute or related textual information. GIS's robust analytical, problem-solving, and visualization capabilities are built on integrating geographical data with relevant attribute data. Businesses can gather and digitize the data they use internally. Third-party suppliers also provide commercial data resources for sale.

1.2.4 Methods

Above all else, a good GIS maintains a well-thought-out plan and business rules, which are the models and operational procedures appropriate to each organization. There are several methods for creating maps and using them in subsequent projects. Maps may be created manually using the scanned images or automatically using a raster to vector generator. These digital maps can either come from satellite images or maps created by any surveying organization.

1.2.5 Users

Users are the most critical element of an excellent geographic information system. GIS technology serves little use if there is no human need for the real-world problem solutions that it offers. Fortunately, GIS technology is vital to almost every business on earth, applicable to them, and maintains a broad spectrum of people employed. Analysts, other professionals who regularly use GIS in their work, and groups of technical experts who create, implement, and manage geographic information systems are all examples of people who use GIS.

1.3 Functions of GIS

GIS software carries out six primary activities: input data, manipulation, management, query and analysis, and visualization.

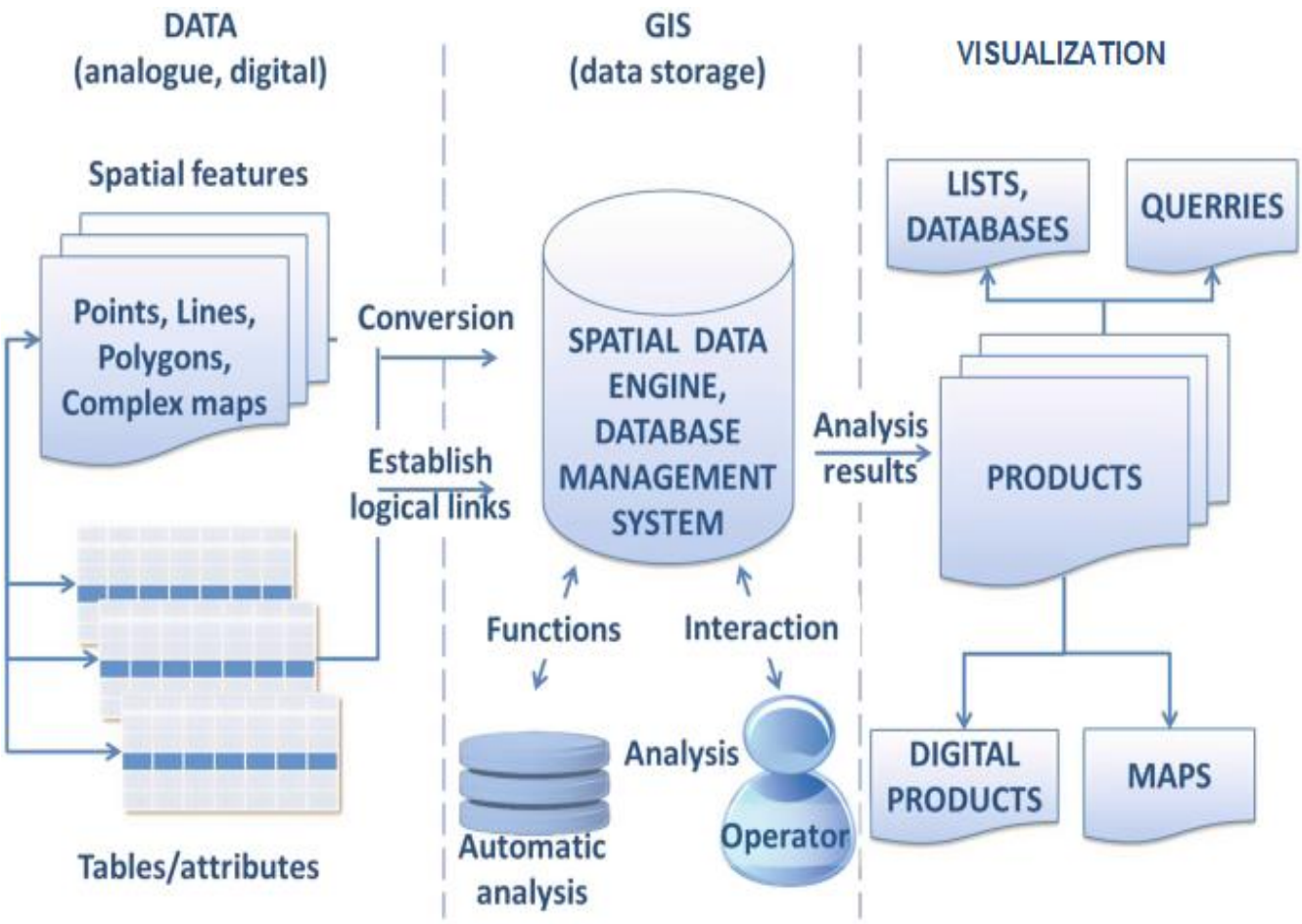


Figure 1.2: The general scheme of the GIS (Bobrowsky and Marker, 2018)

1.3.1 Input data

Digitized maps, images, spatial data, and tabular data are crucial inputs for every GIS. The relational database management system software is typically used to type tabular data on a computer. Before being utilized in a GIS, geographic data must be translated into an appropriate digital format. For example, the DBMS system can generate indexes on data objects to speed up information retrieval from a query. A vector format, which encodes the actual map points, lines, and polygons as coordinates, may be used to replicate maps digitally. A raster data presentation, in which data components are kept as cells in a grid pattern, may also be used to enter data.

The process of digitizing involves transferring information from paper maps into digital files. Large projects can fully automate this procedure with today's GIS technology; more minor works might need some human digitization. It is preferable to use the existing data because digitizing requires a lot of work and time. There are many different kinds of geographic data available now in GIS-compatible formats. These data may be acquired from providers and put into a GIS.

1.3.2 Manipulation

GIS is capable of storing, distributing, and updating text-based geographic data. Latitude and longitude are the geographic coordinate systems to which the spatial data must be referred. Database management software may alter the tabular data related to spatial data. It is likely that to make some data types compatible with the system; they will need to be changed or otherwise altered. For instance, several scales of geographic information are provided (scale of 1:100,000; 1:10,000; and 1:50,000). These must be scaled down to the same level before they can be combined and layered. It may be necessary to undergo a momentary alteration for exhibition or a permanent one for exploration. Furthermore, other additional kinds of data modification are frequently carried out in GIS. These consist of adjustments to projections, data aggregation, generalization, and data cleaning.

1.3.3 Management

Storing geographic data as computer files for minor GIS applications may be sufficient. However, it is advisable to employ a database management system to store, organize, and manage data when data quantities increase, and there are several data

consumers. A database management system, or DBMS, is a set of software tools used to manage a database's integrated data collection, including tables, indexes, queries, and other processes.

Although there are many alternative DBMS models, the relational model DBMS will benefit GIS applications. According to the relational model, data are conceptually kept as tables, each containing the data characteristics of a single shared object. Relations are used to join various tables together by utilizing their shared fields. The relational DBMS software has been widely used because of its straightforward architecture. These have a flexible structure and have been widely used in applications inside and outside GIS.

1.3.4 Query

Using structured query language(SQL), the stored information, whether geographical or related tabular data, may be accessed. Data can be queried using SQL or a menu-driven system to get map data, depending on the kind of user interface. Simple and complex searches using several data layers can give officials and analysts fast information to understand the issue and make a better-educated choice comprehensively.

1.3.5 Analysis

Geographic analysis, also known as spatial analysis or geo-processing, utilizes the geographic characteristics of features to seek patterns and trends and create "what if" scenarios. Various analytical solid tools are available in modern GIS for data analysis. The analyses typically done on geographic data include some of the ones listed below.

a. Overlay Analysis

The overlay is a technique used to integrate many data layers. This might be a visual action at its most basic, but analytical activities need to join one or more data layers physically. This overlay, also known as a spatial join, can incorporate information on soils, slopes, vegetation, and even land ownership. For instance, data layers for soil and land use may be integrated to create a new map with information on soil and land use. This will assist in understanding how the scenario behaves differently depending on various aspects.

b. Proximity Analysis

GIS software can also support the production of new polygons using points, lines, and polygon feature data stored in the database. For instance, you find the answer to the following question: How much area is covered within 1 km of the water canal? What portion of the land is planted with each crop? Furthermore, where is the border or distinction of the watershed, slope, water channels, water harvesting facilities needed, etc.?

1.3.6 Visualization

GIS may offer hardcopy maps, statistical summaries, modeling solutions, and graphical map displays for geographical and tabular data. The easiest way to see the outcome of several different geographic operations is as a map or graph. Geographical information may be stored and transmitted using maps quite effectively. GIS offers customers brand-new, innovative technologies that enhance the art of output information visualization.

1.4 GIS data representation

GIS digitally represents geographic features in real-world locations to store them in a database. Real-world elements can be classified into discrete elements (like a house) and continuous fields (streams, streets). In a GIS, there are primarily two ways to store data for both abstract concepts: Raster and Vector.

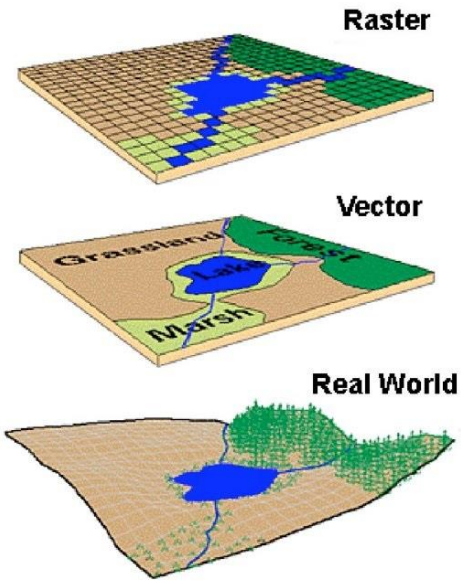


Figure 1.3: Raster and Vector data representation (Jones, 2003)

1.4.1 Raster

In principle, any digital image is a raster data type. The tiniest component of an image, the pixel, is easily recognizable to anyone experienced with digital photography. In contrast to the often-used mountable vector graphics that form the foundation of the vector structure, a conjunction of these pixels will result in an image. The raster data type in a photograph or work of art converted to a computer will reflect an abstraction of reality. At the same time, The output of a digital image is considered to be a depiction of reality. One often used type of raster data is aerial photography, which is solely used for digitizing or showing a detailed image on a map. Further raster data sets will have info on height, a DEM (Digital Elevation Model), or the reflection of a specific wavelength of light.

Each cell in a row or column of a raster data type can have a single value. Images that include raster data can have pixels with individual color values. Each cell may also include additional values that are discrete (like land use), continuous (temperature), or null (no data are obtainable). Whereas a raster cell only holds one value, it may be expanded rendering with raster bands RGB (red, green, and blue) colors, colormaps (which convert a theme code to an RGB value), alternatively, a more extended attribute table, containing a row for each unique value of a cell. The cell width in ground units indicates the resolution of the raster data collection.

Raster data can be saved in various forms, such as the common file-based TIF and JPEG formats, or directly in a relational database management system (RDBMS), similar to other feature classes based on vectors. When correctly indexed, database storage often enables retrieving raster data more quickly. Still, it might also need to store millions of entries that are of a substantial size.

1.4.2 Vector

In a GIS, geographic features are frequently represented as vectors by treating them as geometric forms. Different geometry types can represent different geographical aspects, points, lines, and a polygon.

a. points

Zero-dimensional points are used for geographic characteristics, such as actual location; a single point reference best states that. For instance, well sites, peak altitudes, exciting features, or parking locations. Of these file types, points contain the least amount of information. In small-size displays, points may also be used to indicate areas. For instance, points rather than polygons might be used to depict cities on a global map.

Additionally, some GIS supports the voxel format for data. The term "voxel" is a combination of the terms "volumetric" and "pixel," and it refers to a volume part that represents a rate on a consistent grid in three dimensions. This is comparable to a pixel, which symbolizes 2D image files. Voxels are produced by merging 2D raster slices or 3D point clouds (3D point vector data).

b. Lines or Polylines

Linear features like rivers, roads, railways, trails, and topographic lines are represented by one-dimensional lines, sometimes known as polylines. Linear features exhibited at a small scale will be depicted as linear rather than a polygon, much as point features. The distance can be measured using line characteristics.

c. Polygons

Geographical features covering a certain earth's surface region are represented as two-dimensional polygons. Lakes, park boundaries, buildings, city boundaries, and land usage are a few examples. Polygons contain the most data of all the file kinds with the possibility to calculate area and perimeter.

Each of these geometries has a connection to a database record that lists its properties. For instance, a lake's depth, water quality, and amount of pollution may be listed in a database that defines lakes. Making a map to depict a particular dataset attribute using this data is possible. For instance, the amount of pollution might affect the hue of lakes.

Implementing topological restrictions like "polygons must not overlap," vector features may be forced to preserve spatial coherence. Additionally, continually changing phenomena can be represented using vector data. Elevation or continually changing data are represented via contour lines and triangulated irregular networks (TIN). TINs keep

track of values at point positions joined by lines to create an asymmetric triangular net. The terrain surface is shown on the triangles' faces.

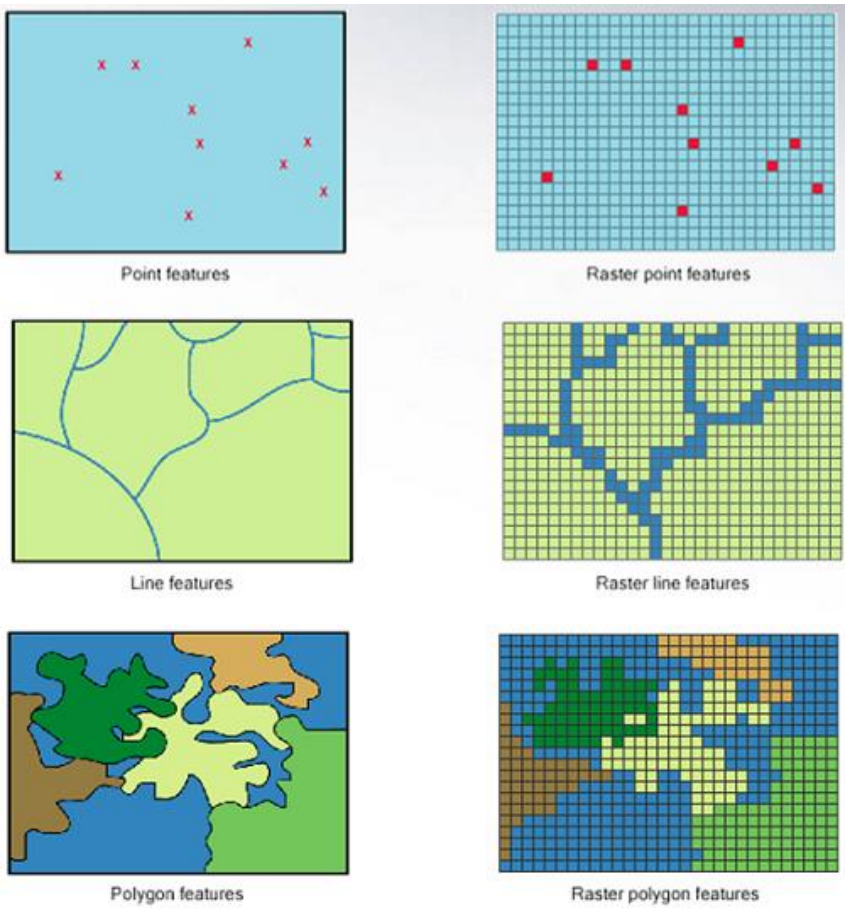


Figure 1.5: Raster vs Vector different type representation[1]

1.4.3 Advantages and disadvantages of Raster and Vector

Data presented in a raster format, which records values for every point in the region covered, data given in a vector form may need additional space for storage, which only stores data where it is required. Additionally, overlay procedures may be carried out quickly with raster data, but they are more challenging with vector data. Raster data will appear as an image that may have a blocky look for object borders depending on the resolution of the raster file, as opposed to vector data, which can be shown as vector graphics used on conventional maps. Vector data may be more straightforward to scale, re-project, and register. This makes it easier to combine vector layers from various sources. Vector data is better suited to relational database settings. They may be used as regular columns in a relational table and processed using a wide range of operators.

[1]Copyright © Crown copyright and database rights 2024 Ordnance Survey EUL 100019606

Vector data often have smaller file sizes than raster data for sharing and storing. Depending on the resolution, image or raster data might be exponentially bigger than vector data. Vector data is also simple to preserve and update, which is another benefit, as in the case of adding a new roadway. It will be necessary to recreate the raster image, but the vector data, or "roads," may be changed by omitting the last portion of the roads.

Additionally, vector data has far greater analytical potential, particularly for "networks" like telecommunications, electricity, and other networks. For instance, the analyst can search for the optimal route or mode of transportation using vector data associated with the features of highways, ports, and airports. Raster data will lack some of the attributes of the features it shows.

Table 1.1: Advantages and disadvantages of Raster and Vector

Mode	Advantages	Disadvantages
Raster	- A good representation of the Ongoing reality. - Simple data structure. - Easy spatial analysis. - Easy combination of layers.	-Uses a lot of space. - low-quality display. - Inaccurate position and shape of objects.
Vector	- Takes up minimal space. - Excellent display quality. - Accurate representation of the position and shape of objects. - Good integration with relational databases. - Object-based approach.	- Inadequate in representing ongoing reality. - complicated data structure. - the overlapping of complex layers.

1.4.4 Non-spatial data

In addition to the geographic data represented by a vector geometry's coordinates or a raster cell's location, additional non-spatial data can also be recorded. Extra information in vector data refers to an object's characteristics. For instance, a forest inventories polygon may include information on the tree variety and a rate for the identifier. In raster files, the cell evaluation can include attribute data and an identity that can be used to connect to records in an alternative table.

1.5 Conclusion

The Geographical Information System (GIS) is the most effective and practical system aimed at decision-making. Using geographical data gathered from diverse sources, GIS will aid in determining the ground reality. In GIS, combining data from several sources with land records, such as remote sensing data and images, is possible. GIS applications would be more suitable for all businesses to determine the extent of operations and monitor activities.

Chapter 2

Remote Sensing for Land Monitoring

Chapitre 2

Remote Sensing for Land Monitoring

2.1 Introduction :

Remote sensing is formally defined as the science and art of obtaining information about an object, area, or phenomenon through analyzing data acquired by a device, not in contact with the object, area, or phenomenon under investigation. Remote sensing refers to a wide range of operations, including using satellite systems, collecting and storing image data, and the subsequent processing, interpretation, and distribution of the resulting data and image products (Karantzalos and Paragios, 2008; Chuvieco and Huete, 2009).

The development of remote sensing as a technique and approach has traditionally progressed in synch with other technological advancements, including the evolution of optics, sensors, satellite platforms, transmission systems, and computer data processing. The first known aerial image was taken in 1858 in France at an altitude of 80 meters using cameras attached to an air balloon. The military achieved the highest aerial surveillance and image interpretation during World Wars I and II. Later, this invention was made available to civilians, which led to its initial use in managing natural resources. NASA launched the Television and Infrared Observation Satellite (TIROS-1) in 1960 to get a fuller knowledge of atmospheric conditions. Afterward, NASA used the term "remote sensing" in the early 1960s to refer to any method of viewing the planet from a distance and introduced digital technology. Today, many satellite sensor systems are being utilized

to view and monitor the globe, collecting vast amounts of data and using various novel techniques to investigate the dynamics of the earth's surface.

Remote sensing now provides vital information about the land, improving our understanding of earth systems and helping to better contribute to their preservation. This information is combined with parallel development in geographic information systems (GIS) and other ground data collection systems. In recent years, in remote sensing, GIS, and spatial modeling, there has been an increase in interest in developing integration tools because of its ability to connect all spatial data sources, GIS is now regarded as the center for managing geographic information (Ghandour and Jezzini, 2018a; Cheng *et al.*, 2020; Zhou and Sha, 2020).

2.2 Satellite and Aerial Imagery

The only fundamental similarity between aerial and satellite images is that both are obtained from heights above the earth. While both can provide a more comprehensive image than conventional images shot on land and can be utilized for tasks like mapping and recognizing geographical features, they also have some fundamental variations that make them different (De Hoog *et al.*, 2020; Iqbal and Ali, 2020; Aksenov and Kozlov, 2021).

Satellite images often cover a considerably bigger region and have larger-scale research uses. Aerial images shot at a lower height covering a smaller area are more appropriate for smaller-scale applications like advertising and marketing. When you consider how each form of image is produced and how it is used, the distinctions become much more apparent.

Figure 2.1: Various platforms used for remote sensing (Xiang, Xia, and Zhang, 2018).

2.2.1 Aerial Images

Aerial photography has a long history, beginning with conventional cameras capturing photos on film. Aerial photography has gotten more digitalized in recent years as technology has advanced. With automated cameras, aerial images have been shot on various platforms, such as balloons, kites, planes, and even pigeons. Since drones provide more flexibility, drones have recently surpassed all other popular methods (Ma *et al.*, 2020).

Many aerial photographs can be captured depending on the camera axis, the craft's height, and the kind of film used. Aerial photography encompasses various techniques, from vertical images that provide a real-time picture of a specific land region to more oblique images that might highlight a specific item or landmark.

Aerial photography is often shot at significantly lower altitudes, ranging from a few hundred meters for drones to as much as ten kilometers for airplanes. The images often have better detail since they are taken at lower elevations. As a result, aerial photography can be more advantageous for various uses, such as real estate transactions, building inspections, and advertising (Suárez *et al.*, 2005).

Aerial photography has several benefits beyond the level of detail the images can capture. Aerial photographs have higher clarity and resolution than other images since they are captured considerably closer to the earth's surface. Aerial photography is also considerably more economical for the typical client since it employs widely available resources like drones and planes, making it the ideal alternative for marketing materials and other private needs.

Figure 2.2: Example of Aerial image[2].

2.2.2 Satellite Images

On the other hand, satellite imaging is a much more recent innovation. Only digital satellite images are captured using a variety of electronic scanners built into satellites that orbit the earth. Satellites typically fly above 100 kilometers, producing images with far broader angles and less detail than aerial photographs. Since satellite images can give a large-scale perspective of an entire weather front, they have been utilized for meteorological monitoring and storm tracking. These satellites, known as Landsats, gather various data using sensors that detect the electromagnetic radiation wavelengths that the earth reflects.

[2] IKONOS Satellite Image of Vatican City – Rome, Italy. Acquired on: May -5, 2003. Copyright © Space Imaging – All rights reserved

While satellite imagery is out of most regular customers' price range, it still provides several advantages over aerial photography. Since satellite images are captured from the earth's orbit, they are unaffected by weather, allowing scientists to watch storms' formation and evaluate their potential threat level even during intense storms. Additionally, since most satellites orbit the earth regularly, obtaining repeated images from them is considerably simpler, providing valuable updates for tracking and mapping reasons. Since satellite images are purely digital, they may be more readily incorporated into current software systems and can capture a far broader angle than aerial imagery.

Figure 2.3: Example of Satellite image[3].

[3] IKONOS Satellite Image of Vatican City – Rome, Italy. Acquired on: May -5, 2003. Copyright © Space Imaging – All rights reserved

2.3 Resolution of Remotely Sensed Data:

Resolution is the aspect of remotely sensed data most important for its utility. For remote sensing imaging, there are four different types of resolution: spatial, spectral, radiometric, and temporal.

2.3.1 Spatial resolution

How much detail is apparent to the human eye in a photographic image is referred to as spatial resolution. One definition of spatial resolution is the capacity to resolve or distinguish small features. Images captured by satellite sensor systems generally have a spatial resolution in meters. For instance, we frequently refer to Landsat as having a 30-meter resolution, which indicates that two side-by-side and thirty-meter-long or broad objects can be distinguished on a Landsat image.

Figure 2.4: Different examples of spatial resolution[4].

2.3.2 Spectral Resolution

The capability of the sensor systems to discern between several Electro-Magnetic Radiation EMR spectrum regions is known as spectral resolution. While some sensors can only detect visible light, others can also detect near-infrared radiation. The bands of an item are the regions of the spectrum to which it is sensitive. Multiple bands with various bandwidths could be present on a sensor. Spectral resolution describes the amount and breadth of bands for a particular sensor.

[4] Spatial Resolution vs Spectral Resolution. By:GISGeography. Last Updated: October 22, 2023 https://gisgeography.com/

A panchromatic band is a broad band which typically covers the visible spectrum and an extensive spectral range. Since we frequently print this kind of image in grayscale, we commonly refer to it, which is delicate to the visual range, as "black and white". The near-infrared region of the spectrum is covered by several analog and digital sensors, which have broad panchromatic bands.

A sensor is a multispectral system when it only captures a small fraction of the spectrum or has a small number of relatively large bands. Red, green, and Blue are the visible colors that makeup two or three of a multispectral sensor's bands. It could also have a few bands in the near-infrared or mid-infrared range. Usually, 4 to 10 bands are present in multispectral systems.

Many relatively tiny bands make up hyperspectral sensors' broad array of frequencies. Compared to multispectral sensors, hyperspectral sensors have a better spectral resolution. A sensor is often hyperspectral if it has a minimum of 20 or 30 bands. These sensors often feature hundreds of bands. Generally speaking, a sensor with additional spectral bands can better distinguish two objects with identical spectral characteristics.

An individual raster layer may represent each band in a digital file. Consider a three-dimensional picture where a cube's x, y, and z coordinates are filled with rows, columns, and bands.

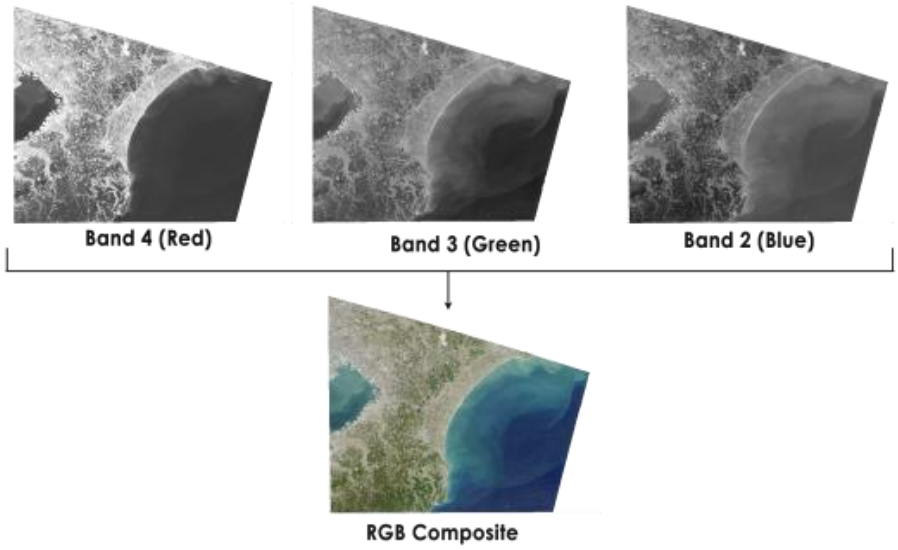


Figure 2.5: Example of Spectral Resolution.

2.3.3 Radiometric Resolution

The number of digital quantization levels utilized to describe the data gathered by the sensor is referred to as radiometric resolution or radiometric sensitivity. The radiometric detail of the data collected by the sensor generally increases with the number of quantization levels. The capacity of a remote sensing system to recognize a target's tiny differences in radiant energy intensity is known as radiometric resolution.

Figure 2.6: Radiometric Resolution 8-Bit and 4-Bit comparition[5]

2.3.4 Temporal resolution

Temporal resolution refers to how frequently a sensor can or will visit a specific location to gather data. This is crucial since many applications rely on the ability to track how phenomena evolve. A satellite, an airplane, a hot air balloon, or another platform is used to support remote sensing equipment.

Some satellites are in Sun-synchronous orbit, which means they never get close to the earth's shadow. Other satellites in geosynchronous orbit retain a constant location above the rotating earth. These satellites have a consistent and predictable temporal resolution in both scenarios. Because specific satellite-based sensors may point at objects close to their default field of view, so they are more adaptable than others. Ad-hoc or on-demand missions are done by sensors installed on aircraft with less predictable but more variable temporal resolution.

---

[5] . What is Bit Depth for Satellite Data (and Images) . By:GISGeography Last Updated: October 29, 2023 . https://gisgeography.com/

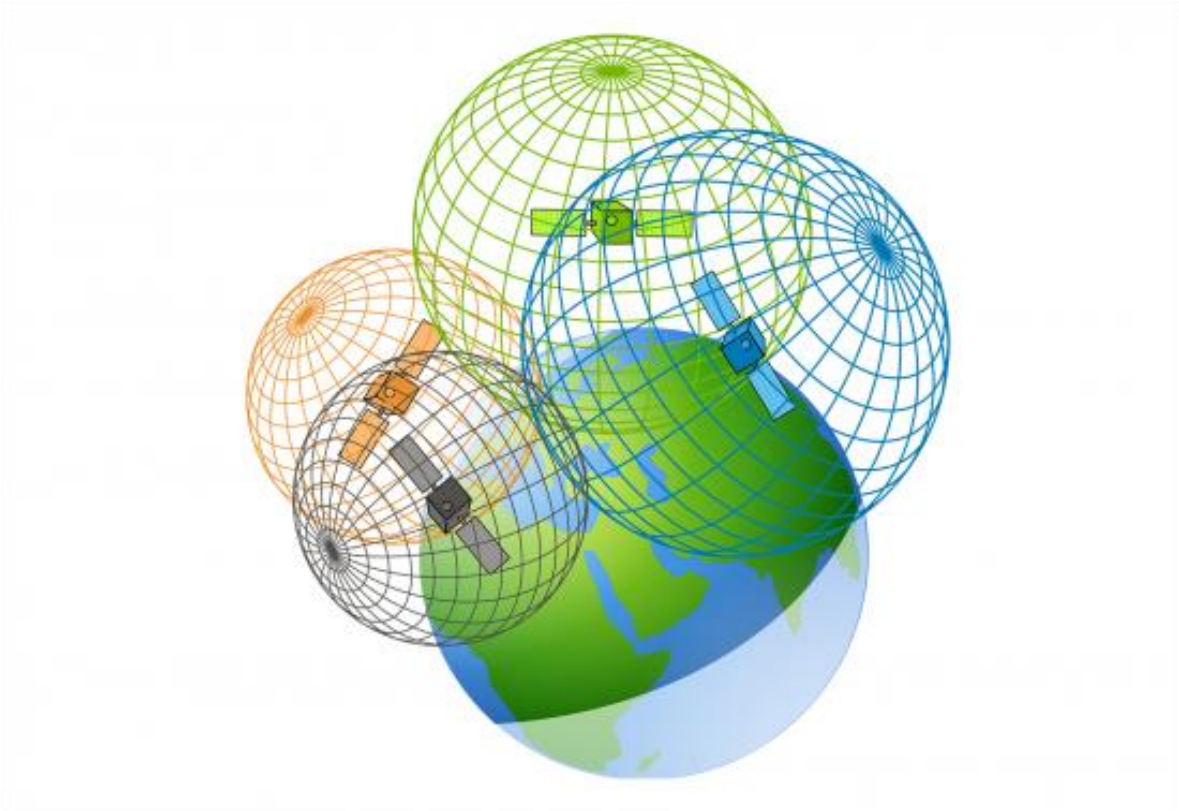


Figure 2.7: Temporal Resolution System[6].

2.4 Types of remote sensing

There are two basic remote sensing types: Active and Passive. Although the difference between the two is minimal, the quality and functionality greatly vary.

2.4.1 Passive remote sensing

Sunlight reflected by the target is the natural energy source used in passive remote sensing. Because of this, it can only be used when there is adequate sunshine since there will not be anything to reflect.

In passive remote sensing, various band combinations are used to measure the obtained quantity through multispectral or hyperspectral sensors. The number of channels used in these combinations varies (two wavelengths and more). The range of bands includes spectra visible to humans and those invisible (visible, IR, NIR, TIR, microwave).

[6] GPS / GLONASS ? c'est qui, c'est quoi ?. https://www.my-trail.fr/. Copyright © 2024 Passion Trail - La course dans le sang

Because of the low strength of the reflected radiation, sophisticated detecting equipment is required. The outputs from the passive sensors are interpreted by radio astronomy equipment. The very low power frequencies these devices can measure are susceptible to the accumulating radiation released from the earth. Due to the availability of Multispectral and Hyperspectral technologies, passive sensors are frequently utilized for satellite photography and in the technical observation of the globe.

The most often used passive remote sensing equipment includes several spectrometers or radiometers.

- A spectrometer distinguishes and analyzes spectral bands.
- Radiometer determines the power of radiation emitted by the object in particular band ranges (visible, IR, microwave).
- A spectroradiometer determines the power of radiation in several band ranges.
- Hyperspectral radiometers operate with the most accurate passive sensor type used in remote sensing. Due to extremely high resolution, it differentiates hundreds of ultimately narrow spectral bands within visible, NIR, and MIR regions.
- An imaging radiometer scans the object or a surface to reproduce the image.
- The sounder senses the atmospheric conditions vertically.
- The accelerometer detects changes in speed per unit of time (e.g., linear or rotational).

2.4.2 Active remote sensing

Active remote sensing, as opposed to passive remote sensing, is independent of the solar radiations reflected from the objects or the ground. Active remote sensors emit their energy focused on the subject or research area. The active remote sensors identify and quantify the radiation that is reflected. Unlike passive sensors, which use microwaves, active remote sensors use radar instead.

Utilizing it throughout the day or night, over terrains, and under weather circumstances where catching solar radiation may be difficult is facilitated by the active sensors' ability to supply their energy source for emission. Active sensors' emissions are reflected by the earth's surface and atmosphere, respectively. That enables many space-based active sensor satellites to participate in various tasks, such as constructing a three-dimensional body of the cloud, which is particularly useful in forecasting weather.

Each active sensor sends its signal in the direction of the object and then measures the response to determine the quantity received. What active remote sensing methods send (light or waves) and what they ascertain are different.

- A radar is a sensor that uses radio frequencies to help in range. Its antenna that emits impulses is a distinctive characteristic. In radar active remote sensing, energy flow encounters an obstruction and, to some extent, scatters back to the sensor. It is possible to calculate how far away the objective is based on the quantity and distance traveled.
- Lidar uses light to measure distance. Remote sensing with a laser involves sending light pulses and measuring the amount received. Calculating the target location and distance involves dividing the time by the speed of light.
- A laser altimeter measures elevation with lidar.
- A scatterometer is a specific device to measure bounced (backscattered) radiation.

2.5 General Remote Sensing Applications

The spectrum, spatial resolution, and temporal resolution prerequisites for satellite sensors vary depending on the application. There are several possible uses for remote sensing across various sectors. Some of them are described below.

2.5.1 Land Cover and Land Use

The terms "land cover" and "land uses," despite being frequently confused, have separate meanings. While land use implies the function of the land, land cover refers to covering the ground's surface. The land cover characteristics detected by remote sensing techniques can infer land use, especially when combined with auxiliary information or prior knowledge.

Examples of land use applications for remote sensing include the management of natural resources, the preservation of wildlife habitats, starting point mapping for GIS input, urban development, logistics management for seismic/exploration/resource extraction activities, damage establishing (tornadoes, flooding, volcanic, seismic, and fire), finding targets, and the recognition of landing strips, roads, clearings, bridges, and land/water interface.

2.5.2 Agriculture

Agriculture is a significant component of the economy in both affluent and underdeveloped nations. Mapping methods such as satellite and aerial photographs are utilized to categorize crops, assess their health and viability, and monitor agricultural operations. Remote sensing is used in agriculture for various tasks, such as identifying crop types, evaluating crop health, estimating crop output, mapping soil traits, mapping soil management activities, and tracking compliance.

2.5.3 Geology

Geology examines landforms, buildings, and the subsurface to comprehend the physical processes that shape and alter the earth's crust. It is most frequently defined as the discovery and utilization of mineral and hydrocarbon resources to raise societal standards of life. Remote sensing applications in geology include bedrock mapping, lithological mapping, structural mapping, mining of sand and gravel, mineral exploration, hydrocarbon exploration, environmental geology, geo-botany, baseline infrastructure, sedimentation monitoring, event/monitoring, geohazard mapping, and planetary mapping.

2.5.4 Hydrology

The study of water on the earth's surface includes water flowing above ground, frozen in ice or snow, or held by the soil. Wetland monitoring, soil moisture estimation, snowpack monitoring, snow thickness measurement, snow-water equivalent calculation, ice monitoring, flood monitoring, glacier dynamics monitoring (surges, ablation), river/delta change detection, drainage basin mapping, watershed modeling, irrigation canal leakage detection, and irrigation scheduling are some examples of hydrological applications.

2.6 Conclusion

The application of remote sensing technology has overgrown the advancement of various industries. In contrast to traditional survey methods, satellite imagery uniquely offers genuine synoptic views of a broad region at once. Remote sensing offers a plethora of data on a worldwide scale for concerns including climate change, natural resources, disaster management, and the environment. Additionally, the process of data collecting and analysis using a geographic information system is also relatively quick.

Chapter 3 I

Features

Determination

and Image

Processing

Chapitre 3

Features Determination and Image Processing

3.1 Introduction

Image processing is applying different procedures to an image to improve or extract useful information. It is a kind of signal processing where the input is an image, and the output may be another image or features or characteristics related to that image. Image processing is one of the technologies that is currently expanding quickly. It is a primary research subject in the engineering and computer science fields.

Image processing techniques may be categorized into two types: analog and digital. The hard copies, such as prints and pictures, may be processed using analog image technology. When interpreting images utilizing these visual methods, image analysts employ a variety of interpretive basics. Through the use of computers, digital image processing techniques enable image alteration. Image processing mainly deals with image acquisition, enhancement, Segmentation, feature extraction, image classification, etc.

In this chapter, we will examine the theoretical principles of several image-processing techniques that we will employ later in our research to propose a novel building detection approach.

3.2 Color SpaceModels

A color model is a structure that helps define and describe colors using numerical values. Many color models utilize various mathematical structures to describe colors; however, most color models combine three or four values or color components. The three most popular color models are RGB (used in computer graphics). YIQ or YUV, and CMYK (used in color printing). Nevertheless, none of these color spaces are directly associated with hue, saturation, or brightness concepts. That led to exploring other models, such as HSL and HSV, to simplify programming, processing, and end-user manipulation (Lei, Techentin, and Gilbert, 1999; F., H. and D., 2003).

3.2.1 RGB Color SpaceModel

The RGB model is used in digital screen-based designs, such as those on a computer or phone display. The three primary colors, Red, Green, and Blue, are given a value in the RGB color model between 0 and 255, where 0 is dark and 255 is brilliant. You can define the precise hue blended by giving the three values for the red, green, and blue phosphors (Carson *et al.*, 1999).

As an additive color system, the RGB color model causes colors to become lighter when combined. Each element of light adds to the mixture to create a new shade. You may produce any color within the device's limitations using various additive primary combinations—white results from the balanced combination of all three colors.

There are some colors that a computer display cannot adequately represent since the RGB color model can only create a limited gamut of colors. The limits of the video hardware in the computer, which may only show black and white up to 16.7 million colors, further reduce the number of colors that can be seen on a monitor.

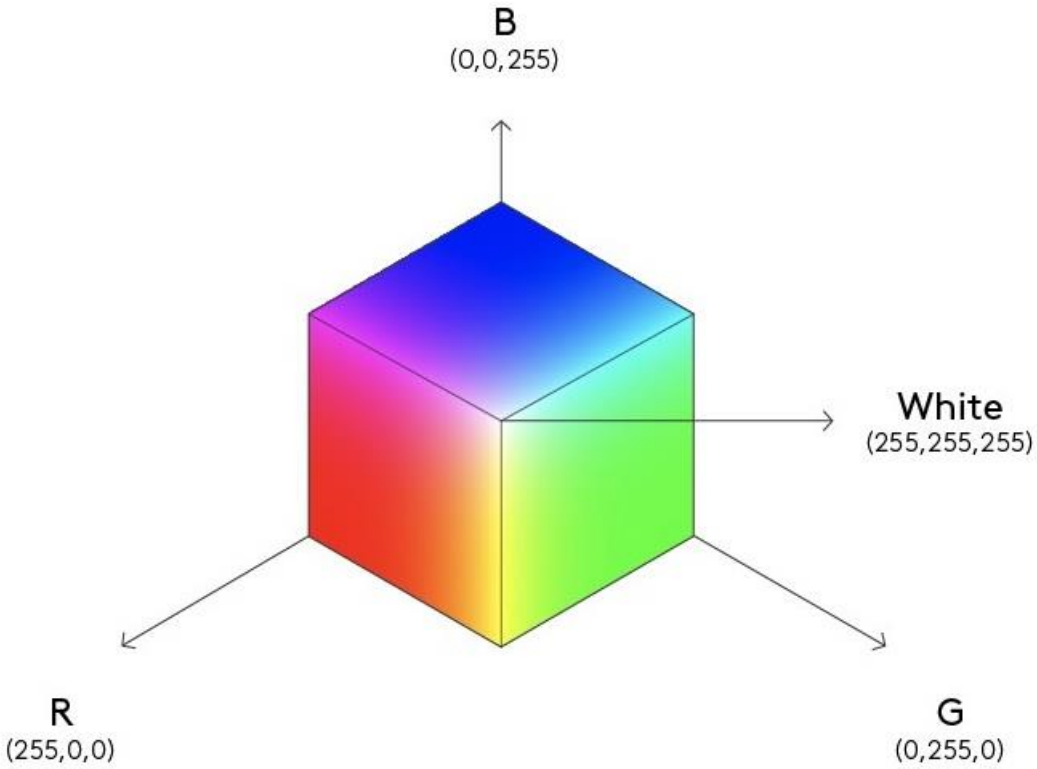


Figure 3.1: RGB Color SpaceModel[7].

3.2.2 YIQ Color SpaceModel

The I and Q components of the YIQ color space represent the color information, whereas the Y component represents the intensity. TV broadcasting finds the YIQ color space to be highly appealing since it helps to retain compliance with monochrome TV standards by isolating the intensity from the color information. The human eye is more sensitive to changes in luminance than changes in hue or saturation, which is another advantage of the YIQ model. The components Y, I, and Q are assumed in the [0, 1] or [0, 255] range (Acharya and Ray, 2005; Camastra, 2007).

The RGB to YIQ conversion is defined as:

$$[Y] = [0.299 \quad 0.587 \quad 0.114]\,[R]$$

$$[I] = [0.596 \quad -0.275 \quad -0.321]\,[G]$$

$$[Q] = [0.212 \quad -0.523 \quad -0.311]\,[B]$$

[7] RGB color space RGB color model Light, light, angle, rectangle, color png. https://www.pngwing.com/en/.

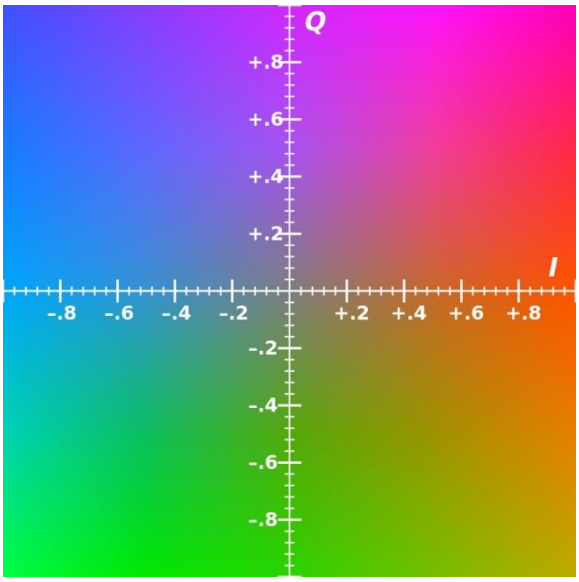


Figure 3.2: The YIQ Color Space Model at Y=0.5.(Guru Prathap Reddy *et al.*, 2024)

3.2.3 CMYK Color SpaceModel

The CMYK color model describes colors according to how much they contain: Cyan, Magenta, Yellow, and Black. The CMYK model is commonly used by computer printers and conventional "four-color" printing machines. In the CMYK paradigm, you may make almost any color you want by combining cyan, magenta, yellow, and black inks or paints (Pham *et al.*, 2007; Bertacchi and Silveira, 2019).

Theoretically, cyan, magenta, and yellow alone may be combined to create any reflecting hue. However, the inks that printers use daily are not flawless. The best example of this is when you combine all three to produce black. The resulting hue is muddy brown because the primaries overlap and don't completely eliminate light when combined.

As a subtractive color model, CMYK causes colors to blend into a darker shade. The blended paints or inks absorb different parts of the light differently. If the appropriate mixture of pigments is used, all of the light elements are absorbed, producing an almost dark color.

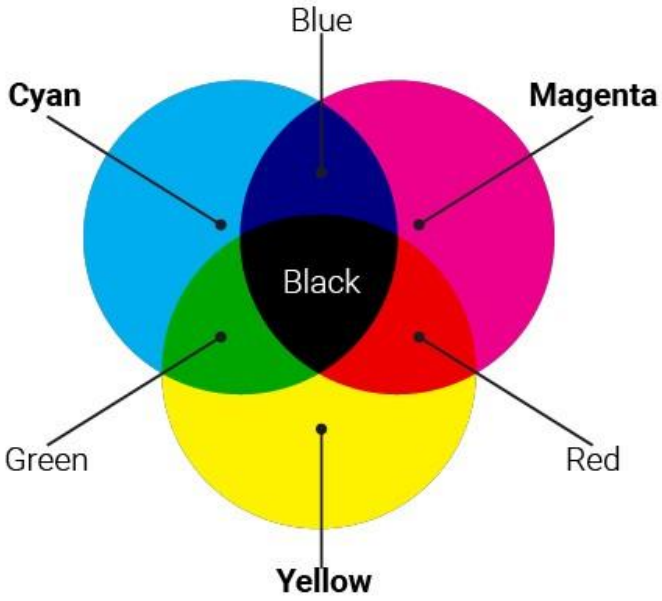


Figure 3.3: CMYK Color SpaceModel[8].

3.2.4 HSL Color SpaceModel

Despite the majority of human eyes seeing colors similarly to how the RGB model does, we often do not consider or discuss colors as a combination of these three elements. However, we may discuss how specific colors are brighter than others, have various hues or tones, or are more or less saturated than others. Due to this, many software programs have color pickers that aim to accommodate human perception of colors.

The abbreviation HSL, which stands for hue, saturation, and lightness, is one such perceptual color model. It was first introduced as "hue/chroma/intensity" by Joblove and Greenberg (Joblove and Greenberg, 1978). Figure 3.4 shows that the vertical axis represents all shades of gray between 0 (black) and 1 (white), so they define the color space as a bizonal solid. Then, at L=0.5, all fully saturated colors sit on the periphery of the common basis of both cones, allowing for the definition of hue as an angle. The circle's radius surrounding the vertical axis at the location of the present brightness is the third parameter, known as saturation.

[8] CMYK . https://techterms.com/. Updated April 20, 2023 by Brian P.

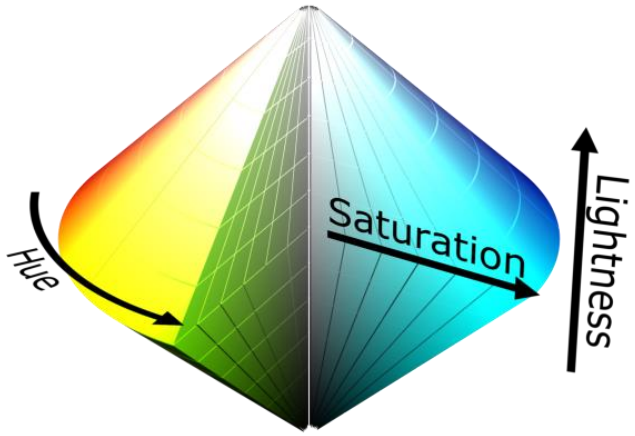


Figure 3.4: HSL Color SpaceModel (Labrecque, 2020).

3.3 Image Segmentation

Image segmentation is a technique that splits a digital image into several subgroups known as Image segments, simplifying future processing or analysis of the image by decreasing the complexity of the original image. In simple words, Segmentation is the process of giving pixels labels. Each pixel or piece of a picture allocated to the same category has a unique label (Doggaz and Ferjani, 2011; Jaiswal and Pandey, 2021; Minaee *et al.*, 2021).

There are three main goals for Segmentation. The first aim is to split the image into sections for further analysis. In simple cases, the environment may be sufficiently managed so that the segmentation procedure consistently recovers the elements that necessitate additional analysis—for instance, a segmentation technique for a human face from a color video clip. The Segmentation is trustworthy if the person's attire or the surroundings don't share the same color components as a human face. The segmentation problem may be particularly challenging in complicated situations, such as when attempting to extract an entire road network from a greyscale aerial image, and may call for extensive domain-specific expertise. Segmentation's second goal is to execute a representational adjustment. Grouping the image's pixels into higher-level units that are either more useful or efficient for subsequent analysis is necessary.

3.3.1 Threshold-based Segmentation

Thresholding splits pixels based on how intense they are compared to a specified value or threshold, making it the most straightforward method for segmenting images. It helps distinguish between objects with higher intensity than backgrounds or other elements (Sivaraj, Malmathanraj, and Palanisamy, 2020; Khan *et al.*, 2022).

$$g(v) = \begin{cases} 0 & v < t \\ 1 & v \geq t \end{cases} \quad (3.1)$$

Wher v represents a grey value, and t is the threshold value.

Using thresholding, a grayscale image may be converted to a binary image. Following the thresholding process, the image has been divided into two segments, each indicated by the pixel values 0 and 1.

Several techniques exist to choose an acceptable threshold value for a segmentation task. Probably the most typical approach is to set the threshold value interactively, with the user adjusting the value and checking the segmentation results until a satisfactory segmentation is achieved. A helpful tool for choosing an appropriate threshold value is generally the histogram.

Threshold segmentation can be expanded to use multiple thresholds to divide an image into more than two segments when several desired segments can be distinguished by their different grey values. For example, all pixels with values below the first threshold are assigned to segment 0, all pixels with values between the first and second threshold are assigned to segment 1, and all pixels with values between the second and third threshold are assigned to segment 2.

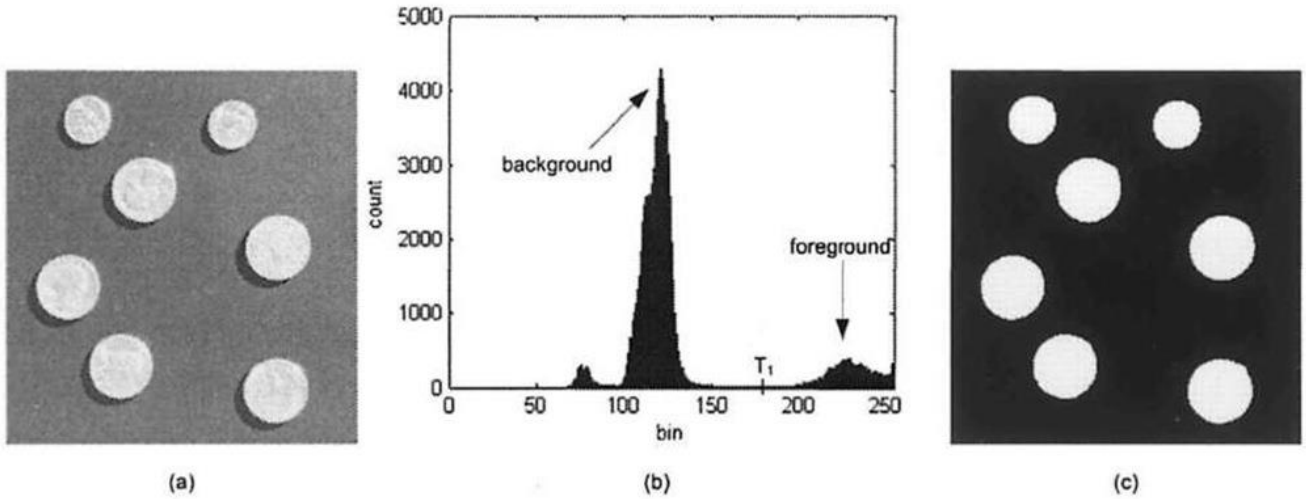


Figure 3.5: Threshold-based Segmentation of an image, (a) Unprocessed coins image, (b) Bimodal histogram of the image, (c) Result of thresholding the image using a value of T=180.(Khan et al., 2022)

3.3.2 Edge-based segmentation

An edge-based segmentation is a standard method for processing images that recognizes the edges of different objects in an image—using the information from the edges assists in locating characteristics of related items in the image. Edge detection helps reduce the size of images and makes analysis easier by removing unnecessary data. Using contrast, texture, color, and saturation variations, edge-based segmentation algorithms detect the edges. Edge chains of various edges allow them to depict objects' borders precisely in an image (Kang, Yang, and Liang, 2009; Maini and Aggarwal, 2009).

Figure 3.6: Edge-based Segmentation using the canny method (Maini and Aggarwal, 2009)

3.3.3 Region-Based Segmentation

An image is divided into regions with connected attributes in region-based segmentation. The method locates a group of pixels for each region using a seed point. After locating the seed points, the technique can enlarge regions by including more pixels or by reducing them and combining them with other points (Tremeau and Borel, 1997; Jun 2010).

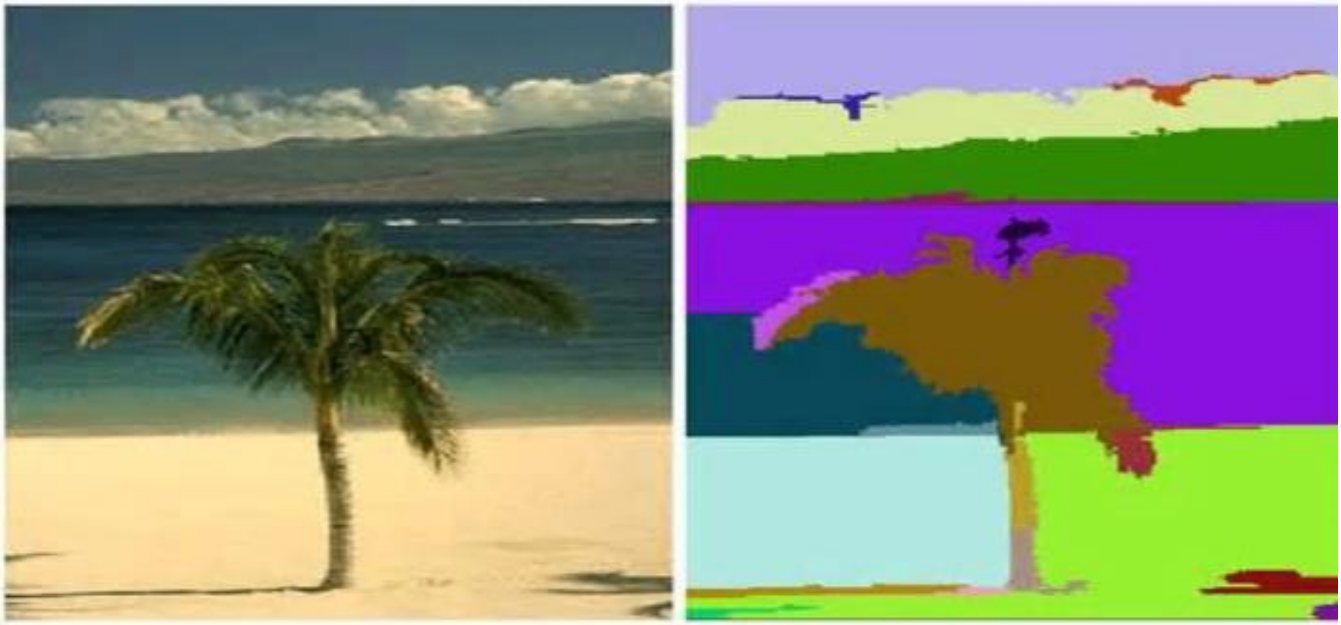


Figure 3.7: Region-Based Segmentation. (Tremeau and Borel, 1997; Jun, 2010)

3.3.4 Cluster-Based Segmentation

Clustering algorithms are unsupervised classification techniques that aid in finding invisible in-image data. They improve human vision by highlighting clusters, shades, and patterns. By dividing data elements and clustering identical elements into groups, the algorithm separates images into groups of pixels with similar properties (Likas, Vlassis and Verbeek, 2011; Dhanachandra, Manglem and Chanu, 2015).

Clustering techniques are often formulated for data of arbitrary dimensions. Still, many clustering methods can readily be applied to two- or three-dimensional images.

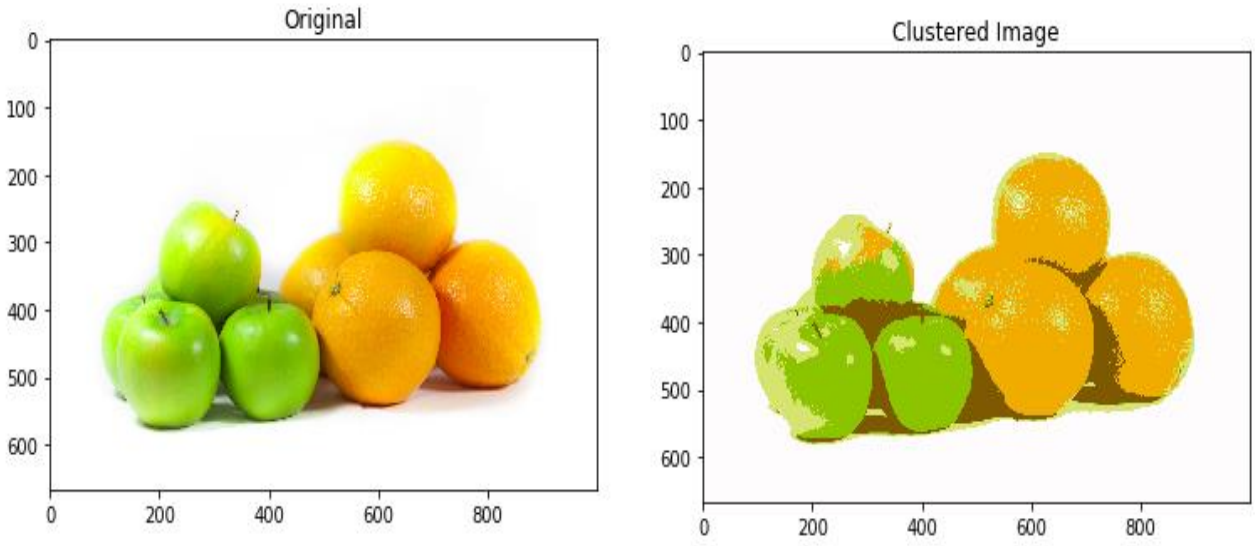


Figure 3.8: Example result of using cluster algorithm (Manglem and Chanu, 2015).

3.3.5 Superpixels Segmentation

In order to divide an image into areas, superpixel approaches take into account similarity measures that are established using perceptual features. Through the use of superpixels, similar-looking clusters of pixels are generated. The goal is to produce areas that convey meaningful descriptions using much less data than when an image's whole pixel count is used. The idea is that by lowering the number of primitives, redundancy decreases, and the complexity of recognition tasks decreases. Superpixels also eliminate the rigid pixel structure by defining areas that preserve meaning in the image. As a result, the regions convey information about the scene's structure, making other processing tasks easier than utilizing individual image pixels. Superpixel methods often rely on measurements that check for color resemblances and measurements of the geometry of the areas. In order to define areas, the segmentation procedure also uses edges or significant variations in intensity (Bobbia *et al.*, 2021; Hassan *et al.*, 2021; Palanikumar, Albert Jerome and Jayan, 2022).

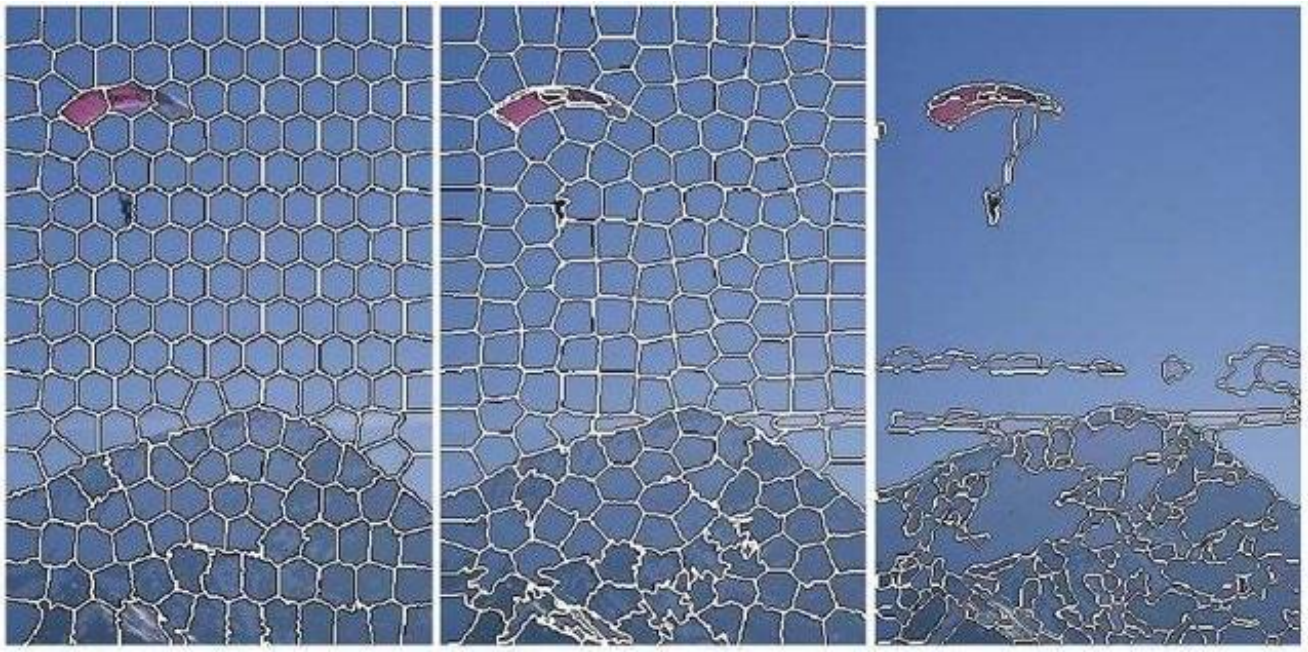


Figure 3.9: Examples of superpixel segmentation using a different approach (Bobbia et al., 2021).

3.4 Morphological Operations

Morphology is a broad range of image processing techniques that manipulate images according to their forms. An input image is given a structural element by morphological procedures, which results in an output image of the same size. In a morphological operation, the value of each pixel in the output image is determined by comparing it to its neighbors in the corresponding pixel in the input image (Li and Shen, 2006; Wang, 2020).

Morphological operations affect an object's form, structure, or shape. They are applied to binary images (black and white) for pre- or post-processing (filtering, thinning, and pruning) or to represent or describe the form of objects or areas (boundaries, skeletons, convex hulls).

Image segmentation and morphology have a small amount of overlap. Morphology consists of techniques that may be used as pre-processing on the input data or as a post-processing approach to the results of the image segmentation stage. In other words, when segmentation is complete, morphological techniques may be used to fix errors in the segmented image and offer information about the layout and form of the image.

3.4.1 Dilation Operation

Dilation is a process in which the binary image is expanded from its original shape. The structural element determines how the binary image is enlarged. The structural element is smaller than the image and is typically 3 x 3 in size. When a structuring element is moved from left to right or from top to bottom as part of the dilatation process, the function looks for any overlapping similar pixels between the structuring element and the binary image. If there is an overlap, the pixels under the structural element's center position will be set to 1 or black.

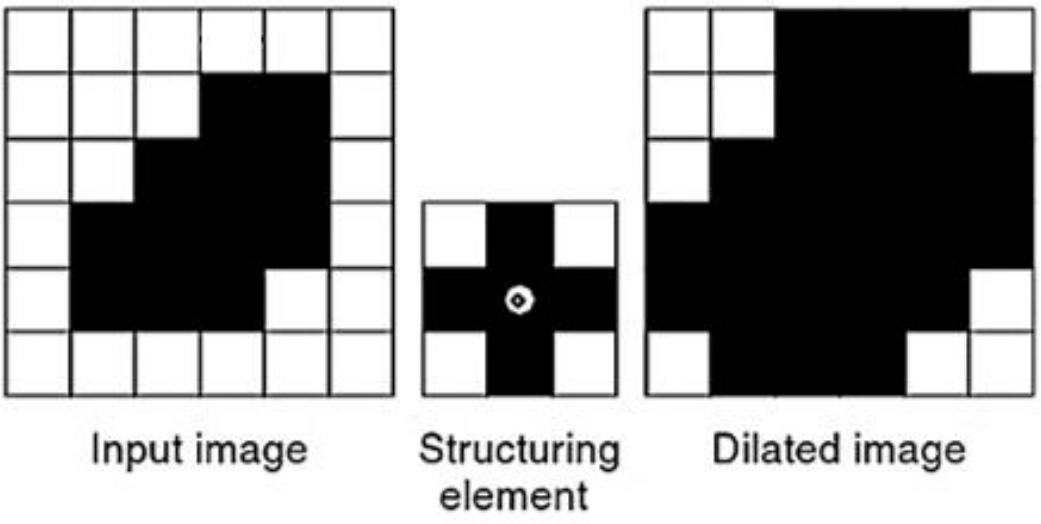


Figure 3.10: Dilation Operation (Wang, 2020).

3.4.2 Erosion Operation

The opposite of dilation is erosion. Erosion shrinks an image to the extent that dilation expands it. Choosing a structural element reduces the image's size, similar to how it is dilated. If there is no total overlapping at that point, the center pixel specified by the structural element's center will be set to white or 0.

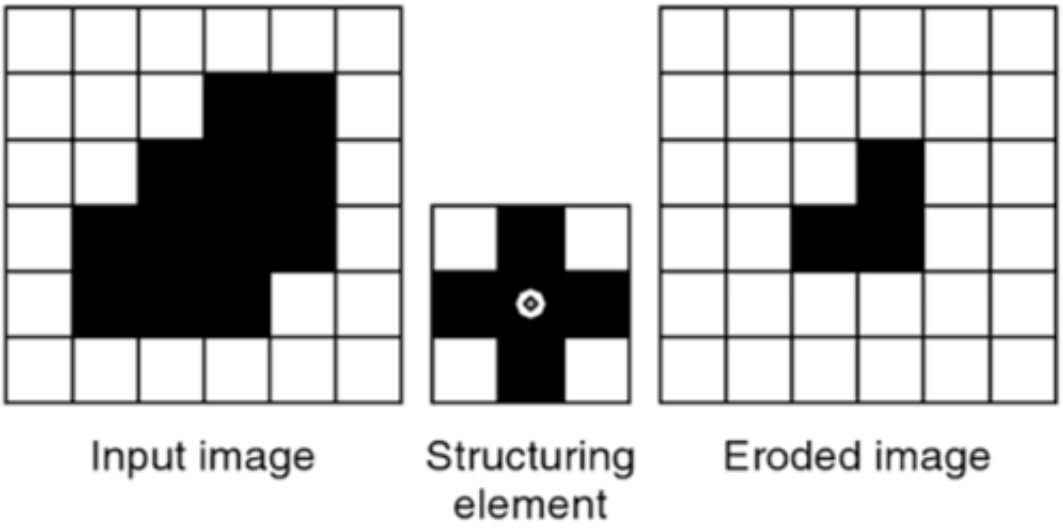


Figure 3.11: Erosion Operation (Wang, 2020).

3.4.3 Opening and Closing Operation

Opening and closing are two consecutive procedures used to repair damaged images in image processing. The opening is typically employed to restore or recover the original image to the maximum extent possible. On the other hand, The purpose of the closing is often to restore the narrow gaps and long, thin coastlines and to smooth out the contour of the deformed image. Closing is further employed to eliminate the tiny holes in the acquired image. The combination of Opening and Closing is generally used to clean up defects in the segmented image before using it in digital analysis.

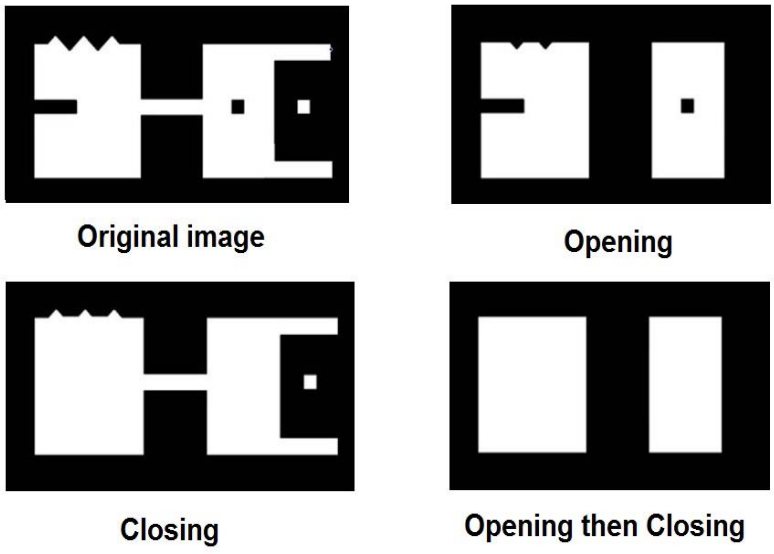


Figure 3.12: Opening and Closing Operation(Wang, 2020).

3.5 Edge Detection

Edge detection is the process of finding sharp contrasts in intensities in an image. While maintaining most of the image's structural details, this method significantly decreases the data in the image. An edge detection technique pulls the essential elements from an image (Maini and Aggarwal, 2009).

There are several approaches to edge detection. However, the majority of methods may be classified into two categories: Gradient−Based Edge Detection and Laplacian–Based Edge Detection. The gradient-based method detects the edges by searching for the maximum and minimum in the first derivative of the image.on the other hand, The Laplacian method searches for zero crossings in the second derivative of the image to find edges.

The gradient-based Display displays a maximum situated at the image's edge center. This method of locating an edge is characteristic of the gradient filter, a family of edge detection filters. A pixel position is classified as an edge location if the gradient's value exceeds a certain threshold. As a result, after a threshold has been established, it is possible to compare the gradient value to the threshold value and identify an edge anytime the threshold is surpassed. The second derivative is 0 when the first derivative is at its maximum or minimum. Finding the second derivative's zeros is an alternative method of determining the position of an edge known as the Laplacian –Based.

3.5.1 Sobel Operator

The operator computes approximate derivatives using two 3*3 kernels, one for horizontal changes and the other for vertical changes, which are convolved with the original image. The computations are as follows if we consider A to be the source image, Gx, and Gy are two images where at each location include, respectively, the approximate horizontal and vertical derivatives; the computations are as follows:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \qquad (3.3)$$

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \tag{3.4}$$

These kernels, one for each of the two perpendicular orientations, are designed to interact with the pixel grid's vertical and horizontal edges as efficiently as feasible. The kernels can be applied individually to the input image for separate measurements of the gradient component in each direction (Gx and Gy). Merging them will provide the absolute gradient magnitude and direction at each point. The gradient's magnitude is as follows:

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2} \tag{3.5}$$

3.5.2 Roberts Operator

The Roberts Cross operator applies a basic, 2-D spatial gradient measurement to an image. The estimated absolute magnitude of the spatial gradient of the input image at each position in the output is represented by the pixel values at those locations. Eq (3.6) shows that the operator consists of two 2*2 convolution kernels. This is highly similar to the Sobel operator.

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \tag{3.6}$$

3.5.3 Canny Operator

Canny edge detection is considered to be the optimal edge detection operator. According to Canny, the best criteria for finding edges are those that increase the probability of getting actual edges while decreasing the possibility of finding false ones. He discovered that a reliable measurement of actual edges was the zero-crossings of the second directional derivative of a smoothed image. The Canny edge detector employs Gaussian convolution to smooth the image; the spread of the Gaussian determines how much smoothing is applied.

$$\boldsymbol{g(m,n)} = G_\sigma(m,n) * f(m,n) \tag{3.7}$$

$$G_{\sigma} = \frac{1}{\sqrt{2.\pi.\sigma^2}} exp[-\frac{m^2+n^2}{2.\sigma^2}] \quad (3.8)$$

After that, a 2D first derivative operator is used to convolve the image to identify areas with sudden changes in intensity. Keep in mind that the first derivative gradient's maximum and minimum correspond with the zero-crossings of the second directional derivative. Figure 3.133 represents the two kernels that the canny operator employed.

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

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

Figure 3.13:Canny operator Kernels(Maini and Aggarwal, 2009).

Because these pixels reflect the image regions with the sharpest intensity fluctuations, only the maxima crossings are of interest. The ridge pixels that indicate the collection of potential edges are these zero-crossings. All other pixels are ignored since they are considered non-ridges.

The final set of edges is then determined along the ridge pixels using a two-threshold approach known as hysteresis. The Canny method employs a linked components analysis technique based on a hysteresis thresholding heuristic instead of a single threshold value for filtering ridge pixels.

This step divides the ridge pixels into edges and non-edges using two thresholds, T1 and T2, where T1 > T2. Confirmed edges are pixels with gradient magnitudes greater than T1. Potential edges are those pixels between T1 and T2. Non-edge pixels fall under the T2 classification. All prospective edges connected to a definite edge through neighboring potential edges are then designated as definite edges. By recognizing firm edges while compensating for relatively weaker ones, the technique addresses some of the problems with edge streaking and discontinuity in the output produced by basic detectors.

3.5.4 Laplacian of Gaussian

Laplacian filters are derivative filters used to identify edges in images where there is a quick change. It is used to smooth the image (for example, using a Gaussian filter) before applying the Laplacian since derivative filters are particularly sensitive to noise. This two-step process is called the Laplacian of Gaussian (LoG) operation. Generally, the operator takes one grayscale image as input and outputs another grayscale image.

The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by:

$$\boldsymbol{L(x,y)} = \left(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}\right) \tag{3.9}$$

Because of the discrete pixel representation of the input image, we must identify a discrete convolution kernel that can approximate the second derivatives used to define the Laplacian. Figure 3.14 displays two typical small kernels.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Figure 3.14: two typical small kernels (Maini and Aggarwal, 2009).

This approach uses much less math since the Gaussian and the Laplacian kernels are often considerably smaller than the image. Since the LoG (Laplacian of Gaussian) kernel can be determined beforehand, a single convolution must be applied to the image at run-time.

The 2-D LoG function centered on zero and with Gaussian standard deviation has the form:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.9}$$

3.6 Texture features

Unlike color, which is often merely a pixel attribute, texture, as a well-studied image aspect, can only be determined from a set of pixels. Due to its outstanding discriminatory powers, the texture feature is frequently used in image retrieval and semantic learning techniques. Several stages of the process might benefit from texture analysis. Images might be split into adjacent sections during pre-processing depending on the textural characteristics of each zone. Texture characteristics might offer hints for categorizing patterns or recognizing objects during the feature extraction and classification stages (Malik and Perona, 1990; Frstner, 1994; Islam, Zhang and Lu, 2008; Nayak, Padhy and Mishra, 2017).

Based on the derived feature, the texture may be divided into two categories: spectral texture feature extraction methods and spatial feature extraction methods.

3.6.1 Spatial texture feature extraction

In the spatial approach, texture characteristics are recovered by calculating pixel statistics or identifying local pixel structures in the source image domain. There are three different methods for extracting spatial texture features:

Structural: In structural approaches, texture primitives (exons or texture components) and their placement requirements are utilized to characterize textures. Syntactical pattern recognition methods are used to compare the resemblance of the two descriptors (C. and E., 2002).

Statistical: Using the statistical texture features, texture assesses the low-level stats of grey-level images. Events, Tamura texture features, and features generated from the grey-level co-occurrence matrix (GLCM) are standard spatial domain statistical characteristics. Statistical characteristics are compact and reliable since they are created from extensive data. However, they are insufficient to represent the diverse spectrum of textures (F., H. and D., 2003; He *et al.*, 2006).

Model-based: Model-based techniques use generative or stochastic models for interpreting texture. Model parameters define the image's fundamental textural characteristics. Popular texturing models include fractal dimension (FD), simultaneous autoregressive (SAR), Markov random field (MRF), and others. Due to their optimization

requirements, these models are generally computationally intensive (Tuceryan and Jain, 1993; Lions *et al.*, 1995; Liu and Picard, 1996; Wang, Li and Wiederhold, 2001; Jiebo Luo and Savakis, 2002).

3.6.2 Spectral texture feature extraction

An image is transformed into the frequency domain, and then a feature is calculated using spectral texture feature extraction techniques. Standard spectrum approaches are Fourier transform (FT), discrete cosine transform (DCT), Wavelet, and Gabor filters.

Features that are spatial or spectral both have benefits and drawbacks. Any shape's spatial properties may be retrieved without losing information, and they often have semantic meaning that people can grasp. Nevertheless, collecting many spatial characteristics for image or area representation might be challenging because spatial features are frequently noise-sensitive. The robustness of spectral texture features contrasts that. Additionally, they need less computing time since FFT is used to perform convolution in the spatial domain as a product in the frequency range.

Moreover, They lack the semantic significance associated with spatial aspects. Spectral texture characteristics are a helpful option for sufficiently large images or areas. Nevertheless, spatial features should be considered for small images or wildly irregular regions.

3.7 Conclusion

Image processing and analysis is the process of looking at images to identify objects and determine their implications. Image analysts examine the information gathered by remote sensing and attempt to detect, categorize, measure, and analyze the significance of natural and manufactured objects and their patterns and spatial relationships. In this chapter, we have covered several methods for processing and analyzing remotely sensed images that have been utilized in the research for feature extraction. The next chapter will discuss exploiting these extracted features to separate different elements presented in the images.

Chapter 4

Machine Learning for Features Extraction and Classification

Chapitre 4

Machine Learning for Features Extraction and Classification

4.1 Introduction

Machine learning (ML) is a subfield of artificial intelligence that uses data to empower computers to learn. ML employs algorithms to learn from given data iteratively, which is the most powerful data analysis technique (Adnan *et al.*, 2019). Iterative follow-up is performed using models that are set to take new data. These models could be able to make essential projections and decisions. According to (Carbonell, 1981; Dietterich and Oregon, 1996; Dietterich, 2002), ML aims to develop computer systems that can learn from and adapt to their surroundings. The main objective of machine learning is to increase the system's efficacy and efficiency. Machine learning approaches can be classified into supervised, unsupervised, and deep learning.

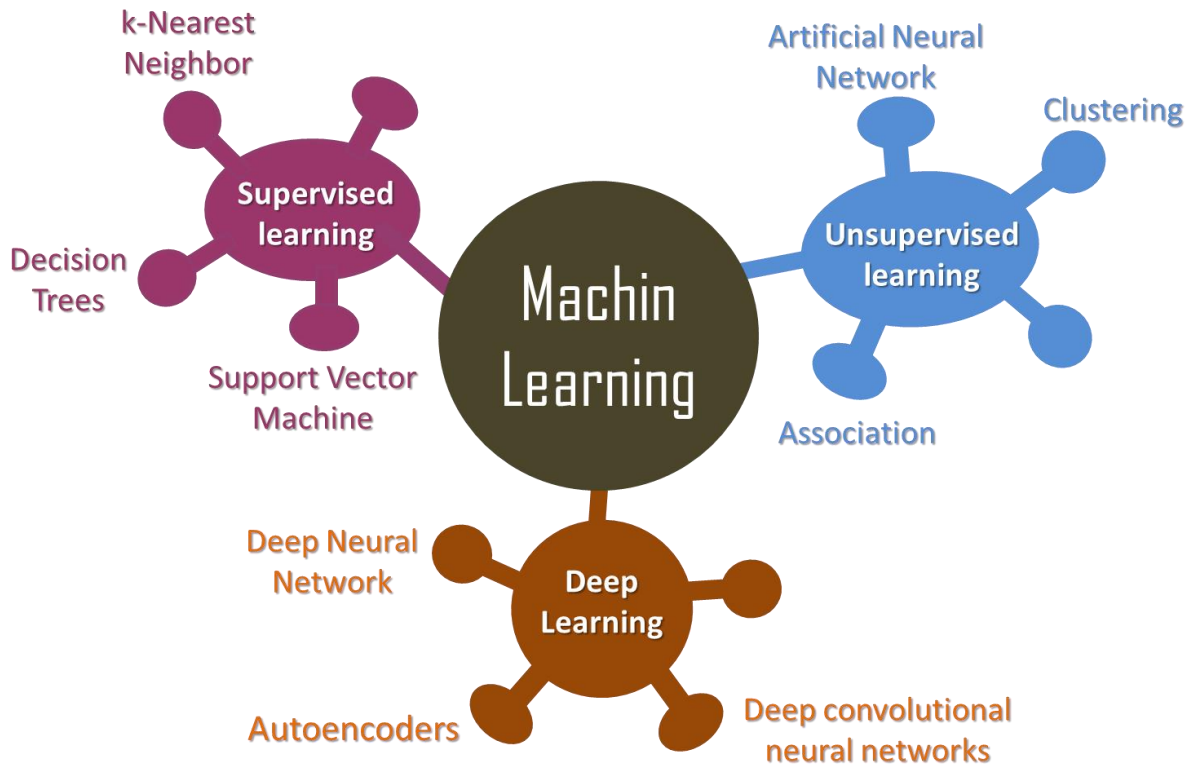


Figure 4.1: diagrammatic representation of ML techniques.

4.2 Supervised-learning

Supervised learning is the research and development of algorithms that generate broad hypotheses—predictions about future instances—from examples provided by external sources. The SL aims to develop an explicit model of the class label distribution in terms of predictive attributes or express it differently. When the class label value is unknown, but the predictor feature values are known, the resulting classifier is then used to assign class labels to the testing instances.

4.2.1 The k-Nearest Neighbor (k-NN)

KNN is a non-parametric supervised approach based on feature space similarity. In its most basic version, a test sample's label is determined by a majority vote of its K-nearest neighbors from the training set. The test data is given the class of the lone closest neighbor if K = 1. In Figure 4.2, the assignment for K=1 and K=4 is given.

The value of the hyperparameter K and the chosen distance measure affects how well KNN performs. The test sample will have a tiny region if K is minimal, which may lead to subpar performance due to sparse, noisy, unclear, or poorly labeled data. Outliers from

other classes arise if we try to raise the value of K. A variety of weighting strategies are also used by advanced KNN algorithms to weight the contributions of the neighbors, allowing the closer neighbors to contribute more to the outcome of the majority vote than the further neighbors.

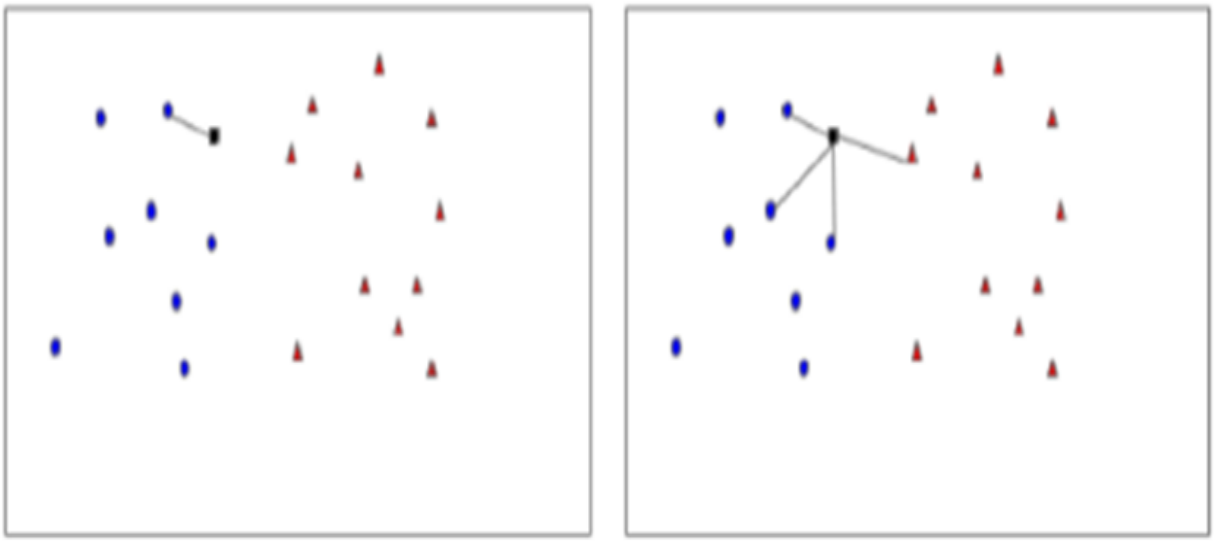


Figure 4.2 : An example of K-nearest neighbour assignment with K = 1 (left) and K = 4 (right) (best viewed in colour).

4.2.2 Support Vector Machine

A separating hyperplane with a maximum margin defines a Support Vector Machine (SVM) discriminative classifier. To put it another way, the algorithm produces a hyperplane that best separates the data relative to their labels (positive/negative) with the most significant margin for labeled training data. Assume that there are n instances in the training set with x_i = Rd and y_i = 1, 1 indicating whether or not it belongs to the class. In SVM, the objective is to discover a hyperplane with parameters w and b in which the following conditions are met for all data points:

$$w \cdot x_i + b >= 1, if\ y_i = 1 \quad (4.1)$$

$$w \cdot x_i + b <= -1, if\ y_i = -1 \quad (4.2)$$

These constraints can be re-written as:

$$y_i(w \cdot x_i + b) >= 1 \quad (4.3)$$

We may approximate this challenging constraint by adding non-negative slack variables ξ_i.

$$y_i(w \cdot x_i + b) >= 1 - \xi_i, \forall i \quad (4.4)$$

This leads to the following optimization problem:

$$\min \frac{\gamma}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i \quad (4.5)$$

$$s.t.\ y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i >= 0\ \forall i \quad (4.6)$$

Here, || • ||2 is the squared L2 norm, which acts as a regularizer on w and guarantees that the most significant possible margin along the hyperplane separates the two classes. A hyperparameter known as > 0 manages the trade-off among the regularization term and the loss function (hinge-loss), penalizing the violation of the restrictions. We determine if a fresh sample x belongs to the specified class using the y = sign(w • xi + b) formula. SVM is frequently employed as the de facto baseline in classification applications. Its valuable benefits include a convex optimization problem, wide margin guarantees, high generalization, scalability, and quick testing time.

4.2.3 Decision Trees

Decision trees (Quinlan, 1986) are a powerful and straightforward form of multiple variable analysis used for attribute values to class mappings. Instead of multiple linear regressions in statistical form analysis, they can be used as intelligent business systems requiring multidimensional data analysis(Lomax and Vadera, 2013). A decision tree is a structure with a leaf node labeled with a class or structure, with a test node linked to multiple subtrees. The test node computes outcomes based on an instance's attribute values, with each possible outcome associated with one of the subtrees. An instance is classified by starting at the tree's root node, with the outcome determined if it is a test node. The process continues using the appropriate subtree. When a leaf is encountered, its label gives the predicted class of the instance. The decision tree classifier is used in decision-making processes (Quinlan, 1996)., as shown in Figure 4.3.

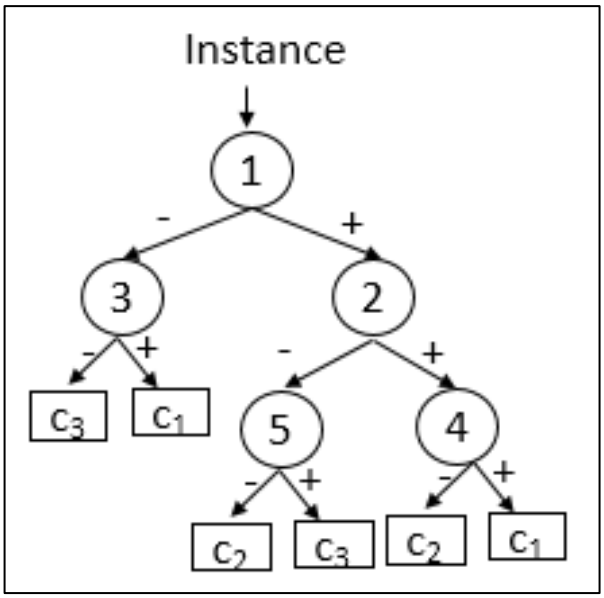


Figure 4.3: Design of Decision Tree (DT) classifier.

4.2.4 Artificial Neural Network

Artificial neural networks (ANNs) are computing systems inspired by biological neural networks found in animal brains. ANNs are based on connected artificial neurons, which loosely model the neurons in a biological brain. Each connection, like synapses, can transmit a signal to other neurons. An artificial neuron receives and processes signals, sending them to connected neurons. A non-linear function of its inputs computes the output of each neuron.

The weight of neurons and edges adjusts as learning progresses, with a threshold determining signal strength. Neurons are typically aggregated into layers, with different layers performing different transformations on their inputs. Signals travel from the input to the output layer, possibly after traversing multiple layers.

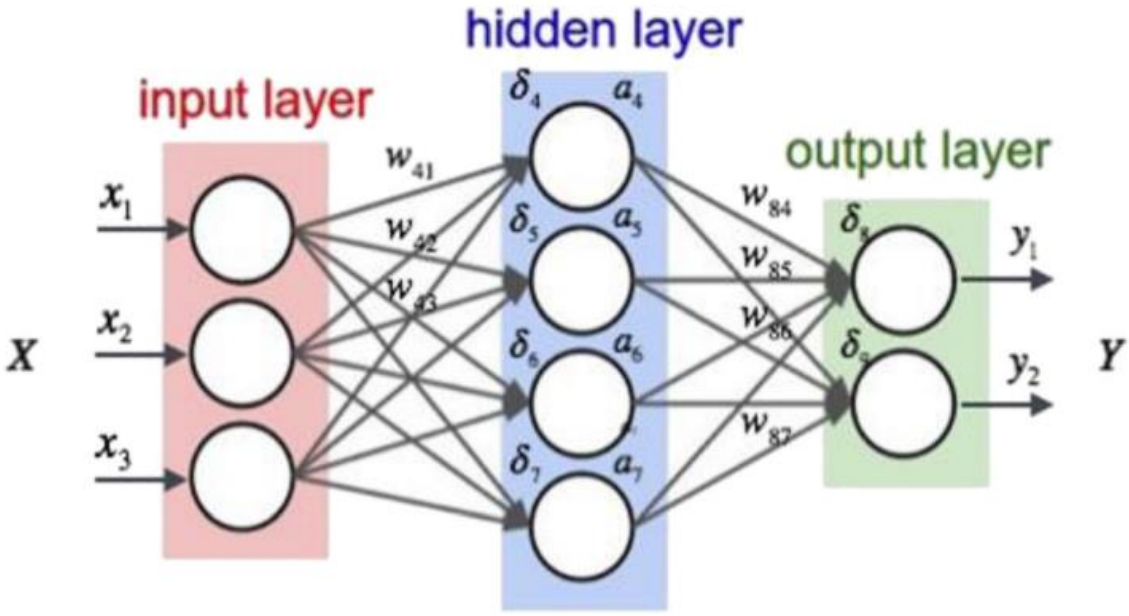


Figure 4.4: A simple Neural Network(Dutta, 2019)

4.3 Unsupervised-learning

Unsupervised learning is a subcategory of machine learning in which models are trained using unlabeled datasets and can operate on the data without being checked by an outside guide. That makes it an ideal ML technique for discovering patterns, groupings, and differences in unstructured data. It is well-suited for customer segmentation, exploratory data analysis, or image recognition processes.

4.3.1 Clustering

Finding a limited number of clusters to describe a data set via unsupervised learning is called clustering [58,59]. Clusters are the groupings formed when a clustering procedure has been carried out. Clustering is the process of dividing a massive collection of things into smaller groupings. When the clusters are unknown in advance, the clustering process—which lacks a training phase—is frequently utilized. Indeed, the optimum grouping characteristics should frequently be determined in the initial step. Objects are grouped according to intra- and inter-class similarity characteristics, with each tiny subset constituting a separate cluster. The quality of the clustering output may depend on the algorithm's usage of similarity metrics like Euclidean distance and Manhattan distance between two numerical data items.

Theoretically, clustering is more effective when intra-class similarity is increased, and inter-class similarity is minimized. Objects inside clusters differ from those in other clusters while bearing a substantial similarity to those in the same cluster. Distance measurements and object attribute values are used to compare and contrast objects. The effectiveness of a particular clustering approach is also determined by its capacity to unearth some or all hidden patterns.

4.3.2 Association

Association rules are created by searching data for frequent if-then patterns and using criteria like Support and Confidence to define essential relationships. Support measures the frequency of an item's appearance in the data, while Confidence measures the number of times if-then statements are factual. Lift, a third criterion compares expected and actual Confidence by indicating the expected number of factual if-then statements. Association rules are calculated from itemsets created by two or more items rather than analyzing all possible itemsets. This approach allows for a more meaningful set of rules, as associations are typically made from well-represented rules in the data.

4.4 Deep learning

Artificial neural networks and representation learning are the foundations of the machine learning technique known as deep learning. It might be either fully or partially supervised. Deep-learning architectures like deep neural networks, deep belief networks, and transformers have been applied in computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programs. These methods have produced results similar to or even surpassing human expert performance.

4.4.1 Deep Neural Network

DNNs are neural networks with several layers and certain complexity levels for processing complex input. (Zhu *et al.*, 2015) offered a multimodal deep learning network architecture for ensuring ideal network start-up and learning intermediate descriptions. The distance metric functions were enhanced by utilizing backpropagation and exponentiated gradient online learning approaches. The number of feature dimensions essential for adequate system performance and increasing the robustness of particular deep learning architectures will require more investigation.

4.4.2 Deep convolutional neural networks

CNN is a feed-forward artificial neural network that can deal with input volumes like multi-channeled pictures and has biases and weights that can be learned. Its connection pattern, influenced by biological processes, is similar to how the visual cortex of animals is set up. CNN initially demonstrated the capacity to incorporate several hidden layers, significantly improving the ability to solve Computer Vision issues. CNNs comprise layers of neurons that translate activation volumes—from raw image pixels to class rankings—through a differentiable function. Except for the wholly linked layers at the end of the network, each layer's neurons are only connected to a small portion of the layer before it. A loss function applied to the output of the last layer allows for end-to-end training of the network's parameters. The CNN architecture comprises an input layer, a stack of convolutional, ReLU, pooling, and, lastly, fully connected layers.

Input Layer: The input is an image with the dimensions height, width, and depth that contains the three color channels (R, G, and B) as well as their raw pixel values.

Convolutional Layer: The fundamental component of a CNN is the convolutional layer, which is made up of a set of parameters that may be learned and are referred to as filters. These filters are 3-dimensional volumes that are tiny spatially (in terms of width and height) and have the same depth as the input volume (Figure 4.4). During the forward pass, each filter is spatially convolved over the input to calculate the dot products between the filter (weights and biases) and the local input area at any place in the space of the input volume. The result is a 2-dimensional activation map displaying filter responses at various spatial places. Alternatively, the network learns filters that turn on when it finds a certain kind of feature (such as edges with a particular orientation, particular patterns, etc.) at a particular spatial location in the input. All of the filters' activation maps that are layered along the depth dimension make up the convolution layer's output volume.

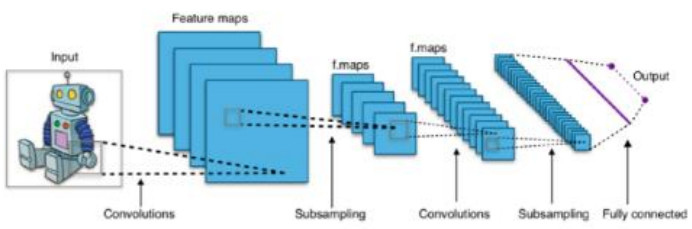


Figure 4.5: Typical convolutional neural network architecture illustration (Moutarde, 2019).

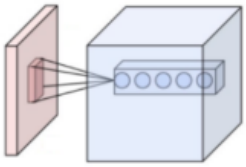


Figure 4.6: the link between the input volume's (red) and convolutional layer's (blue) neurons (Dutta, 2019)

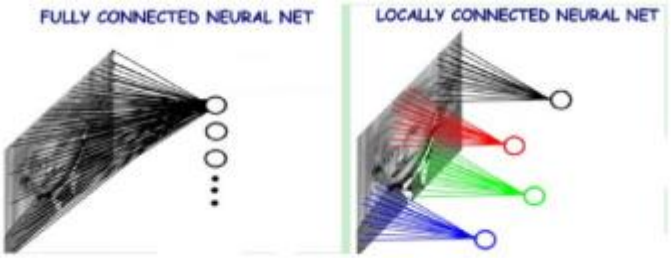


Figure 4.7: Local connectivity of the convolutional layer (Dutta, 2019)

Figure 4.8: Examples of pooling layers (Dutta, 2019)

Pooling Layer: To gradually lower the representation's spatial size in the network, the pooling layer downsamples the input beside its spatial dimensions (width, height). The depth dimension is kept constant since it is carried out individually on each input depth slice. Pooling decreases the amount of network parameters, prevents overfitting, and offers a semblance of translation invariance. Max pooling and average pooling are the two most often utilized types. While average pooling alternates the input region it is linked to with the input region's mean (or average) value, max pooling alternates it by the maximum value (Figure 4.6).

ReLU Layer: The ReLU layer employs the element-wise activation function max(0, x), which thresholds the outputs of the neurons at zero. This layer increases the CNN's nonlinearity.

Fully-Connected Layer: As the name suggests, each neuron in a fully connected layer is linked to every neuron in the layer above it (Figure 4.8). In general, any CNN's final fully connected layer for the classification job comprises C hidden units, where C is

the total number of classes. To produce class probability scores for each class, the output of the C classes is processed through a softmax or sigmoid activation function.

4.4.3 Autoencoders

Autoencoders are simple learning circuits that transform inputs into outputs with minimal distortion. Introduced in the 1980s by Hinton and the PDP group (Rumelhart, Hinton and Williams, 2013), they address the problem of "backpropagation without a teacher" by using input data as the teacher. Together with Hebbian learning rules(Attneave, B. and Hebb, 1950; Oja, 1982), autoencoders provide a fundamental paradigm for unsupervised learning and address the mystery of how synaptic changes induced by local biochemical events can be coordinated in a self-organized manner to produce global learning and intelligent behavior. Recently, autoencoders have taken center stage in the "deep architecture" approach, where they are stacked and trained unsupervised, followed by a supervised learning phase to fine-tune the entire architecture. These deep architectures have been shown to lead to state-of-the-art results on various classification and regression problems. However, little theoretical understanding of autoencoders and deep architectures has been obtained. The term "deep" may have created confusion, as a deep architecture should have nα polynomial-size layers. However, the architectures described in Hinton et al. (Hinton, Osindero and Teh, 2006) and Hinton and Salakhutdinov (Hinton and Salakhutdinov, 2006) seem to have constant or, at best, logarithmic depth, making it difficult to distinguish between finite and logarithmic depth for typical values of n used in computer vision, speech recognition, and other problems.

An auto-encoder consists of 3 components: Figure 4.9 encoder, code, and decoder. The encoder compresses the input and produces the code; the decoder then reconstructs the input using this code. As mentioned earlier, the encoder and decoder are fully connected feedforward neural networks or ANNs. Code is the one layer of an ANN with the chosen dimensions. We set the hyperparameter "code size" before training the autoencoder, which refers to the number of nodes in the code layer. The encoder, a fully connected ANN, creates the code by first processing the input. The decoder, which has an ANN structure, solely uses the code to generate the output. To provide a result that is identical to the input. Keep in mind that the design of the encoder and decoder are identical. Although not required, this is frequently the case. The sole need is that the input and output dimensions must match. Any object in the center is playable. Before training an autoencoder, we need

to establish four hyperparameters: Similar to ANNs, autoencoders are trained using backpropagation.

Code size: number of nodes in the middle layer. A more minor size results in more compression.

The number of layers: We are free to choose the depth of the autoencoder. Without considering the input and output, the encoder and decoder in the

The number of nodes per layer: Since the layers are placed one on top of the other, the autoencoder architecture we are developing is known as a stacked autoencoder. Stacks of autoencoders typically have a "sandwich" appearance. With each additional encoder layer, the number of nodes per layer drops and then rises in the decoder. The decoder and the encoder are also symmetric in terms of layer structure. As previously said, we have complete control over these parameters; thus, this is unnecessary.

Loss function: We employ binary cross-entropy or mean squared error. Cross-entropy is commonly used if the input values are within the range [0, 1] unless the mean square error is used.

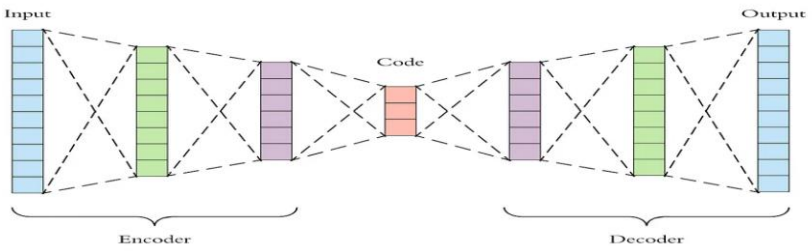


Figure 4.9: Architecture of autoencoder[9]

[9] Applied Deep Learning - Part 3: Autoencoders. https://towardsdatascience.com/. Arden Dertat. Oct 3, 2017

4.5 Conclusion

Machine learning is an artificial intelligence field that focuses on designing algorithms that can learn from and predict data. Its goal is to automate analytical model building and enable computers to learn from data without explicit programming. However, the effectiveness of machine learning relies on the data quality used to train the algorithms. High-quality, representative, real-world data is crucial for accurate predictions. The following chapter will discuss using those strategies to present our progressive approaches.

Chapter 5

Building Detection methodology and Experimental Results

Chapitre 5

Building Detection Methodology and Experimental Results

5. 1 Introduction

Object extraction from data collected by remote sensing has long been a significant study area in computer vision. The development of maps, urban city planning, and land use analysis are some practical uses of this topic. For instance, building extraction is one of the most crucial tasks in retrieving map features. Identifying building locations is crucial for various applications like mapping, military situations, natural disaster management, environmental preparation, and urban planning, ensuring active engagement, counter-terrorism, and peacekeeping measures (Benz *et al.*, 2004; Ghandour and Jezzini, 2018b; Shen *et al.*, 2019; Ullo *et al.*, 2020; Hou *et al.*, 2021; Sirko *et al.*, 2021). Although it is possible, it might take an extensive amount of time or be hard to discern buildings from the images. Because of its intricacy, automatic building detection from aerial or satellite imagery is an essential and challenging task.

Furthermore, due to the fast-expanding urbanization, the necessity for map revision without incurring huge costs or taking a long time is evolving. This becomes possible through technological advances that allow us to capture very high spatial-resolution imagery. As the ground resolution size of the pixels is substantially less than the typical size of objects in such imagery, automatic building detection becomes achievable. A lot of research on building extraction has been conducted during the last ten years (Karantzalos and Paragios, 2008; Zhang *et al.*, 2017; Aamir *et al.*, 2019)**.** However, detecting buildings in urban areas remains problematic due to their diverse shapes, sizes, colors, textures, and similarities between buildings and non-building objects.

5.2 Supervised Machine Learning Approach

5.2.1 Literature Review and Related Work

Most building detection techniques have relied on artificial characteristics (Cohen *et al.*, 2016). They applied the Haar feature approach to implementing machine learning methods to the low-quality RGB geophotos in order to decrease the difficulty of features extracted from such images. (Cretu and Payeur, 2013), Combines traditional and contemporary methods for extracting image descriptions using a collection of algorithms encouraged by the human visual system. Next, machine learning is used to analyze the description of features to determine buildings.

Furthermore, new systems based on deep learning have been suggested (Kang *et al.*, 2018), suggesting an overall classification paradigm using convolutional neural networks. The authors (Guo *et al.*, 2017) offered a collection of convolutional neural networks that could be used with a pixel-level segmentation approach to identify township buildings. (Nogueira, Penatti and dos Santos, 2017), Concentrating on three distinct applications of convolutional neural networks for remote sensing imaging. (Zhang *et al.*, 2020) provides a building detection framework based on a convolutional neural network (CNN), an edge recognition technique, and a building pattern. The building was extracted from high-resolution remote sensing images using the masking R-CNN Fusion Sobe methodology. However, the outcomes demonstrated that it fails severely to extract edges and maintain the authenticity of generated instances.

However, These studies have some drawbacks, namely that the absence of data sets prevents CNN from properly learning the features of the layered spatial in the images and that adding additional layers to the deep model causes more substantial training errors. The ability of CNNs to forecast with less processing comes at the expense of decreased output precision.

The suggested method concentrates on building identification from 2D satellite or aerial images. Consequently, a unique technique handles two fundamental problems: (a) the difficulty with certain buildings' specific colors that are overly similar to other elements in urban imagery and (b) the variation in color shades on buildings' roofs.

Since buildings have so many unique qualities, as we previously stated, building identification is one of the most complex problems to tackle. Several building detection methods have been offered throughout the years, and their effectiveness has been evaluated in various ways. We examine works that make an effort to address those issues in this area. Most of those studies fall into one of three categories: area-based, supervised, or automated extraction of geometric characteristics (Shorter and Kasparis, 2009). A different categorization is made using height data, basic 2D images, or an integration (Ghanea, Moallem and Momeni, 2014).

In (Chen, Shang and Wu, 2014) the authors used the segmentation and classification of color features to create a supervised method based on shadow position. They used the segmentation technique first to divide the picture into superpixel patches. Following that, using a previous selection of selected patches, three classes—buildings, non-buildings, and shadows—were identified using support vector machines (SVM) and linear discriminate analysis (LDA) color characteristics. Finally, they designate a seed point location using the information of where the shadows will fall in the future and using the regional growth approach, they calculate where the buildings will be. In (Salehi *et al.*, 2012), The authors use spot height vector data to build an object-based categorization for metropolitan environments. After segmenting the image, they classified the results into five classes based on the evaluation of the merged spectral, textural, morphological, contextual, and class-related features to allocate a class level of membership to each section (object) based on membership functions or the thresholds: vegetation, shadows, parking lots, roads, and buildings. In (Manandhar, Aung and Marpu, 2017) begin by identifying buildings using a one-class SVM, then use the texture segmentation approach with a conditional threshold level to separate outbuildings of various colors and forms. Regarding the angle of shadows, roads and plants are separated from the buildings. Using Nadir Aerial Images, the authors (Shorter and Kasparis, 2009) have created an automatic secondary classifier to recognize vegetation, buildings, and non-building items with the caveat that the building has a curved roof. They decrease the number of colors from 255 to 17 for every RGB band. Then, segmentation and thresholding are used on the green color band to pinpoint plant areas and the distinction between the blue and green color bands to pinpoint shadows. Entropy filtering and watershed classification are used to measure the solidity of the areas where buildings and non-buildings are found.

In (Ghandour and Jezzini, 2018a), The Building Detection with Shadow Verification (BDSV) method was developed to recognize buildings by combining many criteria, including color, form, and shadow. Because non-tile flat roofs rely on the form features, some roofs, such as those on buildings with sloping roof tiles, can only be retrieved using color cues. The shadow qualities were also included, and candidates with dense shadows would be considered actual buildings.

In 2016 (Kohli, Sliuzas and Stein, 2016), In order to identify slums using the morphology of the built environment, the authors devised an ontology-based approach. The segmentation step in this method is followed by hierarchical classification object-based image analysis. Depending on the classification's objective, each object's spectral values, form, texture, size, and contextual links are all calculated.

The authors proposed (Grigillo, Kosmatin Fras and Petrovič, 2011) an automated method utilizing a multispectral orthophoto and a digital landscape model. The standardized digital landscape model was first used to construct a building mask containing only potential building positions. Then, using an altered vegetation index based on the usage of the near-infrared orthophoto and the adjustments of the vegetation index with the shadow index and the texture evaluation, the vegetation was removed from the building mask. Finally, they determine the building location using the Radon transformation.

In (Wang *et al.*, 2015), Based on the rectangle-shaped structures, the authors suggested an automated method. Using a generated bilateral filter, they improve the edge contrast. After that, lines are extracted using a line segment detector. Lastly, a perceptual grouping method groups previously discovered lines into potential rectangular buildings.

In (Karantzalos and Paragios, 2008), The authors presented an automated method based on level set segmentation with previous knowledge of recognized building form models as constraints.

In (Awrangjeb, Ravanbakhsh and Fraser, 2010), The authors demonstrate an automated method that uses multispectral images and LIDAR data—using a threshold for the height of objects like trees and buildings to isolate from the others. Then, the trees are removed from an orthorectified multispectral image using the normalized difference vegetation index approach.

In (Stankov and He, 2013), The authors created a grayscale image with an improved potential for building localization by implementing an automated technique based on the similarities of building roofs utilizing a previously established reference set. Next, they use the hit-or-miss transform morphology to allocate pixels to potential building positions or nonbuilding sites. Finally, they confirmed the final position of the buildings after establishing the shadow regions.

The authors (Nosrati and Saeedi, 2009) suggested a supervised model that merged the active contour approach with local texture and edge-related data provided by starting seed points on one of the buildings.

In (Lv *et al.*, 2016), a new object-based filter that first divides the image into uniform objects using multi-scale segmentation while also extracting the features vector of those objects, considering that each splitter object is a portion of an actual, fully-existing object in the image and is in the center of his surrounding neighboring. There are thus two options for his location: either within the actual object or outside of it. As a result, it shares characteristics with nearby objects or those at the boundary without them. Therefore, it is suggested that the neighboring objects under consideration be chosen using topology and feature restrictions. By computing the average of the feature of the chosen object, the feature of the core object is finally smoothed.

Nevertheless, building extraction still faces several difficulties. The methods based on color categorization cannot distinguish buildings from trees or lawns since some buildings have a specific color (green, red, etc.) (Shorter and Kasparis, 2009; Awrangjeb, Ravanbakhsh and Fraser, 2010; Grigillo, Kosmatin Fras and Petrovič, 2011; Chen, Shang and Wu, 2014; Ghanea, Moallem and Momeni, 2014). Using a specified shape database or assuming that buildings have a typical shape, like rectangles, will end up in results that are closer to a specific format (Karantzalos and Paragios, 2008; Wang *et al.*, 2015). We are unable to cover all of their possible forms. Furthermore, some research employs texture to address the color issue. However, they only utilize it to identify the entire area of buildings without distinguishing one from another or to compute the entropy or homogeneity of a single pixel paired with the high data of objects (Salehi *et al.*, 2012; Kohli, Sliuzas and Stein, 2016). Therefore, these methods cannot provide the results we seek without much data or in a complicated setting. Given that, and supposing that they are small-sized objects, could we profit from segmentation to a superpixel width unity by using the texture

algorithms on the full superpixel rather than on a single pixel? The general process of our proposed approach is as in the figure 5.1.

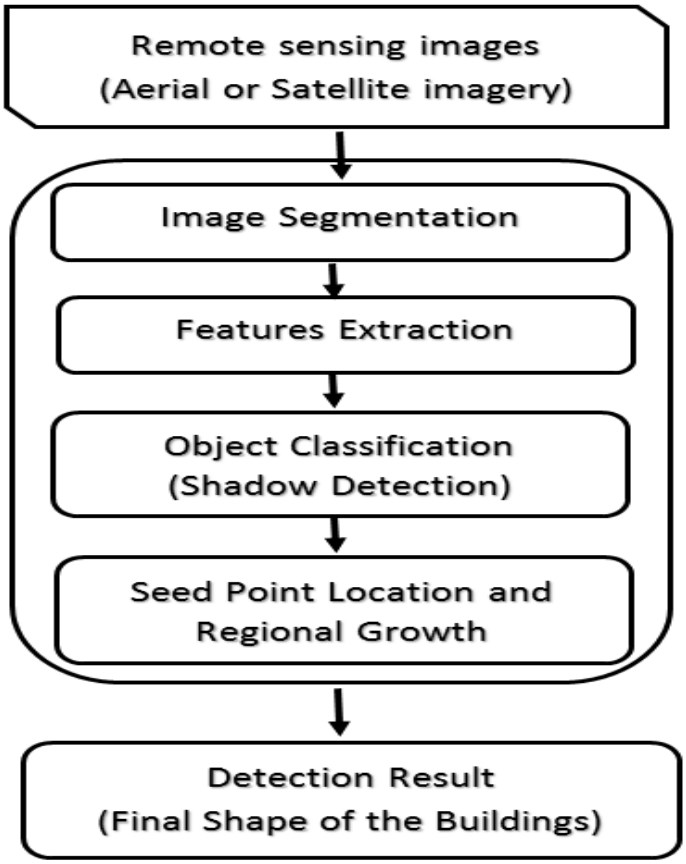


Figure 5.1: Diagram flow of the proposed algorithm(Benchabana *et al.*, 2022)

5.2.2 Features Extraction and classification of image data:

Six different types of land cover are typically included in urban images: trees, grass, shadows, roads, parking lots, and buildings. Grass and trees are often green; however, this might vary depending on the kind and season of the image (they could also be red or yellow). In contrast to the shadows, parts of the roads and parking lots are darker or entirely black. Finally, depending on the location of the image (the variety of cultures and climates), buildings have varied colored rooftops. Figure 5.2 illustrates the study area's different structures, architectural styles, lighting features, landscape features, and complex structures. Complex construction patterns can be rapidly found by visual inspection but not by machine learning.

As a result, parking lots, highways, and trees may all have the same hue as certain buildings. Therefore, we used two approaches to address this issue: (a) the shadow factor

to distinguish between buildings and parking lots and (b) the texture features of trees. Consequently, four classifications emerge based on color and texture: buildings in the first, trees and grass in the second, roads and parking lots in the third, and shadows in the fourth. Hence, three stages have been considered in order to acquire those classes. We will go through them in depth.

Figure 5.2: Aerial shots of houses in the New Zealand study area taken from various angles

5.2.2.1 Superpixel Segmentation Using SLIC

Superpixel segmentation separates an image into a collection of associated pixels with corresponding colors. We can divide a large image into superpixel patches rather than dealing with pixels to preserve as much information as possible. In the 5-D space [l, a, b, x, y], where (l; a; b) is the CIELAB color space and (x; y) is pixel coordinates, the Simple Linear Iterative Clustering (SLIC) technique for superpixel segmentation described by Achanta and Shaji (Achanta *et al.*, 2012) is a k-means-based regional clustering of pixels. By incorporating a new distance measure, Ds, as shown in Eq. (5.1), SLIC modifies the k-means clustering technique to construct superpixels effectively.

$$D_s = d_{lab} + m/S\, d_{xy} \tag{5.1}$$

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{5.2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{5.3}$$

Where m is a variable that enables changing the compactness of superpixels, S is the grid spacing between them, and k, i are the index of the middle and surrounding pixels of the superpixel, accordingly.

Given the previous, this stage seeks to produce superpixels that are as homogeneous as feasible and large enough to allow us to extract texture information without respect to the geometry of any particular superpixel. Therefore, a high value of m causes the spatial distances to exceed the color factors, resulting in more compressed set-sized superpixels, disregarding the boundaries of the objects in the image. Superpixels of small sizes will result if the value is lower, the opposite of what is expected. So, We choose a low-value number in order to achieve good outcomes. The next step is to merge each superpixel smaller than the thresholding number of pixels to its closest neighbor, a superpixel. While maintaining object boundaries, it will combine them into a single bigger superpixel.

5.2.2.2 Texture Features Extraction Using (ICICM)

One of the most effective but straightforward texture descriptors is the color co-occurrence matrix (CCM). It involves taking statistical readings from the image regarding how various colors mix. Vadivel et al. (Vadivel, Sural and Majumdar, 2007) developed the Integrative Color Intensity Co-occurrence Matrix (ICICM) as a CCM extension to mimic how people perceive textures. They claimed that each pixel may be considered a color or a grayscale depending on the contrast level. Therefore, the amount of color and gray have been determined by two measures, Wcol and Wgray, which have been retrieved from each pixel in the image. After that, these measures were used to extract four co-occurrence matrices representing the co-existence of W_{col} / W_{col}, W_{col} = W_{gray}, W_{gray}=W_{col}, and W_{gray} = W_{gray}. Finally, image descriptors based on third-order statistical events have been created. In [95], an enhancement to ICICM was put out using a smooth method for color/gray-level space normalization. Ultimately, the texture characteristics of each finalized superpixel were extracted using ICICM.

5.2.2.3 Identifying Classes Using SVM

A separating hyperplane is the formal definition of the supervised discriminative classifier called the Support Vector Machine (SVM). The SVM training methods build a model that can classify the incoming data into one of those classes, given a collection of training samples and their respective classes. We can thus achieve our final class findings

by training the SVM classifier with a set of combination vectors derived from the LAB color characteristics and the texture features of the training superpixel samples. Figure 4.3(d) displays the findings of a primary linear kernel type of SVM used to analyze data from the four groups.

5.2.3 Accurate Building Position

According to the prior findings in Figure 5.3(d), buildings and trees are entirely autonomous, compared to some commonalities with highways and parking lots. As we previously discussed, the ability to discern between elevated and ground-level objects depends on previous knowledge of the direction of the shadows. How to identify the precise geometry of the buildings remains a challenge.

Considering the building's rooftop often has the same shade, a seed point position using the regional growth approach could work to identify buildings. However, in many places worldwide, relying on the designs of the buildings could result in darker sides to display on the rooftops. Therefore, we adjusted our new regional growth application to address this issue.

5.2.3.1 Seed Point Location and Regional Growth

Initial superpixels seed locations are defined based on three criteria:

- superpixels with a neighbor from the trees class and those in the shadows class are removed;

- Superpixels considered seed points must be in buildings class adjacent to shadows class after the elimination, based on their respective direction;

- Superpixels with more road class neighbors over building class are removed.

Then, using the earlier initial superpixel seed points and the [a, b, ep, h, c, en] vector with the entropy, homogeneity, correlation, and energy from the texture features as inputs, we applied the regional growth technique to ensure that the seed points are dispersed throughout the building's location and overcome the darker side problem. Assuming that Vi is the vector of the initial superpixel seed points Si and Vs are for the other neighbors' superpixel Ss, the subsequent logical criteria have been applied, and the outcomes will be our final superpixel seed points.

$$V_s - V_i \leq T_s \quad (5.4)$$

$$S_s \in C_1 \quad (5.5)$$

Finally, we used an additional regional growth to obtain the entire shape of the buildings. In this case, we started as an initial with the centers C_(i) of each superpixel of the last superpixel seed points result, taking into mind one condition, and utilized the lightness value (L) from the Lab color space for all pixels P_(s) of the image:

$$L_i - L_s \leq T_g \quad (5.6)$$

5.2.3.2 Accurate The Final Shape of The Buildings

The final stage reveals that there are still certain noises and gaps, as shown in Figure 4.3(f), and neither the building's outside nor inside are precisely formed and filled. Therefore, more actions are required. An open morphological procedure has been used to filter out the noise and fill in the gaps. Then, using the basic contour procedure, the borders of the buildings were retrieved. Figure 5.3 displays the final results.

5.2.4 Experimental Results

The appropriate selection of algorithm parameters has first been determined in this section. The classification outcomes with and without texture are then presented. Finally, we evaluate the outcomes by comparing them with alternative approaches. The complete experimentation was employed to a dataset from Land Information New Zealand urban aerial image for Auckland with fragment images from a size of 6400x9600 images and 7.5cm ground resolution. The experiment is realized in an Intel 3.2 GHz CPU with 16G memory.

5.2.4.1 Parameters Selection

In order to calculate the three factors (n, Ts, and Tg), where n is the initial amount of superpixels in the SLIC algorithms, Ts and Tg are the thresholding of the regional growth in Eqs (5.4) and (5.6), it is a two-sided task to initialize an appropriate number n of the superpixels. Fewer computations must be performed in a shorter period when n is minimal. On the other side, we achieve better uniformity the larger it is. Having stated that the size

and resolution of the image that has been analyzed are what matter. On a cut from Land Information New Zealand urban aerial images for Auckland with a size of 1286x1249 and 7.5cm ground resolution, we test various values of n, and the outcomes show that 2500 is the ideal number for n. According to that final result, we may create an equation for every image with the same ground resolution.

$$n = \frac{\text{size image}}{\text{size test} / n \text{ test}} \tag{5.7}$$

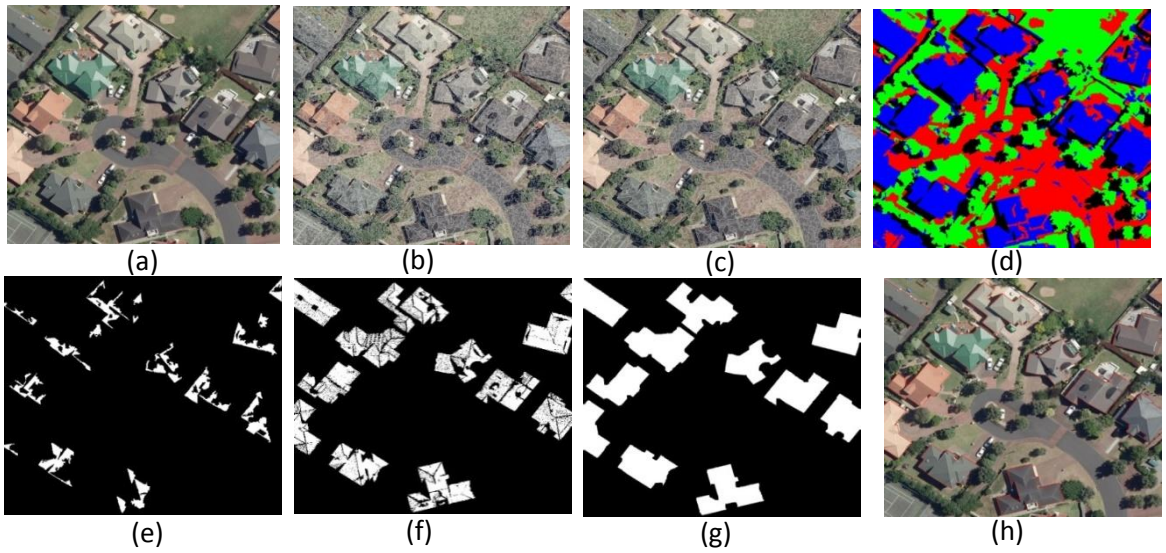
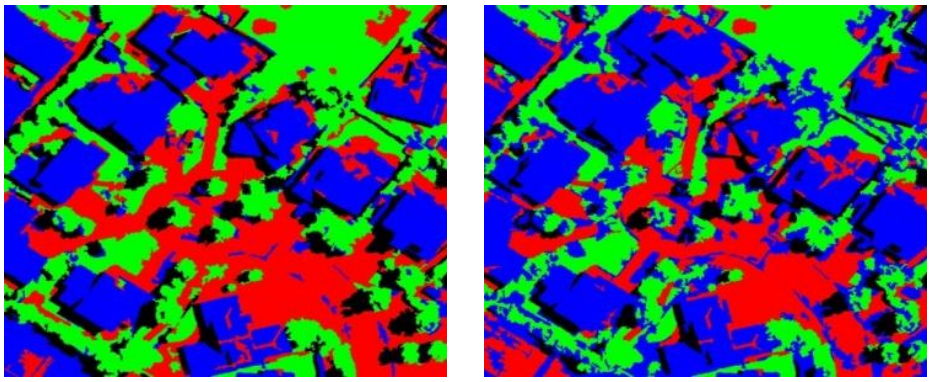


Figure 5.3: Results from multiple algorithmic steps. (a) The original input, (b) Superpixels, (c) Combination Superpixels, (d) classification outcomes, (e) Seed Point Locality, (f) subsequent growth, (g) Final Form of The Buildings, and (h) final result.

5.2.4.2 Classification Results

In order to achieve acceptable classification outcomes, we must focus on two factors. First, as we previously said, properly selecting the initial number of superpixels eliminates overlapping groups caused by the homogeneity leak. The second step is choosing training sets for SVM techniques. The more options we explore, the better the outcomes. Figure 5.4 displays the classification performance assessment with and without adding texture features under these conditions (the type of kernel used for the SVM is linear). The enhancement is much more visible when buildings have colors that match the other classes.



(a) (b)

Figure 5.4: the classification's comparative findings, (a) using textural characteristics to classify, (b) with no texture characteristics, Whereas the blue denotes the buildings, the green, the vegetation, and the red, the streets and sidewalks.

5.2.4.3 Building Detection Comparison

In this section, the images chosen include many building characteristics, including size, form, and various roof shades. To provide a qualitative comparison with our technique, approaches offered by, (Lv *et al.*, 2016) and (Zhang *et al.*, 2020) have been compared to our suggested approach. As seen in Figure 5, the first row of images represents the original input images from which we selected three alternative images, and the second row displays the outcomes of the building extraction process.

First, the terms TP (true positive), which stands for the number of successfully identified buildings; FP (false positive), which stands for the number of erroneously detected buildings; and FN (false negative), which stands for the number of undiscovered buildings, were defined. We utilized the detection rate (DR or precision) to evaluate the degree to which identified buildings are, in fact, buildings to determine the correctness of the building extraction findings. The false-negative rate (FNR) also calculates the proportion of missed detections to all actual buildings. The amount of accurately recognized buildings without changing or enlarging their bounds is known as the COMP, completeness of detection.

Figure 5.5: comparing the findings of three distinct algorithms for building recognition (row a) input images, (row b) The outcomes of the suggested technique, (row c) The method's outcomes of (Zhang *et al.*, 2020)*, (row d) The method's outcomes of* (Chen, Shang and Wu, 2014) *and (row e) The method's outcomes of* (Lv *et al.*, 2016)*. (Red signifies building) .*

$$DR = \frac{TP}{TP + FP} \times 100 \tag{5.8}$$

$$FNR = \frac{FN}{TP + FN} \times 100 \tag{5.9}$$

Table 5.1: Evaluation of the four methods' building identification precision (numerical examination of Figure 5.5).

	Our method	**method** (Chen, Shang and Wu, 2014)	**method** (Lv *et al.*, 2016)	**method** (Zhang *et al.*, 2020)
TP	3719	3799	3806	3802
FP	110	749	1083	324
FN	83	56	23	27
DR (%)	97.13	83.53	77.85	92.14
FNR (%)	2.18	1.45	0.6	0.7
COMP	3570	2966	2323	3445
COMP (%)	95.99	78.07	61.03	90.61

Table 5.1. displays the comparison between our method and currently used techniques. For testing, 50 images of various sizes (3883 buildings). Our approach has a lower missed detection rate (2.18%) than previous methods. Nevertheless, it is more accurate and has a higher detection rate (97.13%, 95.99%). These algorithms mistakenly identified items like crosswalks, lawns, and parking lots as buildings, which explains the high detection rate. In addition, the suggested approach overlooked 164 buildings out of the total number of buildings due to backdrop interference and other visual elements. The tightly spaced little buildings might have been removed and combined into one large building. It is also likely that little buildings are more difficult to find.

5.2.4.4 Computation Time

Calculating time is yet another crucial factor to consider while using this strategy. The suggested techniques were applied in a MATLAB environment; more details on calculation durations are provided in Table 5.2. Fifty test images totaling 3,883 buildings were used to evaluate our proposed method; each line in the table below shows the time that passed for each area. The overall duration was 2146.18 seconds, with a 42.92-second average.

Segmentation is the fastest stage, frequently occurring in only 0.34 seconds. On the other hand, the SVM classification takes a long time, taking up 92.1 percent of the total time. However, processing an image takes just 42 seconds on average, substantially faster than the methods described in (Chen, Shang and Wu, 2014) and (Lv *et al.*, 2016).

Figure 5.6 demonstrates that the segmentation images assessment took 0.06 seconds longer than the feature extraction time, which was 0.03 seconds in Figure 6a, and that the zoning time, shown in Figure 5.6b, was less than the feature time needed for extraction. This is because the areas in the 50 processed images had entirely distinct dimensions from the evaluated image's area (figure 5.6B), which has dimensions of (1286*1249), including 2537*3665, 2961*1761, 3465*5601, etc.

Table 5.2: The duration of each phase of the intended building's identification

Section	Total Time (s)	Average time (s)	Percentage (%)
Image segmentation	17.02	0.3404	0.8%
Feature extraction	65.83	1.3166	3.1%
Object classification (shadow detection)	1975.58	39.5116	92.1%
The final shape of the building	87.75	1.755	4.1%
Total	2146.18	42.92.36	100%

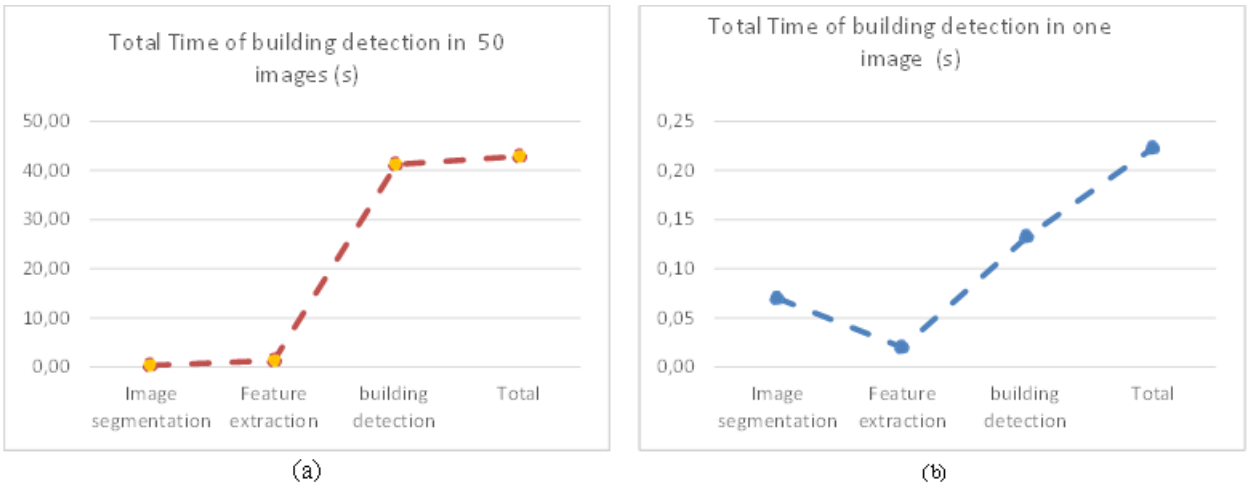


Figure 5.6: Computing time for building recognition in seconds. (a) Overall Time of building recognition in 50 images and (b) Time of building recognition in one image.

5.3 Deep Learning Approach

5.3.1 Literature Review and Related Work

Building and other object detection in remotely sensed images have attracted much study attention recently (Deng *et al.*, 2022; Zheng *et al.*, 2022; Cao and Huang, 2023; Chen *et al.*, 2023; Khan, Alarabi and Basalamah, 2023; Kokila and Jayachandran, 2023; Nurkarim and Wijayanto, 2023), with several suggested methods. According to the theory that U-Net does well in recognizing irregular edges, the authors (Kusz *et al.*, 2021) introduced a modified U-Net convolutional neural network segmentation to recognize buildings via LiDAR data. According to the study, the model did well for domestic buildings but failed with bigger ones and had trouble recognizing diagonally angled ones, leading to some combined or divided buildings.

In (Chen *et al.*, 2022), The Res2-Unet model was developed by the authors, who used multiscale learning to enlarge the responsive fields for every bottleneck layer. They recommended applying a boundary loss function to enhance detection efficiency and provide precise building borders. The model, however, was still unable to distinguish between certain roads and buildings, and occasionally, it mistakenly identified background objects as buildings while completely deleting a few with spectral and textural characteristics resembling the surrounding countryside.

In (Ojogbane *et al.*, 2021), a High-resolution aerial imager, the digital surface model (DSM), and deep feature extracting by combining prior channels have all been utilized to locate buildings in three simultaneous CNN stream channels. Morphological procedures produced the final building form. Although this method accurately detects tiny structures, it cannot identify spaces between buildings in heavily inhabited or low-lying areas.

However, the conventional CNN (pixel-based deep learning technique) needs a lot of processing and storage capacity. Consequently, superpixel-based grouping has received much attention recently. Moreover, in (Mao, Li and Sun, 2019), The CNN model is trained to classify superpixels produced by Simple Linear Iterative Clustering (SLIC) using OpenStreetMap as semantic tags. Their primary issue was that certain superpixels included both building and non-building pixels, which caused feature data to overlap and classification results to be incorrect. Furthermore, (Lv *et al.*, 2019) utilize the same suitable circumstances from earlier experiments to assess the compactness index value of four

superpixel techniques. They discovered that the tightness values for the SLIC and SEED superpixel techniques are equally low, but they favored the latter due to the segment's alignment with object boundaries.

In (Sun *et al.*, 2021), The ConvCRF model, as suggested by the authors, combines CNN and conditional random field CRF with a superpixel border constraint. They compared SVM, CNN, and other widely used machine-learning methods. It has been demonstrated that the deep models are less sensitive to comparable object types with weak backscattering. In addition, (Song *et al.*, 2020) presented a fused DeepLab v2 neural network classification result with the SLIC segment for boundary recognition to evaluate earthquake-damaged buildings. The background noise was then reduced using a mathematical morphological technique, eliminating the tiny lines at the borders. So, it is essential to avoid using morphological procedures at random.

5.3.2 Methodology

The proposed SP_VAE-CNN approach for building detection, as illustrated in Figure 1, is composed of four primary phases: image segmentation using adaptable SLIC into patches, extracting features out of the segmented patches employing VAE, classification of these features using CNN (Bensaci *et al.*, 2021), and finally, a regional growth from the seeds point location followed by the morphological operation techniques to identify the final form of the buildings.

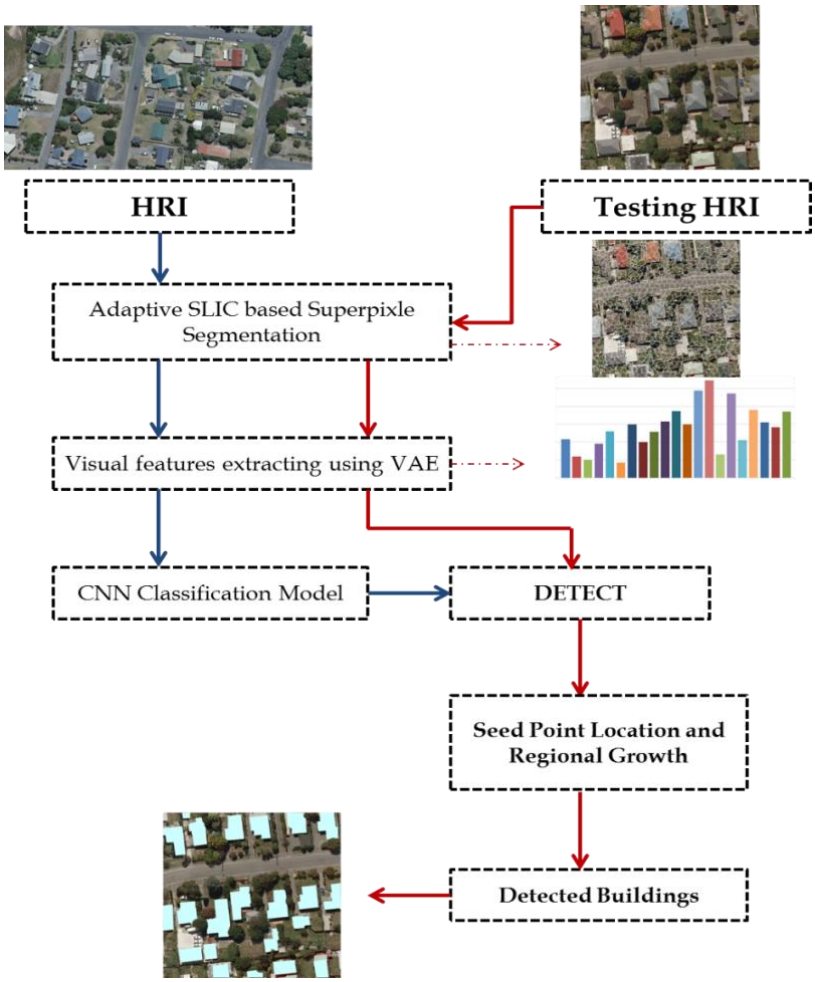


Figure 5.7: An overall process of the building detection technique we suggest. Solid blue arrows represent training imagery, while solid red arrows represent test imagery.(Benchabana *et al.*, 2023)

5.3.2.1 Superpixel segmentation

The effectiveness of superpixel segmentation techniques has been demonstrated in computer vision. They have effectively reduced the quantity of image primitives required for subsequent processing. Superpixels, the compressed version that combines similar pixels into homogenous patches with perceptual implications, substantially decrease the computing time. Simple linear iterative clustering (SLIC) has specifically proven to be beneficial in terms of object boundary adherence, speed, and little memory needed (Achanta *et al.*, 2012). Depending on how similar their colors are and how close together they are in the image, the pixels are localized into superpixels using the k-means algorithm. Each pixel is often represented as a five-dimensional [l, a, b, x, y] feature vector, where [l, a, b] are the channels of the CIELAB color space and [x, y] are the location of the pixel from which the distance is calculated. Color values offer little information and are primarily used to calculate a superpixel's components.

Additionally, the distance-weighted controls the spatial closeness, which may cause boundary warping and losses according to their high values, regulating the size and density of the superpixel. We compute the texture features using the integrative color intensity co-occurrence matrix (ICICM) (Vadivel, Sural and Majumdar, 2007) and then extend the feature vector into an eight-dimensional [l, a, b, e, h, c, x, y] space, where e, h, and c are accordingly the energy, homogeneity, and correlation. Eq(5.10) provides the formula for the distance.

$$D_s = \alpha D_{lab} + \beta D_{ehc} + \gamma D_{xy} \tag{5.10}$$

$$D_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{5.11}$$

$$D_{ehc} = \sqrt{(e_k - e_i)^2 + (h_k - h_i)^2 + (c_k - c_i)^2} \tag{5.12}$$

$$D_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{5.13}$$

Whereas k and i are the index of the centered and surround pixels of the superpixels, correspondingly, α, β, and γ are the balancing weight factors that regulate the proportion of color resemblance, texture, and spatial closeness of superpixels. S is the grid gap among them.

Figure 5.8: Outcomes of SLIC segmentation; (a) with a five-dimensional vector [l, a, b, x, y]; (b) with an eight-dimensional vector [l, a, b, e, h, c, x, y].

Figure 5.8 demonstrates how adding texture characteristics to the distance equation considerably enhances the segmentation outcome, particularly in areas with dense vegetation like trees. Adjustments were made for bare soil and driveway sections, even

though the building parts had no notable alterations. The portions had more variation in content and were formed arbitrarily.

5.3.2.2 Variational Autoencoders for Feature Extraction

Unsupervised neural networks that use an encoder and a decoder to develop efficient representations of input data are known as autoencoders. With the help of the encoder, features may be extracted from untrained input and turned into a latent representation (Bengio, 2009). Variational Auto-Encoder by (Kingma and Welling, 2013) is a cutting-edge non-linear feature extraction method that can accurately and repeatedly characterize data structure. The autoencoder model's design is purposefully restricted to a bottleneck at the model's midline, where the input data rebuilding is executed. The latent representation must adhere to a normal distribution according to a variational constraint in VAE for the decoder's output distribution to match the observed data. The latent outputs are chosen at random from the encoder's learned distribution. Figure 5.9 displays the VAE's network architecture.

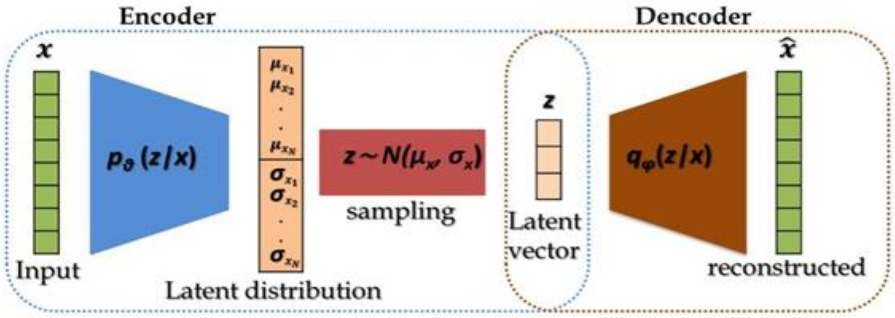


Figure 5.9: Fundamental aspects of the Variational Auto-Encoder.

x and $\hat{x}$ signify the original and updated versions of the initial data. μ and σ indicate the latent variable's Gaussian distribution's mean and variance. z is an instance from $N(\mu, \sigma^2)$ and h is the network's hidden layer. The main goal of VAE is to teach a network how to recreate its input data as x using the following loss function:

$$L(x, \hat{x}) = \|x - \hat{x}\| \tag{5.14}$$

Let us consider a dataset $D = \{x_1, x_2, \ldots, x_N\}$ of N distinct variables with similar distributions, each representing a realization of random variable $x \in X$. We suppose that a random process including an unknown continuous variable z produced from a previous normal distribution created $p_\theta = N(\mu, \sigma^2)$.

The actual posterior density $p_\theta(z|x)$ is intractable. Hence, we use a recognition model. $q_\varphi(z|x)$, which approximates the unsolvable true posterior $p_\theta(z|x)$. We will reduce the K.L. approximation's departure from the genuine posterior. While K.L. divergence is zero, $q_\varphi(z|x)$ is equal to $p_\theta(z|x)$, that is $p_\theta(z|x) = q_\varphi(z|x)$. The K.L. approximation's departure from the genuine posterior $D_{KL}\left(q_\varphi(z|x)\|p_\theta(z|x)\right)$ can be written as follows:

$$\begin{aligned} D_{KL}\left(q_\varphi(x)\|p_\theta(x)\right) &= \int_{-\infty}^{\infty} q_\varphi(x)\ log\ log\ \frac{q_\varphi(x)}{p_\theta(x)}dz \\ &= log\ log\ p_\theta(x) + D_{KL}\left(q_\varphi(x)\|p_\theta(z)\right) \\ &\quad - E_{q_\varphi(x)}[log\ log\ p_\theta(z)\ ] \geq 0 \end{aligned} \tag{5.15}$$

That is:

$$\begin{aligned} log\ \log p_\theta(x) \geq\ & - D_{KL}\left(q_\varphi(x)\|p_\theta(z)\right) \\ & + E_{q_\varphi(x)}[log\ log\ p_\theta(z)\ ] \end{aligned} \tag{5.16}$$

The variational lower bound on the marginal probability of data x is the term used to describe the inequality's right half.

$$L(\theta, \varphi; x) = - D_{KL}\left(q_\varphi(x)\|p_\theta(z)\right) + E_{q_\varphi(x)}[log\ log\ p_\theta(z)\ ] \tag{5.17}$$

The first term $D_{KL}\left(q_\varphi(z|x)\|p_\theta(z)\right)$ of Eq. (5.17) can be combined analytically, and the second term $E_{q_\varphi(z|x)}[\log p_\theta(x|z)]$ requires approximation by sampling:

Initially, we reparameterize the approximation$q_\varphi(z|x)$ using a differentiable transformation g_φ (*x*,) of an auxiliary noise variable; Then, we estimate $E_{q_\varphi(z|x)}[\log p_\theta(x|z)]$:

$$E_{q_\varphi(x)}[log\ log\ p_\theta(z)\ ] = \frac{1}{M}\sum_{m=1}^{M} log\ log\ p_\theta(z^m) \tag{5.18}$$

The parameters φ and θ of Eq. (5.17) are valued via a fully connected neural network. Moreover, parameters can be refreshed using SGD, Adagrad(Duchi, Hazan and Singer, 2012), Adadelta(Zeiler, 2012), and Adam (Kingma and Ba, 2015) optimizer.

Only the encoder has been retrieved and utilized for feature extraction from segments of the images once the VAE's training is finished.

5.3.2.3 Accurate Buildings Locations and Shapes

Finding the initial set of superpixel seed locations is the first stage. The main set, which comprises superpixels in the building class and has a neighbor in the shadow's classes, depends on the angle at which the shadow is formed. As a result, we typically add each neighbor to the seed set in a building class with similar characteristics. Then, using the centers of the seed set as an initial location, regional growth is used to obtain the basic geometry of the buildings. In order to recognize the final shape of the buildings, noise and gaps have been eliminated using open morphological techniques.

Algorithm for Accurate Building Locations and Shapes:

1. Select the first batch of superpixel seed points:
    - Determine which superpixels are neighbors with classes from the shadow and which are members of the building class.
    - Add each surrounding superpixel with a building class location and seed set-like characteristics.
2. Regional development for estimating primary form:
    - From the seed set's centers, perform a regional growth operation.
    - Expand the region by including neighboring superpixels with similar features in the building class.
    - Continue doing this until the buildings' main shapes are achieved.
3. Use open morphological techniques to improve the form:
    - Apply open morphological techniques on the primary form to eliminate noise and fill in any gaps.
    - Utilize these techniques to determine the buildings' ultimate shapes precisely.

5.3.3 Experiments and Results Analysis

This section demonstrates the effectiveness of the suggested plan through three subsections. We look at the effects of changing our algorithm's variables in the first subsection. The second one evaluates how different deep learning algorithms operate when it comes to creating detection. Finally, a comparison with particular pertinent works is made to highlight the superiority of our suggested algorithms.

Three datasets with distinct properties were used in the trials in question.

- The WHU aerial images 2016 dataset was initially provided by the New Zealand Land Information Services. We utilized the modified dataset presented by (Ji, Wei and Lu, 2019) for our research. It contains 8,189 512x512 pixel tiles made from aerial images of 18,7000 buildings downscaled to 0.3 meters in ground resolution. The samples were divided into three groups: a training set of 130,500 buildings, a validation set of 14,500 buildings, and a test set of 42,000 buildings.

- Masterton urban aerial image file from Land Information New Zealand: It was chopped into 1024x1024 pixel blocks with a ground resolution of 0.075 m. The datasets were chosen because of their capacity to cover a range of terrain types, various building kinds, and roof color patterns. They were, therefore, ideal for evaluating the efficacy of the suggested building detection methodology.

- From several locations in Touggourt, Algeria, we gathered high-resolution Google Earth images. The images are picked to illustrate various architectural details, including sizes and forms.

The tests were performed on a machine with an NVIDIA GeForce GT 720 GPU card, an Intel® Core i7 CPU operating at 2.00GHz with four cores, and 16GB of RAM. Both Python and Matlab have been used to implement the procedure and evaluate the results.

5.3.3.1. Parameters Selection

a. . SLIC parameter

The number of superpixels in the input image is an essential factor in the SLIC method since it determines the size of the superpixel and may interfere with the semantics of individual image components. Small superpixels will not have enough characteristics for semantic recognition, while bigger superpixels could be able to capture diverse kinds of objects. Initializing an appropriate number of superpixels depends on the dimensions of the researched region images and its least estimated size. We configured it to 16*16 to guarantee that the smallest size has enough information.

b. VAE performances and parameters

The VAE architecture we employed in our tests included four hidden layers—two in the encoder and two in the decoder. The input size was decided based on the largest possible superpixel. The framework consisted of four hyperparameters, including mean square error (MSE) for calculating loss, 50-bit code size, 128 nodes in the hidden layer, and ReLU activation functions for all levels. The code size was selected to 50 since this number produced the most remarkable performance after testing the method with values ranging from 5 to 200. Figure 5.10 illustrates how changing the code size affects the method's accuracy.

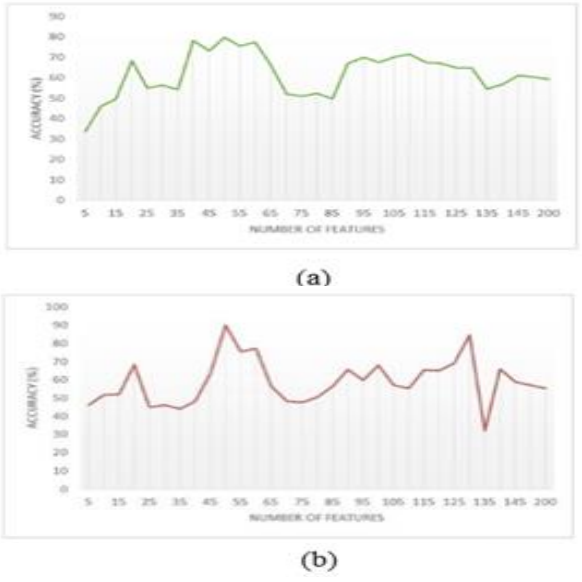


Figure 5.10: Building recognition precision and dimensionality, (a) assessment data (%) IN WHU images 2016 dataset, (b) assessment data (%) IN Land Information New Zealand images dataset of Masterton.

A comparison with two CNN models, Vgg-16 and MobileNet, was done to show how superior the VAE is to the CNN. The effect of utilizing the three models on the accuracy and recall of the classification outcomes is shown in Figure 5.11. The findings of the VAE and Vgg-16 models are similar. However, it takes longer to calculate because the Vgg-16 has more parameters (13 convolutional layers and three fully linked layers).

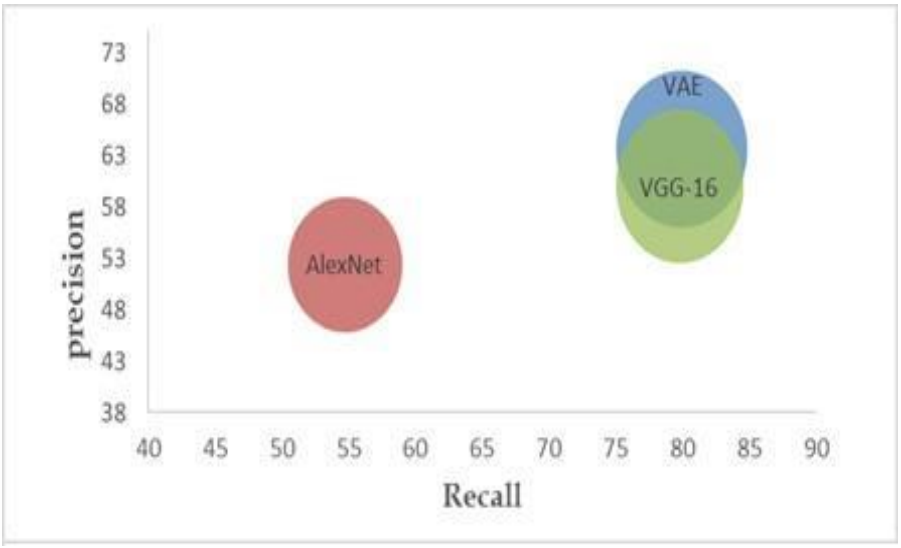


Figure 5.11: The effects of the models VAE, Vgg-16, and MobileNet on the classification outcomes.

c. CNN hyperparameters

The number of filters, optimizer, hidden units in the fully connected layer, batch size, and dropout rate are just a few of the hyper-parameters that CNN models consider. We used a grid search to find the ideal hyper-parameters for the CNN model. An overview of the training's best results and the associated hyper-parameters is shown in Table 5.3.

Table 5.3: The Values Of Tuning CNN Hyperparameters

Parameter (Items)	Search space	Optimal value
Number of filters (f)	16 ; 32; 64; 128	128
number of batch b	2; 4; 8; 16; 62; 64; 128	8
kernel size (k)	3; 5	5
Dropout rate	0; 0.1; 0.2; 0.3; 0.4	0.2
number of hidden nodes (h)	5; 10; 50; 100; 500	10
types of the optimizer (o)	RMSprop; Adagrad; Adadelta; Adam; Adamax; Nadam	Adamax

The section provided illustrates how a CNN model is trained using a dataset. The dataset has been split into two sets at random: a training set that makes up 70% of the dataset and a test/validation set that makes up 30%. Input parameters like CNN layers and training choices are defined throughout the training process. A few of these are choosing an optimizer, deciding on the number of iterations, and deciding on the mini-batch size. The CNN model gradually picks up valuable traits by improving accuracy and decreasing loss. It classifies images and determines building classes.

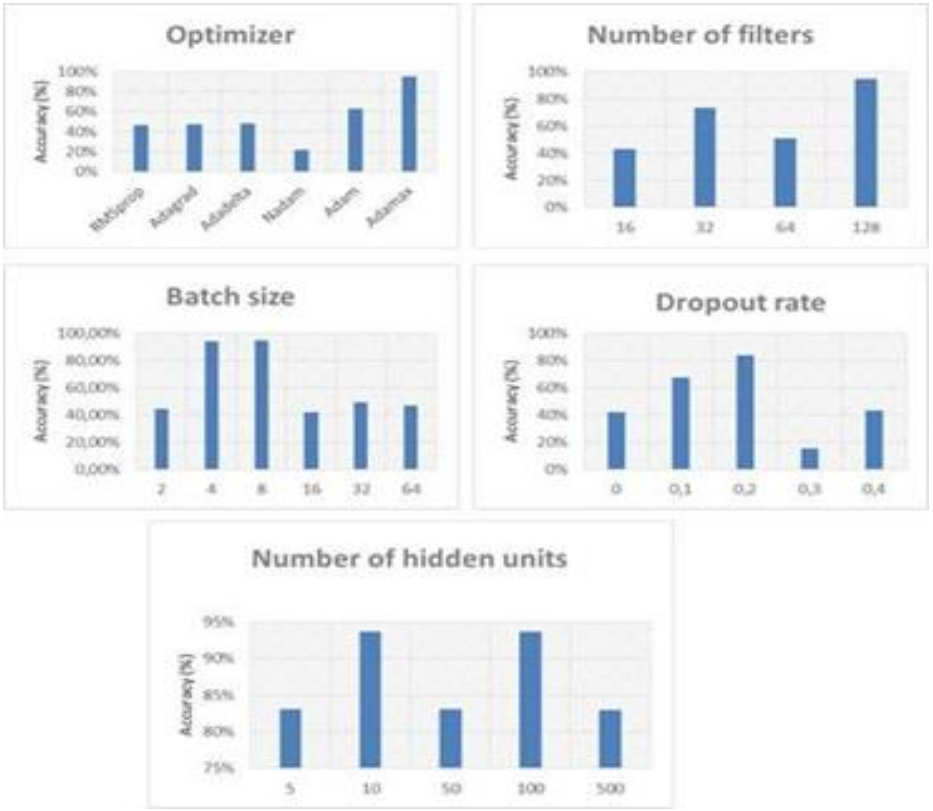


Figure 5.12: CNN hyperparameters' value's effect on efficiency

According to the results in Figure 5.12, the Adamax optimizer has the highest accuracy (94.98%), proving to be the best to utilize. In addition, the CNN model performed best with 128, achieving an accuracy of 94.83%, while the CNN model performed worst with 16, achieving an accuracy of 22.97%. According to a sensitivity study, the model's ideal batch size is 8, and its accuracy is 94.74%. Notably, achieving an accuracy of 94.05% with a batch size of 4 was comparable. Nevertheless, accuracy was significantly decreased (by around 50%) when batch sizes greater than eight were used.

Furthermore, the dropout rate directly impacts the accuracy of building recognition. A dropout rate of 0.2 was found to be ideal, with a 93.78% accuracy rate. However, using fewer units in the fully linked layer is thought to improve computation performance,

making ten units the ideal value for this parameter. The accuracy was best when utilizing 10 and 100 units at 83.77%.

5.3.3.2. Building Detection Results Comparison

We evaluated the efficacy and efficiency of our suggested method using the algorithms Res2- Unet (Chen *et al.*, 2022) and Slic-CNN. The evaluation images intentionally included buildings of various forms, sizes, and color combinations to increase the task's difficulty. The first row of Figure 5.13 displays the original input images utilized for assessment. These images represent two unique images from every set of the WHU aerial imagery 2016, New Zealand aerial image of Masterton, and Touggourt Google Earth image datasets. Figure 5.12 shows the final building extraction results of the involved methods.

Figure 5.13 illustrates that the other algorithms struggle to recognize buildings with ill-defined borders, resulting in numerous missed detections. The Res2-Unet strategy focuses on creating border rectification; however, it is impractical since it requires too much previous segmentation. In contrast, Our suggested method has many competitive benefits, notably in terms of applicability and stability when identifying buildings with different attributes. It effectively reduces boundary mistakes and precisely preserves the general architectural structure.

The performance of each strategy in the following experiment was assessed using five metrics: precision, recall, F1 score, false-negative rate (FNR), and authenticity of detection (AUT).

$$P = \frac{TP}{TP + FP} \tag{5.19}$$

$$R = \frac{TP}{TP + FN} \tag{5.20}$$

$$F1 = 2 * \frac{P * R}{P + R} \tag{5.21}$$

$$FNR = \frac{FN}{TP + FN} \tag{5.22}$$

False positives (FP) are the number of wrongly detected buildings, while TP (true positive) is the number of successfully detected buildings. The number of undiscovered buildings is a false negative (FN). In contrast to recall, which is the percentage of adequately predicted buildings based on the ground truth, precision is defined as the proportion of correctly predicted buildings among all the predicted buildings. For a clearer sense of how well our model is functioning, consider the F1 score, which combines precision and recall. The degree of missed detection to the total number of genuine buildings is also measured by the false-negative rate (FNR). Last but not least, the authenticity of detection (AUT) is a ratio of the accurately recognized buildings matching their shape in the ground truth.

The results presented in Table 5.4 reflect the improvement in score metrics that the suggested strategy produced. Compared to previous algorithms with greater false-positive and false-negative rates, it improves precision and recall while retaining a reduced false-negative ratio. These results validate VAE's better feature learning capabilities for building detection. Furthermore, as we depended on employing superpixels to establish the forms of the buildings precisely, the authenticity of detection was shown to improve the greatest, even with complicated backdrops and different scales, sizes, and shapes of buildings. Our suggested technique may have certain limits in certain unusual instances, as we discovered after carefully evaluating the findings. For instance:

- On sidewalks or roads with characteristics similar to the buildings, they are close to them when shadows from other nearby objects fall on them.
- Another problem arises when a group of tiny, closely spaced buildings is regarded as one building.
- When trees obscure portions of buildings or when shadows divide a single building into two.

Additionally, it is crucial to remember that utilizing Google Earth satellite imagery for building detection may have restrictions, such as image resolution and quality. Furthermore, the tropical desert climate and flat roof construction in the chosen study area cause sand to collect on rooftops, making distinguishing between the buildings more difficult. This explains why the findings were so poor compared to the other two datasets.

However, compared to other approaches, the outcomes produced by our suggested methodology are much better.

In conclusion, it is evident from the testing that the recommended building detection technique performs quite well and consistently across various challenging test imagery.

Table 5.4: Evaluation of the three techniques utilizing Precision(%), Recall(%), F1-Score(%), False-Negative Rate (FNR) (%), and the Authenticity of Detection (AUT) (%).

Methods	WHU aerial imagery 2016 dataset					New Zealand aerial image of Masterton dataset					GOOGLE Earth				
	P	R	F1	FNR	AUT	P	R	F1	FNR	AUT	P	R	F1	FNR	AUT
Res2-Unet (Chen *et al.*, 2022)	95,83	95,13	95,48	4,87	90,79	96,57	95,17	95,86	4,83	91,12	77,32	81,96	79,57	18,04	62,81
Slic-CNN	92,14	94,16	93,14	5,84	91,73	92,89	94,69	93,78	5,31	92,35	69,11	71,89	70,47	28,11	44,17
SP_VAE-CNN	96,74	97,57	97,15	2,43	94,76	97,12	97,58	97,35	2,42	95,82	85,88	87,95	86,90	12,05	83,37

Figure 5.13: Optical evaluation of aerial images dataset segmentation. Column 1 designates the original image, column 2 is the outcome of Res2-Unet (Chen *et al.*, 2022)*; column 3 is the outcome of SLIC-CNN; column 4 is the outcome of our proposed method SP_VAE-CNN, (a,b) are examples from WHU images 2016 dataset, (c,d) are examples from New Zealand imagery of Masterton; (e,f) are examples from high-resolution Google Earth images.*

5.4 Comparative analysis between the two approaches

The comparative analysis of building detection approaches employing machine learning SVM and deep learning methodologies provides valuable insights into their strengths and limitations. The SVM-based model, rooted in traditional machine learning, demonstrates commendable interpretability and robust performance, particularly in scenarios with limited data. Its reliance on handcrafted features and a well-defined decision boundary contributes to its adaptability and efficiency.

On the other hand, the deep learning approach, explicitly utilizing the Variational Autoencoder (VAE) and the Convolutional Neural Network (CNN), showcases unparalleled performance in complex and diverse urban landscapes. The ability of the deep learning model to automatically learn hierarchical features from raw data proves advantageous, especially when dealing with intricate patterns and variations within building structures. However, the inherent complexity of deep learning models comes at the cost of increased computational requirements.

The study further reveals that the deep learning model's performance benefits significantly from larger datasets, demonstrating its capacity to generalize well in data-rich environments. In contrast, the SVM model maintains its effectiveness with smaller datasets, making it a viable option when resources are constrained.

In practical terms, specific application requirements and available resources should inform the choice between these approaches. The interpretability and efficiency of the SVM model make it suitable for scenarios with limited data or when transparent decision-making is crucial. Conversely, the deep learning model emerges as a robust solution for applications demanding high accuracy in diverse and data-rich environments.

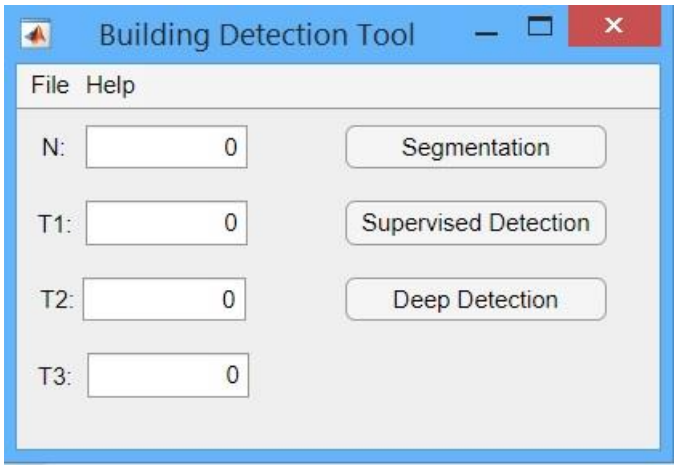
Ultimately, this comparative analysis contributes to the broader discourse on building detection methodologies, providing researchers and practitioners with valuable insights to guide the selection of the most appropriate approach based on the unique characteristics of the task at hand.

Table 5.5: Advantages and Disadvantages of each approach

	Advantages	Disadvantages
Supervised machine learning	- Speed in performance and calculation. - Does not need high-performance equipment. - It can be adapted to address specific problems	- Accuracy and identification ratio are less than desirable. - Does not give adequate results in highly complex areas.
Deep learning	- Takes a long time to process. - Needs high-performance equipment. - Cannot be adapted because it operates without supervision.	- High accuracy and identification ratio. - Gives good results even in highly complex areas

5.5 Building detection tool

We have created a simple tool interface using MATLAB r2020a 9.8.0.1323502 with an aerial or satellite image as an input. The output is a multiple files format (txt, tiff) of the building detection result (boundaries and total area of buildings) that most GIS software supports, as shown in Figure 5.13. Moreover, the tool has two phases: the first is for testing the selected segmentation parameters for a suitable choice Figure 5.14. Next is the detection stage, which begins with selecting the training sets for the classification Figure 5.15. The tool uses the same function sequence scenario as explained previously.



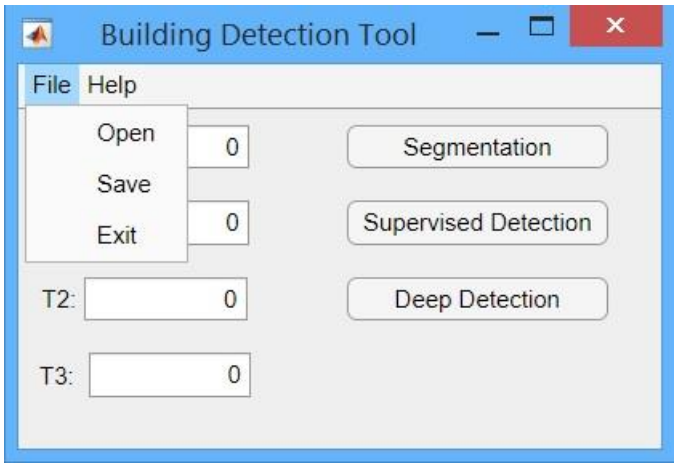


Figure 5.14: Building detection tool interface (n, t1, t2, t3 are the segmentation parameters).

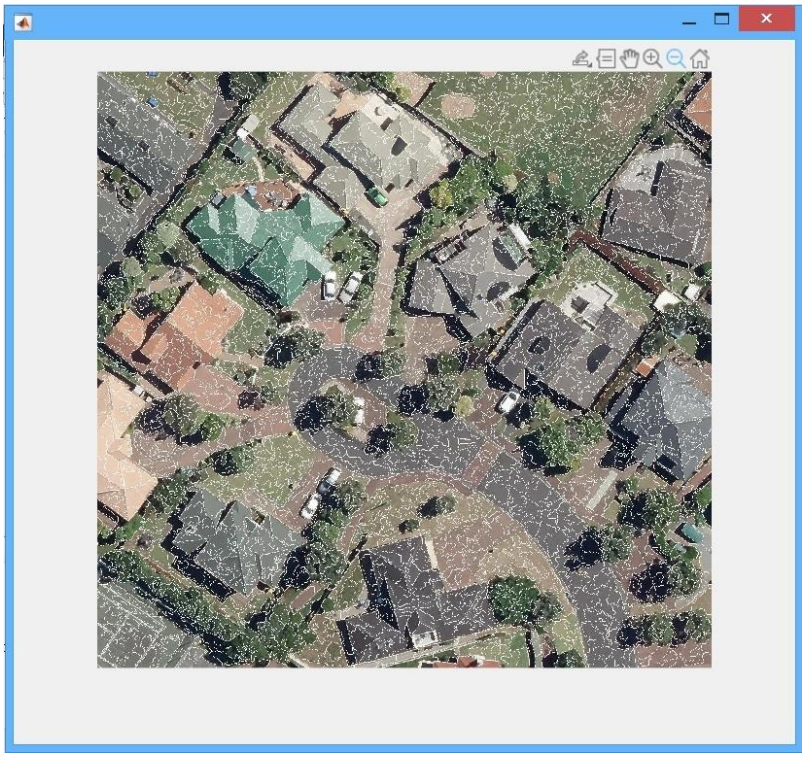
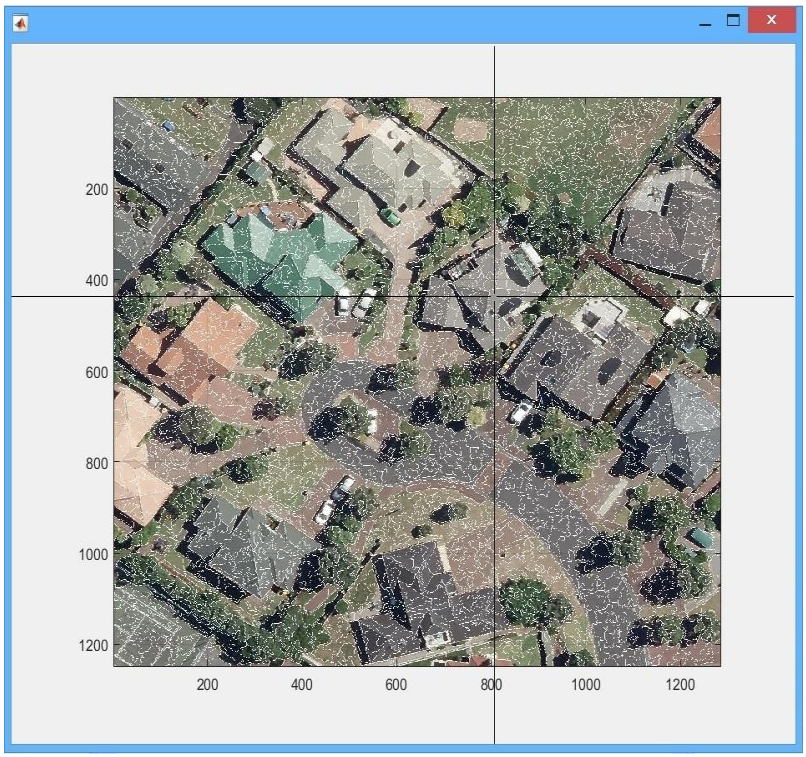
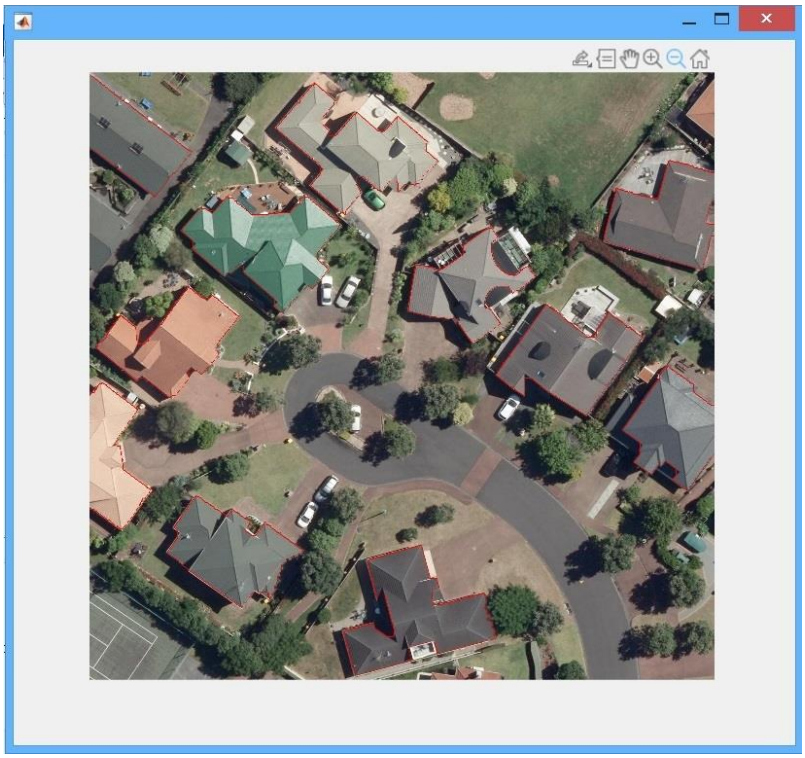


Figure 5.15: Segmentation test result.



(a)



(b)

Figure 5.16 : (a) Selecting training sets. (b) Building detection result.

5.5 Conclusion

Almost every building detection method has some limitations due to the restrictions that have been applied. This chapter presents a pipeline for building detection in 2D urban images using two different approaches. By combining the superpixel segmentation and machine learning methods, our approaches efficiently reduced the computation time while maintaining high precision.

The experimental findings showed a significant improvement after a thorough study. The range detection process was rapid and accurate, and it also used less processing time to detect the region than the previous approaches. The trustworthy methodology utilized in this work's automatic detection techniques might be applied in the case of a significant catastrophe.

General Conclusion and Perspectives

In this study, we focus on object identification, which can be considered the main part of developing land use. We briefly explain GIS, remote sensing, and its advancements over time. Then, we introduce a brief background about various image processing techniques, the machine learning field, and how we can utilize them to our advantage to create and design a building detection system that will be used in GIS for applications involving urban change.

Moreover, we have presented novel approaches for building detection based on prior knowledge of shadow position using Segmentation and machine learning methods. The current approaches strongly emphasize speed and minimal computation, and we compared them against other standard methods in terms of time consumption. Results proved its rapidity and low resource consumption, making it suitable for other small entities.

The thesis is a contribution to the field of Remote sensing, specifically automatic building detection. The building detection is improved by understanding the image contents. Segmentation of images using proper techniques leads to better region finding and increases performance. Working on a superpixel instead of a simple pixel increases the possibility of features that have been extracted from the image. This appears in the development of detection results that the accuracy enhancement has proved.

Nevertheless, a few restrictions were occasionally seen. Despite reaching good performance, further research is needed to understand our suggested techniques fully. These include examining the generalizability of more extensive and varied datasets, evaluating computational effectiveness, examining parameter sensitivity, doing robustness tests, and researching ways to include semantic data for better segmentation outcomes.

These studies will improve the validity and usefulness of our methods. Moreover, we want to expand our research to include partially or entirely hidden buildings in the images.

Bibliography

Aamir, M. *et al.* (2019) 'A framework for automatic building detection from low-contrast satellite images', *Symmetry*, 11(1), pp. 1–19. doi: 10.3390/sym11010003.

Achanta, R. *et al.* (2012) 'SLIC Superpixels Compared to State-of-the-Art Superpixel Methods', *IEEE Trans. on Pat. Anal. and Mach. Intel.*, 34(1), pp. 1–8.

Acharya, T. and Ray, A. K. (2005) *Image Processing: Principles and Applications*, *Image Processing: Principles and Applications*. doi: 10.1002/0471745790.

Adnan, M. M. *et al.* (2019) 'A Survey Automatic Image Annotation Based on Machine Learning Models', *journal of engineering and applied Sciences*, 14(20), pp. 7627–7635. doi: 10.36478/jeasci.2019.7627.7635.

Aksenov, A. L. and Kozlov, O. I. (2021) 'Satellite and aerial imagery geo-referencing using ground features', *Geodezia i Kartografia*, 975(9). doi: 10.22389/0016-7126-2021-975-9-21-29.

Ali, E. (no date) 'Geographic Information System (GIS): Definition , Development , Applications & Components', (79).

Attneave, F., B., M. and Hebb, D. O. (1950) 'The Organization of Behavior: A Neuropsychological Theory', *The American Journal of Psychology*, 63(4), p. 633. doi: 10.2307/1418888.

Awrangjeb, M., Ravanbakhsh, M. and Fraser, C. S. (2010) 'Automatic detection of residential buildings using LIDAR data and multispectral imagery', *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5), pp. 457–467. doi: 10.1016/j.isprsjprs.2010.06.001.

Benchabana, A. *et al.* (2022) 'A Supervised Building Detection Based on Shadow using Segmentation and Texture in High-Resolution Images', *Advances in Science, Technology and Engineering Systems Journal*, 7(3), pp. 167–174. doi: 10.25046/aj070319.

Benchabana, A. *et al.* (2023) 'Building Detection in High-Resolution Remote Sensing Images by Enhancing Superpixel Segmentation and Classification Using Deep Learning Approaches', *Buildings*, 13(7). doi: 10.3390/buildings13071649.

Bengio, Y. (2009) 'Learning Deep Architectures for AI', *Foundations and Trends® in Machine Learning*, 2(1), pp. 1–127. doi: 10.1561/2200000006.

Bensaci, R. *et al.* (2021) 'Deep convolutional neural network with knn regression for automatic image annotation', *Applied Sciences (Switzerland)*, 11(21), pp. 1–20. doi: 10.3390/app112110176.

Benz, U. C. *et al.* (2004) 'Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information', *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3–4), pp. 239–258. doi: 10.1016/j.isprsjprs.2003.10.002.

Bertacchi, M. G. and Silveira, I. F. (2019) 'Facial Makeup Detection using the CMYK Color Model and Convolutional Neural Networks', in *Proceedings - 15th Workshop of Computer Vision, WVC 2019*. doi: 10.1109/WVC.2019.8876943.

Blakemore, M. and Chorley, Lord (1988) 'Handling Geographic Information: Report to the Secretary of State for the Environment of the Committee of Enquiry', *Transactions of the Institute of British Geographers*, 13(1). doi: 10.2307/622781.

Bobbia, S. *et al.* (2021) 'Iterative Boundaries Implicit Identification for Superpixels Segmentation: A Real-Time Approach', *IEEE Access*, 9. doi: 10.1109/ACCESS.2021.3081919.

Bobrowsky, P. T. and Marker, B. (eds) (2018) *Encyclopedia of Engineering Geology*. Cham: Springer International Publishing (Encyclopedia of Earth Sciences Series). doi: 10.1007/978-3-319-73568-9.

'Book Review' (1991) *Journal of the American Planning Association*, 57(2). doi: 10.1080/01944369108975494.

C., G. R. and E., W. R. (2002) 'Digital image processing'.

Camastra, F. (2007) 'Image Processing: Principles and Applications [book review]', *IEEE Transactions on Neural Networks*, 18(2). doi: 10.1109/tnn.2007.893088.

Cao, Y. and Huang, X. (2023) 'A full-level fused cross-task transfer learning method for building change detection using noise-robust pretrained networks on crowdsourced labels', *Remote Sensing of Environment*, 284, p. 113371. doi: 10.1016/j.rse.2022.113371.

Carbonell, J. G. (1981) 'Machine learning research', *ACM SIGART Bulletin*, 18(77), pp. 29–29. doi: 10.1145/1056743.1056744.

Carson, C. *et al.* (1999) 'Blobworld: A system for region-based image indexing and retrieval', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1614, pp. 509–517. doi: 10.1007/3-540-48762-x_63.

Chen, D., Shang, S. and Wu, C. (2014) 'Shadow-based Building Detection and Segmentation in High-resolution Remote Sensing Image', *Journal of Multimedia*, 9(1), pp. 181–188. doi: 10.4304/jmm.9.1.181-188.

Chen, F. *et al.* (2022) 'Res2-Unet, a New Deep Architecture for Building Detection from High Spatial Resolution Images', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, pp. 1494–1501. doi: 10.1109/JSTARS.2022.3146430.

Chen, S. *et al.* (2023) 'Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach', *ISPRS Journal of Photogrammetry and Remote Sensing*, 195, pp. 129–152. doi: 10.1016/j.isprsjprs.2022.11.006.

Cheng, G. *et al.* (2020) 'Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13. doi: 10.1109/JSTARS.2020.3005403.

Chuvieco, E. and Huete, A. (2009) *Fundamentals of satellite remote sensing*, *Fundamentals of Satellite Remote Sensing*. doi: 10.1201/b18954.

Cohen, J. P. *et al.* (2016) 'Rapid building detection using machine learning', *Applied Intelligence*, 45(2), pp. 443–457. doi: 10.1007/s10489-016-0762-6.

Cretu, A. M. and Payeur, P. (2013) 'Building detection in aerial images based on watershed and visual attention feature descriptors', *Proceedings - 2013 International Conference on Computer and Robot Vision, CRV 2013*, pp. 265–272. doi: 10.1109/CRV.2013.8.

Deng, X. *et al.* (2022) 'Towards optimal HVAC control in non-stationary building environments

combining active change detection and deep reinforcement learning', *Building and Environment*, 211, p. 108680. doi: 10.1016/j.buildenv.2021.108680.

Dhanachandra, N., Manglem, K. and Chanu, Y. J. (2015) 'Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm', *Procedia Computer Science*, 54, pp. 764–771. doi: 10.1016/j.procs.2015.06.090.

Dietterich, T. G. (2002) 'Machine learning for sequential data: A review', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2396, pp. 15–30. doi: 10.1007/3-540-70659-3_2.

Dietterich, T. G. and Oregon (1996) 'Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Models', *Oncogene*, 12(2), p. pp 1-15(265-275).

Doggaz, N. and Ferjani, I. (2011) 'Image segmentation using normalized cuts and efficient graph-based segmentation', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6979 LNCS(PART 2), pp. 229–240. doi: 10.1007/978-3-642-24088-1_24.

Duchi, J., Hazan, E. and Singer, Y. (2012) 'Adaptive subgradient methods for online learning and stochastic optimization', *Proceedings of the IEEE Conference on Decision and Control*, 12, pp. 5442–5444. doi: 10.1109/CDC.2012.6426698.

Dutta, A. (2019) *Blending the Past and Present of Automatic Image Annotation*.

F., L., H., Z. and D., F. D. (2003) 'Fundamentals of content-based image retrieval', *Multimedia Information Retrieval and Management*, pp. 1–26.

Frstner, W. (1994) 'A framework for low level feature extraction', *Computer*.

Ghandour, A. and Jezzini, A. (2018a) 'Autonomous Building Detection Using Edge Properties and Image Color Invariants', *Buildings*, 8(5), p. 65. doi: 10.3390/buildings8050065.

Ghandour, A. and Jezzini, A. (2018b) 'Post-War Building Damage Detection', *Proceedings*, 2(7), p. 359. doi: 10.3390/ecrs-2-05172.

Ghanea, M., Moallem, P. and Momeni, M. (2014) 'Automatic building extraction in dense urban areas through GeoEye multispectral imagery', *International Journal of Remote Sensing*, 35(13), pp. 5094–5119. doi: 10.1080/01431161.2014.933278.

Grigillo, D., Kosmatin Fras, M. and Petrovič, D. (2011) 'Automatic extraction and building change detection from digital surface model and multispectral orthophoto', *Geodetski vestnik*, 55(01), pp. 011–027. doi: 10.15292/geodetski-vestnik.2011.01.011-027.

Guo, Z. *et al.* (2017) 'Village building identification based on Ensemble Convolutional Neural Networks', *Sensors (Switzerland)*, 17(11), pp. 1–22. doi: 10.3390/s17112487.

Guru Prathap Reddy, V. *et al.* (2024) 'ND Evaluation of Chemically Induced Deterioration in Concrete: A Colour Spaces Study', *Arabian Journal for Science and Engineering*. doi: 10.1007/s13369-023-08605-y.

Hassan, H. *et al.* (2021) 'Real-time image dehazing by superpixels segmentation and guidance filter', in *Journal of Real-Time Image Processing*. doi: 10.1007/s11554-020-00953-4.

He, X. J. *et al.* (2006) 'A new feature of uniformity of image texture directions coinciding with the human eyes perception', *Lecture Notes in Computer Science (including subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics), 3614 LNAI, pp. 727–730. doi: 10.1007/11540007_90.

Hinton, G. E., Osindero, S. and Teh, Y.-W. (2006) 'A Fast Learning Algorithm for Deep Belief Nets', *Neural Computation*, 18(7), pp. 1527–1554. doi: 10.1162/neco.2006.18.7.1527.

Hinton, G. E. and Salakhutdinov, R. R. (2006) 'Reducing the Dimensionality of Data with Neural Networks', *Science*, 313(5786), pp. 504–507. doi: 10.1126/science.1127647.

De Hoog, J. *et al.* (2020) 'Using Satellite and Aerial Imagery for Identification of Solar PV: State of the Art and Research Opportunities', in *e-Energy 2020 - Proceedings of the 11th ACM International Conference on Future Energy Systems*. doi: 10.1145/3396851.3397681.

Hou, X. *et al.* (2021) 'High-resolution triplet network with dynamic multiscale feature for change detection on satellite images', *ISPRS Journal of Photogrammetry and Remote Sensing*, 177(May), pp. 103–115. doi: 10.1016/j.isprsjprs.2021.05.001.

Iqbal, J. and Ali, M. (2020) 'Weakly-supervised domain adaptation for built-up region segmentation in aerial and satellite imagery', *ISPRS Journal of Photogrammetry and Remote Sensing*, 167. doi: 10.1016/j.isprsjprs.2020.07.001.

Islam, M. M., Zhang, D. and Lu, G. (2008) 'A geometric method to compute directionality features for texture images', *2008 IEEE International Conference on Multimedia and Expo, ICME 2008 - Proceedings*, (3), pp. 1521–1524. doi: 10.1109/ICME.2008.4607736.

Jaiswal, S. and Pandey, M. K. (2021) 'A Review on Image Segmentation', *Advances in Intelligent Systems and Computing*, 1187, pp. 233–240. doi: 10.1007/978-981-15-6014-9_27.

Ji, S., Wei, S. and Lu, M. (2019) 'Fully Convolutional Networks for Multisource Building Extraction From an Open Aerial and Satellite Imagery Data Set', *IEEE Transactions on Geoscience and Remote Sensing*, 57(1), pp. 574–586. doi: 10.1109/TGRS.2018.2858817.

Jiebo Luo and Savakis, A. (2002) 'Indoor vs outdoor classification of consumer photographs using low-level and semantic features', in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*. IEEE, pp. 745–748. doi: 10.1109/ICIP.2001.958601.

Joblove, G. H. and Greenberg, D. (1978) 'Color spaces for computer graphics', in *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1978*. doi: 10.1145/800248.807362.

Jones, M. A. A. J. (2003) *Conceptualizing Space: Mapping Schemas as Meaningful Representations*. doi: http://dx.doi.org/10.13140/2.1.3030.1767.

Jun, T. (2010) 'A color image segmentation algorithm based on region growing', *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, 6, pp. 634–637. doi: 10.1109/ICCET.2010.5486012.

Kang, J. *et al.* (2018) 'Building instance classification using street view images', *ISPRS Journal of Photogrammetry and Remote Sensing*, 145, pp. 44–59. doi: 10.1016/j.isprsjprs.2018.02.006.

Kang, W. X., Yang, Q. Q. and Liang, R. P. (2009) 'The comparative research on image segmentation algorithms', *Proceedings of the 1st International Workshop on Education Technology and Computer Science, ETCS 2009*, 2, pp. 703–707. doi: 10.1109/ETCS.2009.417.

Karantzalos, K. and Paragios, N. (2008) 'Automatic model-based building detection from single

panchromatic high resolution images', *International Archives of the Photogrammetry. Remote Sensing & Spatial Information Sciences*, 37, pp. 225–230. Available at: http://www.isprs.org/proceedings/XXXVII/congress/3_pdf/19.pdf.

Khan, A. *et al.* (2022) 'A Novel Threshold-Based Segmentation Method for Quantification of COVID-19 Lung Abnormalities', *Signal, Image and Video Processing*. doi: 10.1007/s11760-022-02183-6.

Khan, S. D., Alarabi, L. and Basalamah, S. (2023) 'An Encoder–Decoder Deep Learning Framework for Building Footprints Extraction from Aerial Imagery', *Arabian Journal for Science and Engineering*, 48(2), pp. 1273–1284. doi: 10.1007/s13369-022-06768-8.

Kingma, D. P. and Ba, J. L. (2015) 'Adam: A method for stochastic optimization', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15.

Kingma, D. P. and Welling, M. (2013) 'Auto-Encoding Variational Bayes'. doi: 10.48550/arxiv.1312.6114.

Kohli, D., Sliuzas, R. and Stein, A. (2016) 'Urban slum detection using texture and spatial metrics derived from satellite imagery', *Journal of Spatial Science*, 61(2), pp. 405–426. doi: 10.1080/14498596.2016.1138247.

Kokila, S. and Jayachandran, A. (2023) 'Bias variance Toeplitz Matrix based Shift Invariance classifier for building detection from satellite images', *Remote Sensing Applications: Society and Environment*, 29, p. 100881. doi: 10.1016/j.rsase.2022.100881.

Kusz, M. *et al.* (2021) 'Building Detection with Deep Learning', *ACM International Conference Proceeding Series*. doi: 10.1145/3437359.3465573.

Labrecque, L. I. (2020) 'Color research in marketing: Theoretical and technical considerations for conducting rigorous and impactful color research', *Psychology & Marketing*, 37(7), pp. 855–863. doi: 10.1002/mar.21359.

Lei, G. T., Techentin, R. W. and Gilbert, B. K. (1999) 'High-frequency characterization of power/ground-plane structures', *IEEE Transactions on Microwave Theory and Techniques*, 47(5). doi: 10.1109/22.763156.

Li, S. and Shen, Q. (2006) 'Page segmentation using morphological operations', *Advances in Modelling and Analysis B*. doi: 10.5120/20654-3197.

Likas, A., Vlassis, N. and Verbeek, J. (2011) 'The global k-means clustering algorithm Intelligent Autonomous Systems', *ISA technical report series*.

Lions, P. L. *et al.* (1995) 'Texture Segmentation Using Fractal Dimension', 17(1), pp. 72–77.

Liu, F. and Picard, R. W. (1996) 'Periodicity , directionality , and randomness : Wold features for image modeling and retrieval', 18(320), pp. 722–733.

Lomax, S. and Vadera, S. (2013) 'A survey of cost-sensitive decision tree induction algorithms', *ACM Computing Surveys*, 45(2). doi: 10.1145/2431211.2431215.

Lv, X. *et al.* (2019) 'Very high resolution remote sensing image classification with SEEDS-CNN and scale effect analysis for superpixel CNN classification', *International Journal of Remote Sensing*, 40(2), pp. 506–531. doi: 10.1080/01431161.2018.1513666.

Lv, Z. *et al.* (2016) 'Novel object-based filter for improving land-cover classification of aerial

imagery with very high spatial resolution', *Remote Sensing*, 8(12). doi: 10.3390/rs8121023.

Ma, J. *et al.* (2020) 'Building extraction of aerial images by a global and multi-scale encoder-decoder network', *Remote Sensing*, 12(15). doi: 10.3390/RS12152350.

Maini, R. and Aggarwal, H. (2009) 'Study and Comparison of Various Image Edge Detection Techniques', in *International Journal of Image Processing*, pp. 1–12.

Malik, J. and Perona, P. (1990) 'Preattentive texture discrimination with early vision mechanisms', *Journal of the Optical Society of America A*, 7(5), p. 923. doi: 10.1364/JOSAA.7.000923.

Manandhar, P., Aung, Z. and Marpu, P. R. (2017) 'Segmentation based building detection in high resolution satellite images', *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017-July, pp. 3783–3786. doi: 10.1109/IGARSS.2017.8127823.

Mao, B., Li, B. and Sun, J. (2019) 'Large area building detection from airborne lidar data using OSM trained superpixel classification', in *Proceedings - 2019 7th International Conference on Advanced Cloud and Big Data, CBD 2019*. Institute of Electrical and Electronics Engineers Inc., pp. 145–150. doi: 10.1109/CBD.2019.00035.

Mierzejowska, A. and Pomykoł, M. (2019) 'Calibration of raster image using GIS class software- A ccuracy analysis', in *IOP Conference Series: Earth and Environmental Science*. doi: 10.1088/1755-1315/261/1/012033.

Minaee, S. *et al.* (2021) 'Image Segmentation Using Deep Learning: A Survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8828(c), pp. 1–20. doi: 10.1109/TPAMI.2021.3059968.

Moutarde, F. (2019) 'IA : vers des robots intelligents ?', (October 2018).

Nayak, S. R., Padhy, R. and Mishra, J. (2017) 'Texture analysis methods: A review', *Journal of Advanced Research in Dynamical and Control Systems*, 9(11), pp. 46–52.

Nogueira, K., Penatti, O. A. B. and dos Santos, J. A. (2017) 'Towards better exploiting convolutional neural networks for remote sensing scene classification', *Pattern Recognition*, 61, pp. 539–556. doi: 10.1016/j.patcog.2016.07.001.

Nosrati, M. S. and Saeedi, P. (2009) 'A combined approach for building detection in satellite imageries using active contours', *Proceedings of the 2009 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2009*, 2, pp. 1012–1017.

Nurkarim, W. and Wijayanto, A. W. (2023) 'Building footprint extraction and counting on very high-resolution satellite imagery using object detection deep learning framework', *Earth Science Informatics*, 16(1), pp. 515–532. doi: 10.1007/s12145-022-00895-4.

Oja, E. (1982) 'Simplified neuron model as a principal component analyzer', *Journal of Mathematical Biology*, 15(3), pp. 267–273. doi: 10.1007/BF00275687.

Ojogbane, S. S. *et al.* (2021) 'Automated building detection from airborne LiDAR and very high-resolution aerial imagery with deep neural network', *Remote Sensing*, 13(23), pp. 1–16. doi: 10.3390/rs13234803.

Palanikumar, S., Albert Jerome, S. and Jayan, J. P. (2022) 'Automatic detection of solitary pulmonary nodules using superpixels segmentation based iterative clustering approach', *Journal of Ambient Intelligence and Humanized Computing*, 13(6). doi: 10.1007/s12652-021-03148-2.

Pham, N. A. *et al.* (2007) 'Quantitative image analysis of immunohistochemical stains using a CMYK color model', *Diagnostic Pathology*, 2(1). doi: 10.1186/1746-1596-2-8.

Quinlan, J. R. (1986) 'Induction of decision trees', *Machine Learning*, 1(1), pp. 81–106. doi: 10.1007/bf00116251.

Quinlan, J. R. (1996) 'Learning decision tree classifiers', *ACM Computing Surveys*, 28(1), pp. 71–72. doi: 10.1145/234313.234346.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (2013) 'Learning Internal Representations by Error Propagation', *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, (V), pp. 399–421. doi: 10.1016/B978-1-4832-1446-7.50035-2.

Salehi, B. *et al.* (2012) 'Object-based classification of urban areas using VHR imagery and height points ancillary data', *Remote Sensing*, 4(8), pp. 2256–2276. doi: 10.3390/rs4082256.

Shen, X. *et al.* (2019) 'Inundation extent mapping by synthetic aperture radar: A review', *Remote Sensing*, 11(7), pp. 1–17. doi: 10.3390/RS11070879.

Shorter, N. and Kasparis, T. (2009) 'Automatic vegetation identification and building detection from a single nadir aerial image', *Remote Sensing*, 1(4), pp. 731–757. doi: 10.3390/rs1040731.

Sirko, W. *et al.* (2021) 'Continental-Scale Building Detection from High Resolution Satellite Imagery', pp. 1–15. Available at: http://arxiv.org/abs/2107.12283.

Sivaraj, S., Malmathanraj, R. and Palanisamy, P. (2020) 'Detecting anomalous growth of skin lesion using threshold-based segmentation algorithm and Fuzzy K-Nearest Neighbor classifier', *Journal of Cancer Research and Therapeutics*, 16(1). doi: 10.4103/jcrt.JCRT_306_17.

Song, D. *et al.* (2020) 'Integration of super-pixel segmentation and deep-learning methods for evaluating earthquake-damaged buildings using single-phase remote sensing imagery', *International Journal of Remote Sensing*, 41(3), pp. 1040–1066. doi: 10.1080/01431161.2019.1655175.

Stankov, K. and He, D.-C. (2013) 'Using the Spectral Similarity Ratio and Morphological Operators for the Detection of Building Locations in Very High Spatial Resolution Images', *Journal of Communication and Computer*, 10(March 2013), pp. 309–324. Available at: http://www.researchgate.net/profile/Katia_Stankov/publication/261403266_Using_the_spectral_similarity_ratio_and_morphological_operators_for_the_detection_of_building_locations_in_very_high_spatial_resolution_images/links/0deec5343310486190000000.pdf.

Suárez, J. C. *et al.* (2005) 'Use of airborne LiDAR and aerial photography in the estimation of individual tree heights in forestry', *Computers and Geosciences*, 31(2). doi: 10.1016/j.cageo.2004.09.015.

Sun, Z. *et al.* (2021) 'Sar image classification using fully connected conditional random fields combined with deep learning and superpixel boundary constraint', *Remote Sensing*, 13(2), pp. 1–27. doi: 10.3390/rs13020271.

Tomlinson, R. F. (1969) 'A Geographic Information System for Regional Planning', *Journal of Geography (Chigaku Zasshi)*, 78(1). doi: 10.5026/jgeography.78.45.

Tremeau, A. and Borel, N. (1997) 'A region growing and merging algorithm to color segmentation', *Pattern Recognition*, 30(7), pp. 1191–1203. doi: 10.1016/S0031-3203(96)00147-1.

Tuceryan, M. and Jain, A. K. (1993) 'texture analysis', *Handbook of Pattern Recognition and Computer Vision*. doi: 10.1142/9789814343138_0010.

Ullo, S. L. *et al.* (2020) 'Lidar-based system and optical vhr data for building detection and mapping', *Sensors (Switzerland)*, 20(5), pp. 1–23. doi: 10.3390/s20051285.

Vadivel, A., Sural, S. and Majumdar, A. K. (2007) 'An Integrated Color and Intensity Co-occurrence Matrix', *Pattern Recognition Letters*, 28(8), pp. 974–983. doi: 10.1016/j.patrec.2007.01.004.

Wang, J. *et al.* (2015) 'An efficient approach for automatic rectangular building extraction from very high resolution optical satellite imagery', *IEEE Geoscience and Remote Sensing Letters*, 12(3), pp. 487–491. doi: 10.1109/LGRS.2014.2347332.

Wang, J. Z., Li, J. and Wiederhold, G. (2001) 'SIMPLIcity : Semantics-Sensitive Integrated Matching for Picture LIbraries', 23(9), pp. 947–963.

Wang, Z. (2020) 'A new clustering method based on morphological operations', *Expert Systems with Applications*, 145. doi: 10.1016/j.eswa.2019.113102.

Xiang, T.-Z., Xia, G.-S. and Zhang, L. (2018) 'Mini-Unmanned Aerial Vehicle-Based Remote Sensing: Techniques, Applications, and Prospects'. doi: 10.1109/MGRS.2019.2918840.

Zeiler, M. D. (2012) 'ADADELTA: An Adaptive Learning Rate Method'. Available at: http://arxiv.org/abs/1212.5701.

Zhang, A. *et al.* (2017) 'Building Detection from Satellite Images on a Global Scale', (Nips). Available at: http://arxiv.org/abs/1707.08952.

Zhang, L. *et al.* (2020) 'An Efficient Building Extraction Method from High Spatial Resolution Remote Sensing Images Based on Improved Mask R-CNN', *Sensors*, 20(5), p. 1465. doi: 10.3390/s20051465.

Zheng, H. *et al.* (2022) 'HFA-Net: High frequency attention siamese network for building change detection in VHR remote sensing images', *Pattern Recognition*, 129, p. 108717. doi: 10.1016/j.patcog.2022.108717.

Zhou, G. and Sha, H. (2020) 'Building shadow detection on ghost images', *Remote Sensing*, 12(4). doi: 10.3390/rs12040679.

Zhu, S. *et al.* (2015) 'Deep neural network based image annotation', *Pattern Recognition Letters*, 65, pp. 103–108. doi: 10.1016/j.patrec.2015.07.037.