**Extraction of Main Urban Roads from High Resolution Satellite Images by Machine Learning**

**ABSTRACT:**

Road extraction from high-resolution remote sens- ing images is a challenging but hot research topic in the past decades. A large number of methods are invented to deal with this problem. This article provides a comprehensive review of these existing approaches. We classified the methods into heuristic and data-driven. The heuristic methods are the mainstream in the early years, and the data-driven methods based on deep learning have been quickly developed recently. With regard to the heuristic methods, the road feature model is first introduced, then, the classic extraction methods are reviewed in two subcategories: semiau- tomatic and automatic. The principles, inspirations, advantages, and disadvantages of these methods are described. In terms of the data-driven methods, the road extraction methods based on deep neural network, particularly those based on patched convolutional neural network, fully convolutional network, and generative ad- versarial network are reviewed. We perform subjective compar- isons between the methods inner each type. Furthermore, the quantity performances achieved on the same dataset are com- pared between the heuristic and data-driven methods to show the strengthening of the data-driven methods. Finally, the conclusion and prospects are summarized.

**INTRODUCTION :**

In 1972, the first Earth Resources Observation Technol- ogy Satellite, later renamed Landsat, was launched by the United States. The interpretation technologies of remote sensing imagery have been developed rapidly, including image compression, transmission, segmentation, fusion, understand- ing, etc. With the use of modern sensors (e.g., IKONOS, Quick- Bird, and GeoEye), the spatial, spectral, and time resolutions of the RS images have gradually increased. HRSI provides a new way to obtain detailed geographic information, which consequently motivates further development of the processing technology of HRSI [10]. At present, Image recognition technologies have been suc- cessfully applied to many specific fields, such as fingerprint identification [11], face recognition [12], scene description [13], etc. However, there still have many problems with understanding RS (including aerial) images [8]. As man-made objects, roads are the important information processed in the GIS. Road infor- mation can be used in many aspects of social life, such as vehicle navigation, traffic management, map updating, and geological disaster emergency [14]. With the further application of GIS, the manual extraction methods cannot fulfill the quick updating requirements. Therefore, extracting road information from RS images by (or aided by) machine is in demand. We can easily collect a large number of articles about road ex- traction method from Web of Science or Google Scholar, but the classification of them is very difficult due to the various applied technologies. In accordance with distinct classification criteria, road extraction methods can be classified into various types. For example, based on the different algorithms, these methods can be categorized into clustering, classification, morphology, dynamic programming, active contour models, etc. [8]. According to the objects to be handled in the algorithms, these methods can be pixel based, region based, and knowledge based [5]. On the basis of the different outputs of these methods, they can be divided into road segmentation and road centerline extraction [15]. In this work, we divide the road extraction methods into two types: the heuristic methods and the data-driven methods. The heuristic methods are further divided into semiautomatic methods and automatic methods according to the degree of the manual intervention. The data-driven methods are subdivided into several types based on the architecture of neural network

**LITERATURE SURVEY:**

**Semi automatic road extraction from digital images**

**AUTHORS:**

**H. R. R. Bakhtiari, A. Abdollahi, and H. Rezaeian,**

**ABSTRACT:**

Road extraction from digital images is of fundamental importance in the context of automatic mapping, effective urban planning and updating GIS databases. Very high spatial resolution (VHR) imagery acquired by airborne and space borne sensors is the main source for accurate road extraction. Manual techniques are fading away as they are time consuming and costly. Hence, road extraction method that is significantly more automated has become a research hotspot in remote sensing information processing. This paper proposes a semi-automatic approach to extract different road types from high-resolution remote sensing images. The approach is based on edge detection and SVM and mathematical morphology method. First the outline of the road is detected based on Canny operator. Then, Full Lambda Schedule merging method combines adjacent segments. Then the entire image was classified using Support Vector Machine (SVM) and various spatial, spectral, and texture attributes to form a road image. Finally, the quality of detected roads is improved using morphological operators. The algorithm was systematically evaluated on a variety of satellite images from Worldview, QuickBird and UltraCam airborne Images. The results of the accuracy evaluation demonstrate that the proposed road extraction approach can provide high accuracy for extraction of different road types.

**EXISTING SYSTEM :**

Manuscript received June 17, 2019; revised December 23, 2019, April 7, 2020,and July 28, 2020; accepted September 1, 2020. Date of publication September11, 2020; date of current version September 25, 2020. This work was supportedin part by the Natural Science Foundation of China under Grant 61170147, in partby the Education and Scientific Research Project for Middle-Aged and YoungTeachers in Fujian Province under Grant JT180595, in part by the Key Program ofFostering Young Scientific Research Talents in Fujian Jiangxia University underGrant JXZ2018002 and Grant JXZ2016001, and in part by the Fuzhou Scienceand Technology Project under Grant 2020-G-066. (Corresponding authors:Weixing Wang; Liqin Huang.)Renbao Lian is with the College of Physics and Information Engineering,Fuzhou University, Fuzhou 350008, China, with the Digital Fujian, Internet-of-Things Key Lab of Information Collection and Processing in Smart Home,Fuzhou 350108, China, and also with the College of Electronics and Infor-mation Science, Fujian Jiangxia University, Fuzhou 350108, China (e-mail:luoshao@163.com

**EXISTING SYSTEM DISADVANTAGES:**

1.LESS ACCURACY

2. LOW EFFICIENCY

**PROPOSED SYSTEM :**

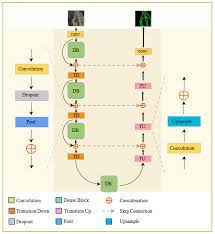
Establishing a road model can help us extract road more effectively. Baumgartner et al. [20] proposed a classic road model based on the composition of roads. The road model is divided into three layers, i.e., realistic road network layer, geographic geometric feature layer, and image feature layer. The model shows how the different of road materials and geometric shapes in the real world presented in the images. The model also demonstrates the road features from the perspective of high and low resolutions. More precise information can be extracted from RS image with higher resolution, such as road lanes and zebra crossings. However, higher resolution may introduce more interferences, which will disturb the extraction of global road networks. In the coarse scale, most interferences on road surfaces are eliminated, and prominent road edges are preserved to iden- tify road networks. However, the extracted roads are typically broken and imprecise given the lack of resolution. On this basis, the road extraction methods based on multiscale segmentation are extensively researched

**PROPOSED SYSTEM ADVANTAGES:**

1.HIGH ACCURACY

2.HIGH EFFICIENCY

**SYSTEM ARCHITECTURE:**



**RELATED WORK :**

The key issue is how to recognize the objects of interest in HRSI quickly and accurately [3], [4]. The basic processing of HRSI is to extract information for objects classification and recognition according to not only the spectral features but also the shape characters and spatial relations [5]. The bottleneck of these problems is the recognition of the small terrestrial objects like roads, junctions, buildings, etc. [6]. Man-made objects are most important elements in spatial geographic information databases, including buildings, bridges, roads, farmlands, etc. Buildings and roads are unquestionably the two most important object classes. A large number of articles have been published discussing the detection of buildings and roads during the last decades [7]. For example, we will get more than 2.6 million related papers if we search Google Scholar with the keyword “road extraction,” and more than 129 000 articles remain if the publish date is restricted to post 2016. Faced with such a large number of works of literature, a systematic review of road extraction algorithms is valuable for beginners. Although we have reviewed the traditional methods before [8], we stick to present a more comprehensive review from a new perspective given the rapid development of deep learning and the numerous articles discussing automatic road extraction employing deep learning models [9]. We divide the existing methods into two categories, namely, heuristic and data-driven. The heuristic methods extract road features using the prior of road regions, whereas the data-driven methods depend on big data. For heuristic methods, we supplemented some methods missing in the previous version and made a more concrete description of the basic principles of each method, which will be more friendly to beginners. In order to review the heuris- tic and data-driven methods systematically, we collected and sieved more 200 papers discussing road extraction from HRSI published in the past two decades. We only kept the pieces of literature that extract roads directly from optical images captured by satellites or aircraft instead of from the data generated by light detection and ranging or synthetic aperture radar. We restricted the articles to peer-reviewed ones and preferred those with high citations.

**PURPOSE :**

The road network is difficult to extraction from HRSI because the road features can be affected by different sensor type, distinct spectral and spatial resolution, volatile weather conditions, di- verse road material, complex backgrounds, etc. It is important to analyze the road features and road model in the normal situation without considering noise interference. Overall, roads in HRSI present ribbons with slow intensity change [17]. We summarize road features as follows. 1) Geometric Features: Roads are stripe-shaped objects whose width does not change dramatically. Its length is much longer than its width. The intersections of roads can be visually described as the shape of “+,” “Y,” or “T.” Besides, when the spatial resolution of RS images reduced, road ribbons may be degraded as linear geometric objects. 2) Spectral Features: Roads are characterized by the evident parallel edges. Gradient magnitudes are high on road edges but low inside roads. Strong intensity contrast exists between roads and the surroundings, but the contrast may be completely opposite due to the different road materials [18]. For example, cement roads generally present brighter than bitumen roads in panchromatic satellite images. 3) Topological Features: Road networks typically have in- tersections, and roads are not suddenly interrupted. In images, roads generally span across the entire scenes, unless the roads are dead ends. The road network can be viewed as a graph composed by vertices and edges

HARDWARE & SOFTWARE REQUIREMENTS:

**Non- Functional Requirements:**

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are> 10000. Description of non-functional requirements is just as critical as a functionalrequirement.

* Usabilityrequirement
* Serviceabilityrequirement
* Manageabilityrequirement
* Recoverabilityrequirement
* Securityrequirement
* Data Integrity requirement
* Capacityrequirement
* Availabilityrequirement
* Scalabilityrequirement
* Interoperabilityrequirement
* Reliabilityrequirement
* Maintainabilityrequirement
* Regulatoryrequirement
* Environmentalrequirement

**HARD REQUIRMENTS :**

* System    :   i3 or above.
* Ram    :   4 GB.
* Hard Disk : 40 GB

**SOFTWARE REQUIRMENTS :**

* Operating system   : Windows8 or Above.
* Coding Language  : python

# SYSTEM STUDY FEASIBILITY STUDY

# The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILI**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# 4.SYSTEM DESIGN :

**4.1 .UML DIAGRAMS :**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



# CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**COLLABRATION DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**MODEL :**

To implement this project we have apply following techniques

1. Input images: using this module we will input satellite images
2. Canny Edge Detection: using this method we will extract edges from images
3. Hough Transformation: if extracted edges contains straight line then we got road in input images and then extract features
4. LBP: Extracted features will be input to LBP algorithm to extract out road lines from images
5. AdaBoost Learning: extracted LBP features will be input to AdaBoost algorithm to train a model.
6. Road Extraction: AdaBoost trained model will be applied on test image to get road

To implement this project we have designed following Modules

1. Upload Satellite Images Dataset: using this module we will upload satellite images dataset to application
2. Run Canny, Hough & LBP Features Extraction Algorithms: using this module we will read all images and then extract features using Canny, Hough and LBP
3. Train AdaBoost Algorithm: using this module we will input extracted features to AdaBoost algorithm to train a model
4. Road Extraction from Test Images: using this module we will input test image and then AdaBoost will learn and extract road from given satellite images

**SOFTWARE ENVIRONMENT :**

**What is Python :**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

**Advantages of Python :-**

Let’s see how Python dominates over other languages.

## 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

## 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

## 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

## 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

## 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

## 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

## 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

## 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

## 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

## 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

# Advantages of Python Over Other Languages :

## 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

## 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

## 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

## Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

## History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

## What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

## Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/" \t "_blank), [TensorFlow](https://www.tensorflow.org/" \t "_blank), [Scikit-learn](https://scikit-learn.org/stable/" \t "_blank), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### Advantages of Machine learning :-

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning :-

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [https://www.python.org](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

**6.SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

**Unit testing**

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

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

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

**System Test**

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

**White Box Testing**

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

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

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

**Test strategy and approach**

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

**Test objectives**

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

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**TESTING**

### SOFTWARETESTING

## Testing

Testingisaprocessofexecutingaprogramwiththeaimoffindingerror.Tomakeoursoftware perform well it should be error free. If testing is done successfully it will remove all the errors from thesoftware.

#### 6.1.1 Types ofTesting

* + - 1. White BoxTesting
      2. Black BoxTesting
      3. Unit testing
      4. IntegrationTesting
      5. AlphaTesting
      6. BetaTesting
      7. Performance Testing and so on

#### White BoxTesting

Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performedbysoftwaredevelopers

**Black BoxTesting**

A method of software testing that verifies the functionality of an application without having specificknowledgeoftheapplication'scode/internalstructure.Testsarebasedonrequirementsandfunctionality.

**Unit Testing**

Software verification and validation method in which a programmer tests if individual unitsof source code are fit for use. It is usually conducted by the developmentteam.

**IntegrationTesting**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

**Alpha Testing**

Type of testing a software product or system conducted at the developer's site. Usually it is performed by the end users.

**BetaTesting**

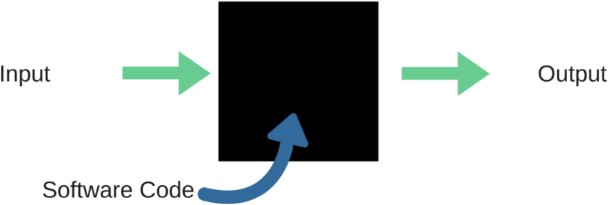
Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

**PerformanceTesting**

Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.

#### Black BoxTesting

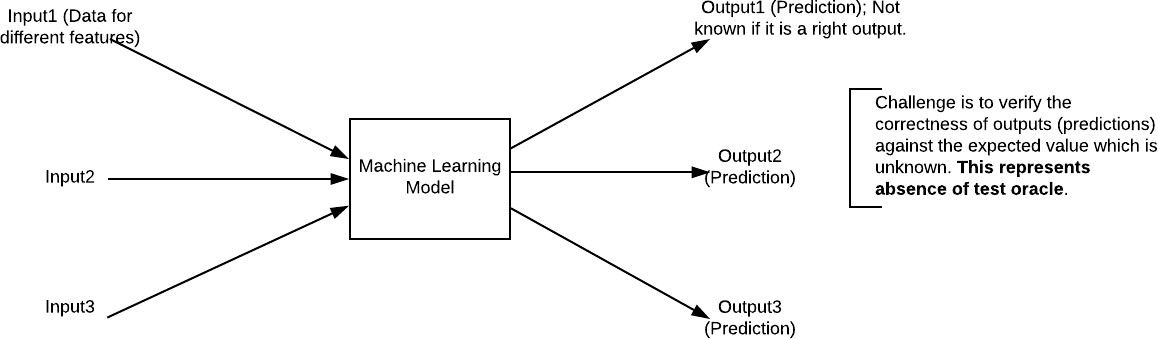
Blackbox testing is testing the functionality of an application without knowing the details of itsimplementationincludinginternalprogramstructure,datastructuresetc.Testcasesforblack box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing. Fig.4.1 represents the black box testing:



**Fig.:**Black Box Testing

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning

model, the algorithm used to create the model etc. The challenge, however, is to verify the test outcome against the expected values that are known beforehand.



**Fig.:**Black Box Testing for Machine Learning algorithms

The above Fig.4.2 represents the black box testing procedure for machine learning algorithms.

**Table.4.1:**Black box Testing

|  |  |  |
| --- | --- | --- |
| **Input** | **Actual Output** | **Predicted Output** |
| [16,6,324,0,0,0,22,0,0,0,0,0,0] | 0 | 0 |
| [16,7,263,7,0,2,700,9,10,1153,832,9,2] | 1 | 1 |

The model gives out the correct output when different inputs are given which are mentioned in Table 4.1. Therefore the program is said to be executed as expected or correct program

## Testing

Testingisaprocessofexecutingaprogramwiththeaimoffindingerror.Tomakeoursoftware perform well it should be error free. If testing is done successfully it will remove all the errors from thesoftware.

#### 7.2.2 Types ofTesting

* + - 1. White BoxTesting
      2. Black BoxTesting
      3. Unit testing
      4. IntegrationTesting
      5. AlphaTesting
      6. BetaTesting
      7. Performance Testing and so on

#### White BoxTesting

Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performedbysoftwaredevelopers

**Black BoxTesting**

A method of software testing that verifies the functionality of an application without having specificknowledgeoftheapplication'scode/internalstructure.Testsarebasedonrequirementsandfunctionality.

**Unit Testing**

Software verification and validation method in which a programmer tests if individual unitsof source code are fit for use. It is usually conducted by the developmentteam.

**IntegrationTesting**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

**Alpha Testing**

Type of testing a software product or system conducted at the developer's site. Usually it is performed by the end users.

**BetaTesting**

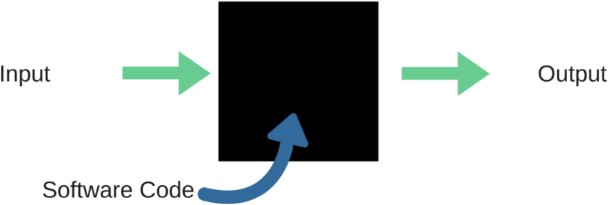
Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

**PerformanceTesting**

Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.

#### Black BoxTesting

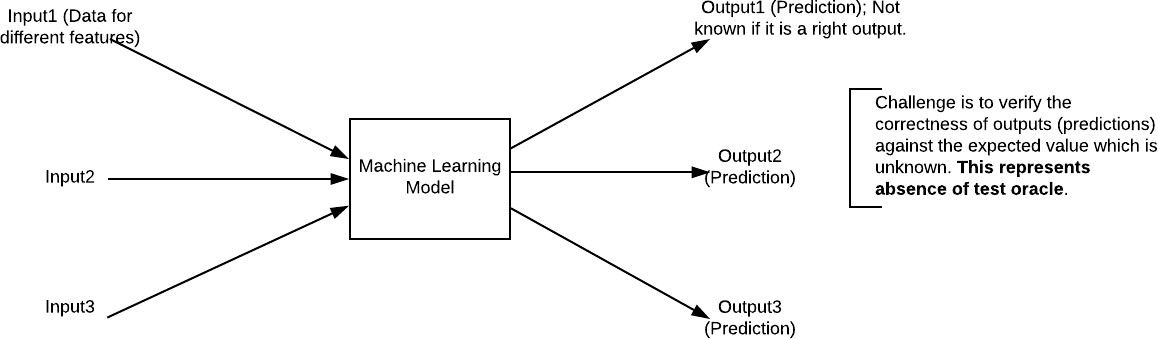
Blackbox testing is testing the functionality of an application without knowing the details of itsimplementationincludinginternalprogramstructure,datastructuresetc.Testcasesforblack box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing. Fig.4.1 represents the black box testing:



**Fig.:**Black Box Testing

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning

model, the algorithm used to create the model etc. The challenge, however, is to verify the test outcome against the expected values that are known beforehand.



**Fig.:**Black Box Testing for Machine Learning algorithms

The above Fig.4.2 represents the black box testing procedure for machine learning algorithms.

**Table.4.1:**Black box Testing

|  |  |  |
| --- | --- | --- |
| **Input** | **Actual Output** | **Predicted Output** |
| [16,6,324,0,0,0,22,0,0,0,0,0,0] | 0 | 0 |
| [16,7,263,7,0,2,700,9,10,1153,832,9,2] | 1 | 1 |

The model gives out the correct output when different inputs are given which are mentioned in Table 4.1. Therefore the program is said to be executed as expected or correct program

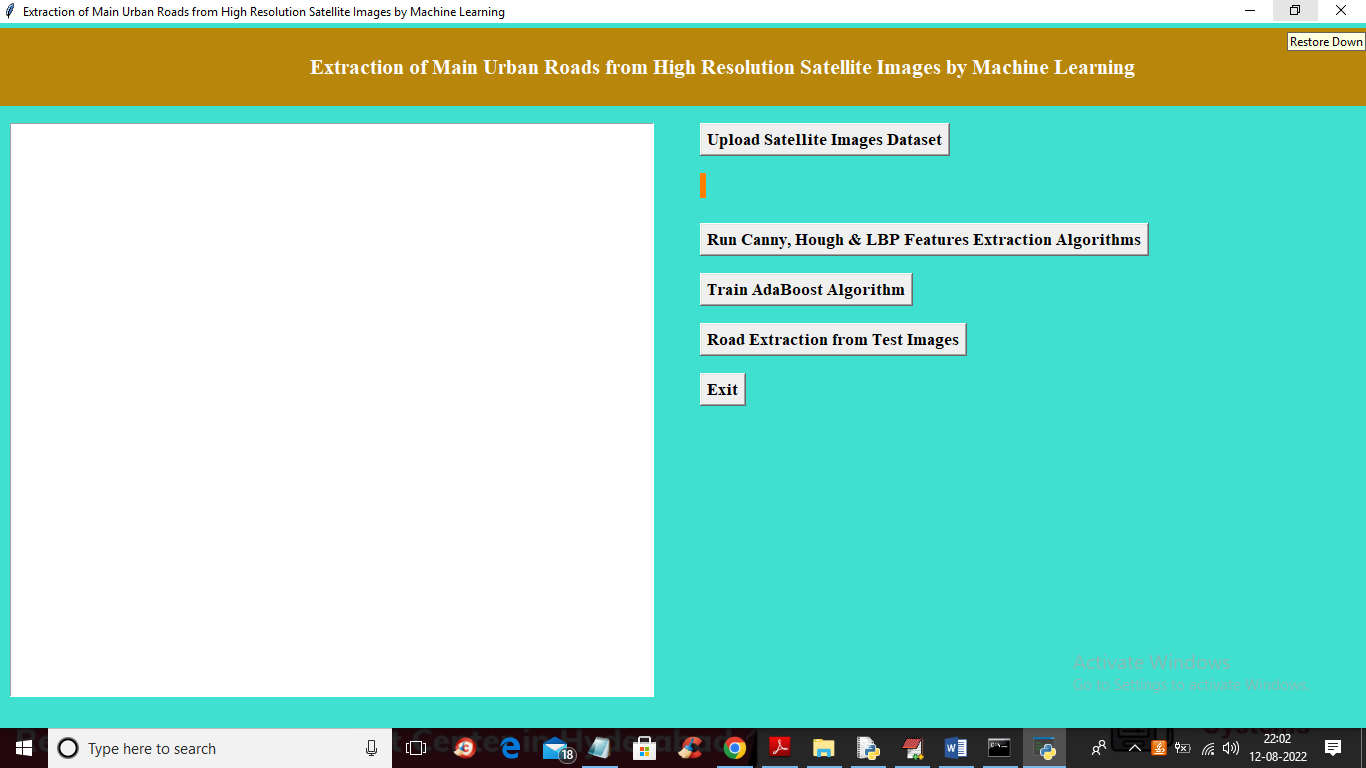
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | Test Case | Test Case | Test Steps | | | Test | Test |
| Cas | Name | Description | Step | Expected | Actual | Case | Priorit |
| e Id |  |  |  |  |  | Statu | Y |
|  |  |  |  |  |  | s |  |
| 01 | Start the | Host the | If it | We | The | High | High |
|  | Applicatio | application | doesn't | cannot | application |  |  |
|  | N | and test if it | Start | run the | hosts |  |  |
|  |  | starts |  | applicati | success. |  |  |
|  |  | making sure |  | on. |  |  |  |
|  |  | the required |  |  |  |  |  |
|  |  | software is |  |  |  |  |  |
|  |  | available |  |  |  |  |  |
| 02 | Home Page | Check the | If it | We | The | High | High |
|  |  | deployment | doesn’t | cannot | application |  |  |
|  |  | environmen | load. | access | is running |  |  |
|  |  | t for |  | the | successfully |  |  |
|  |  | properly |  | applicati | . |  |  |
|  |  | loading the |  | on. |  |  |  |
|  |  | application. |  |  |  |  |  |
| 03 | User | Verify the | If it | We | The | High | High |
|  | Mode | working of | doesn’t | cannot | application |  |  |
|  |  | the | Respond | use the | displays the |  |  |
|  |  | application |  | Freestyle | Freestyle |  |  |
|  |  | in freestyle |  | mode. | Page |  |  |
|  |  | mode |  |  |  |  |  |
| 04 | Data Input | Verify if the | If it fails | We | The | High | High |
|  |  | application | to take the | cannot | application |  |  |
|  |  | takes input | input or | proceed | updates the |  |  |
|  |  | and updates | store in | further | input to application |  |  |
|  |  |  | The |  |  |  |  |
|  |  |  | Database |  |  |  |  |

**CODE :**

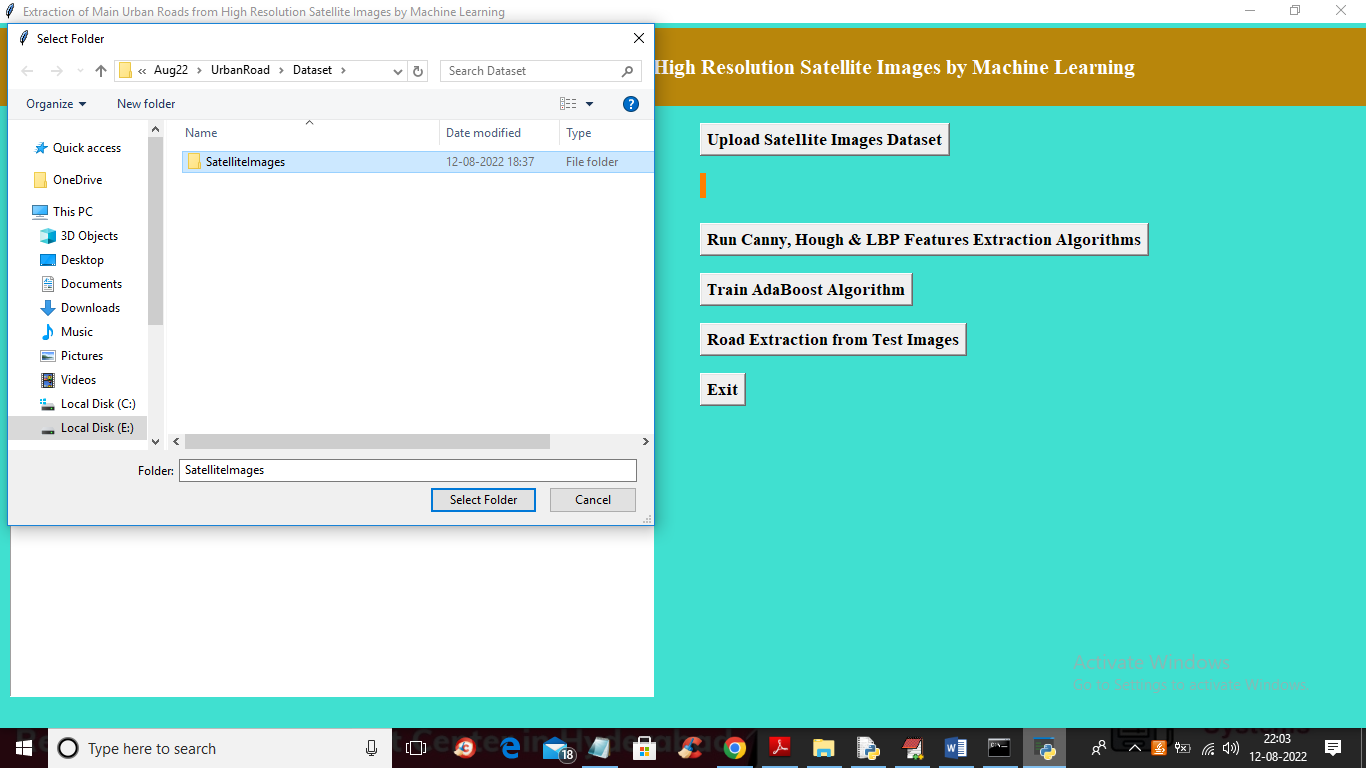
U HAVE TO ADD THE CODE:

**SCREEN SHORTS :**

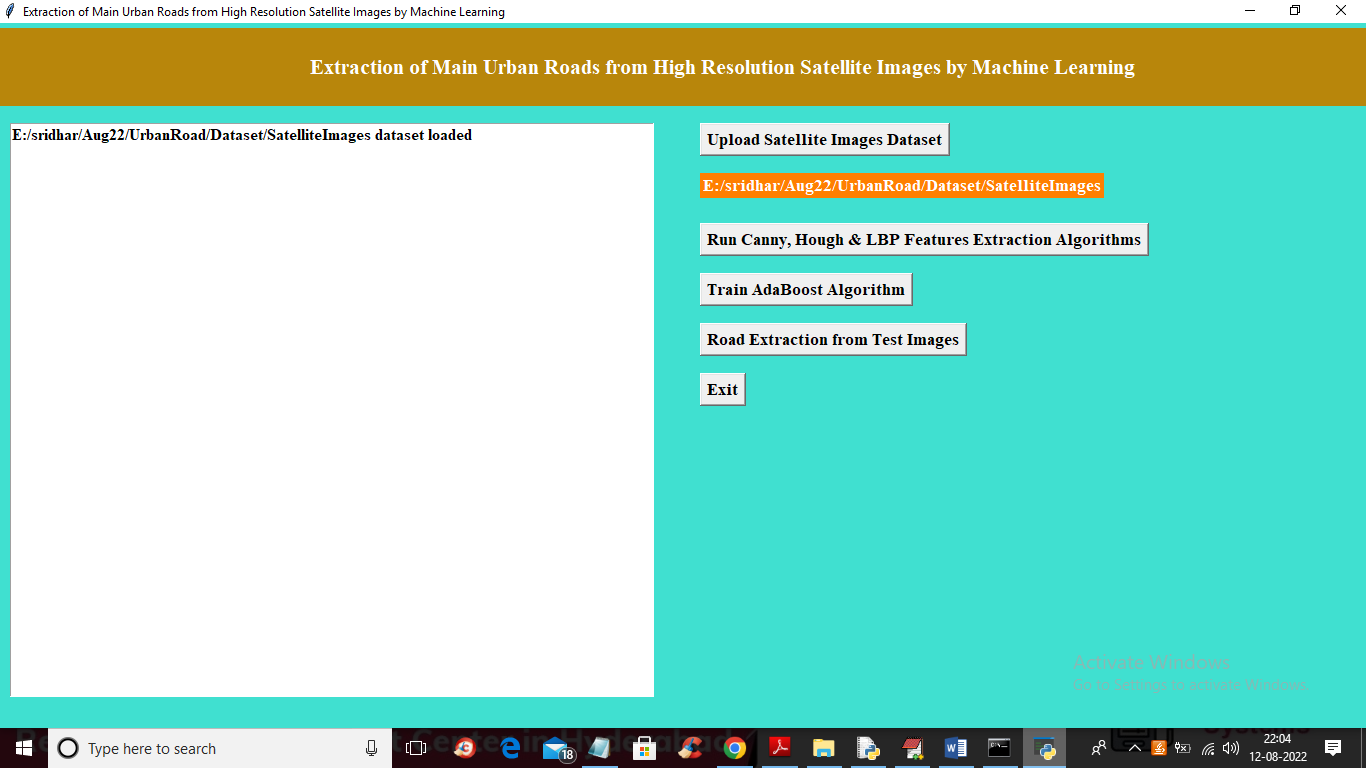
To run project double click on ‘run.bat’ file to get below screen



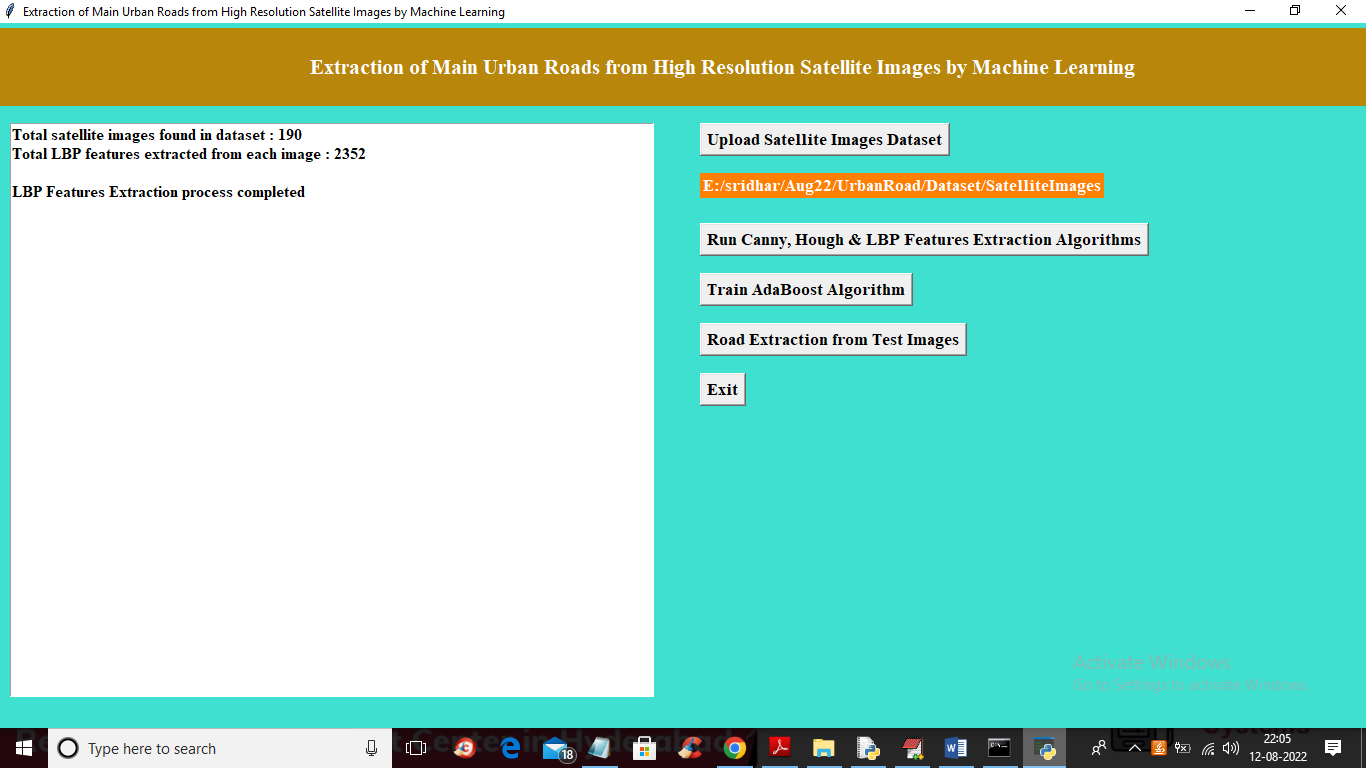
In above screen click on ‘Upload Satellite Images Dataset’ button to upload dataset and get below output



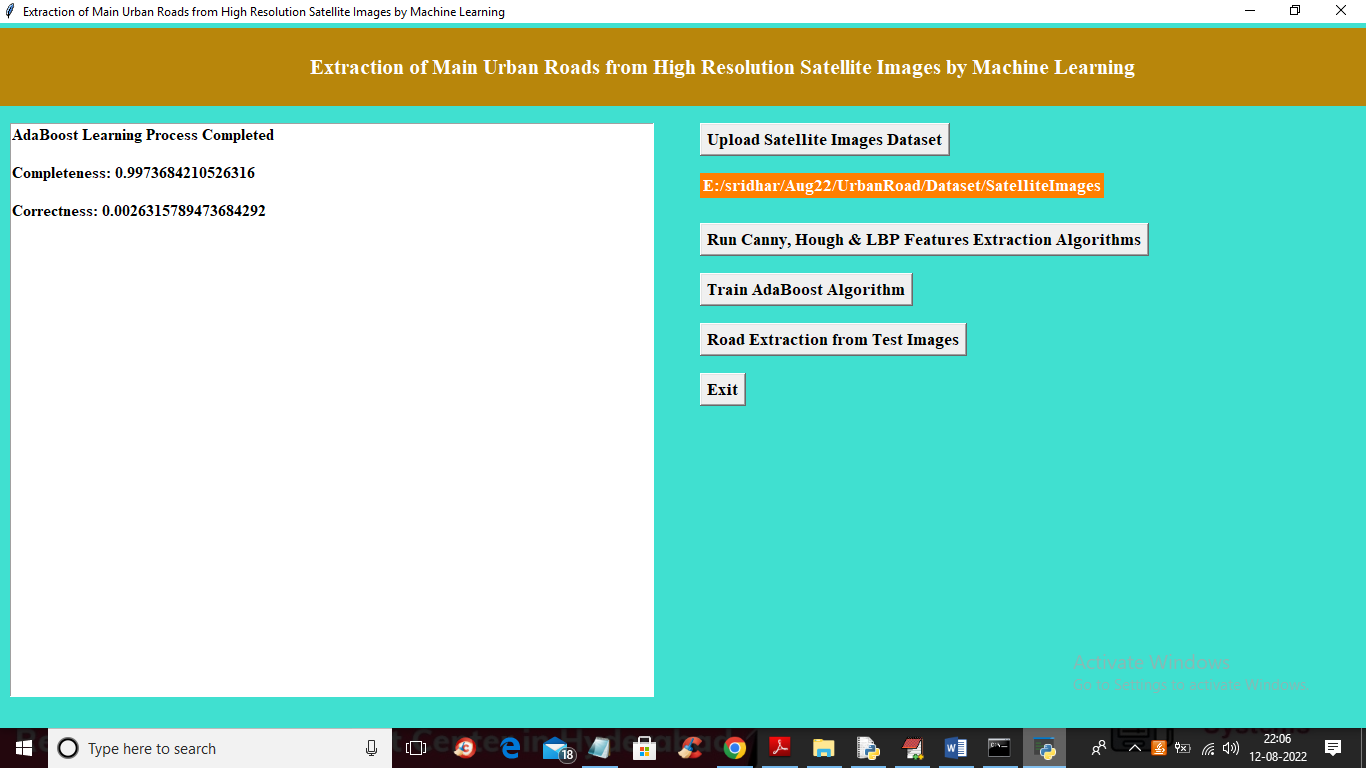
In above screen selecting and uploading ‘Satellite Images’ folder and then click on ‘Select Folder’ button to load dataset and get below output



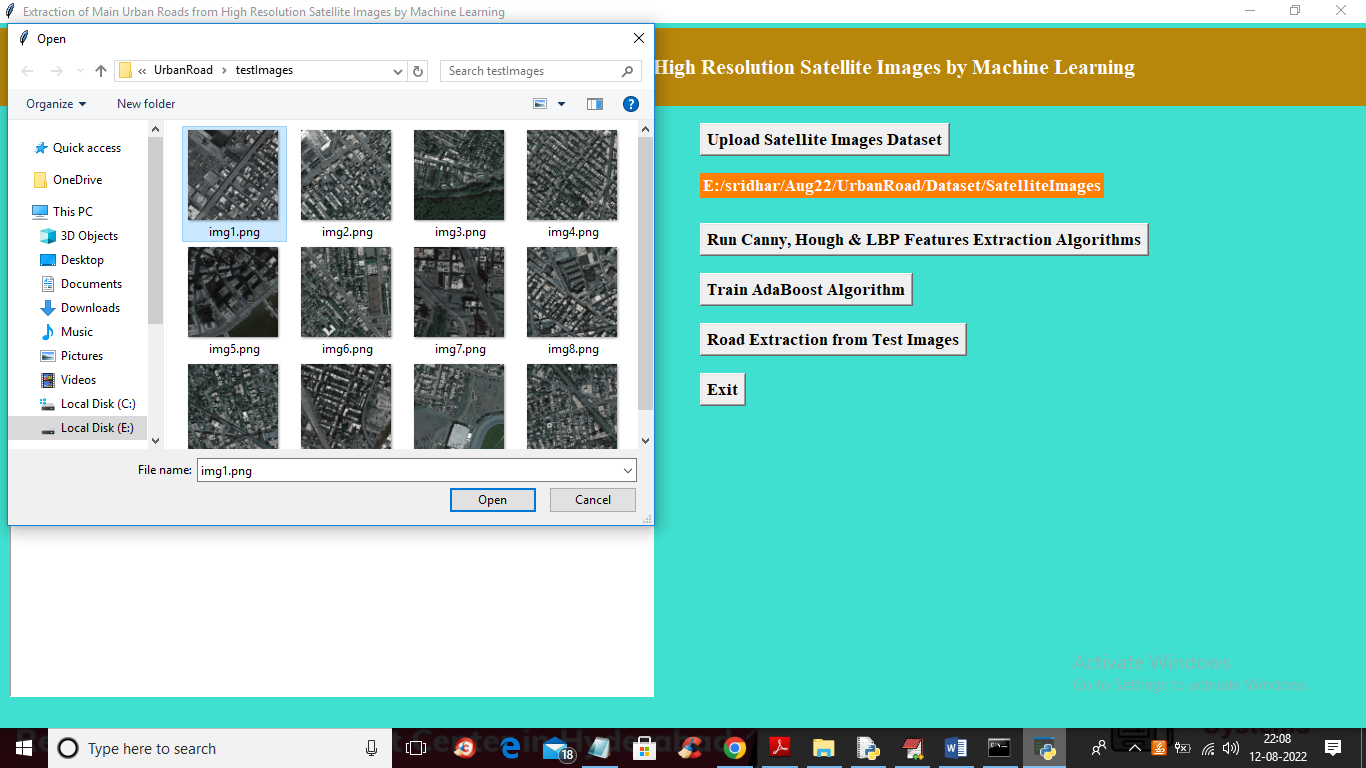
In above screen dataset loaded and now click on ‘Run Canny, Hough & LBP Features Extraction Algorithms’ button to extract features from all dataset images and get below output



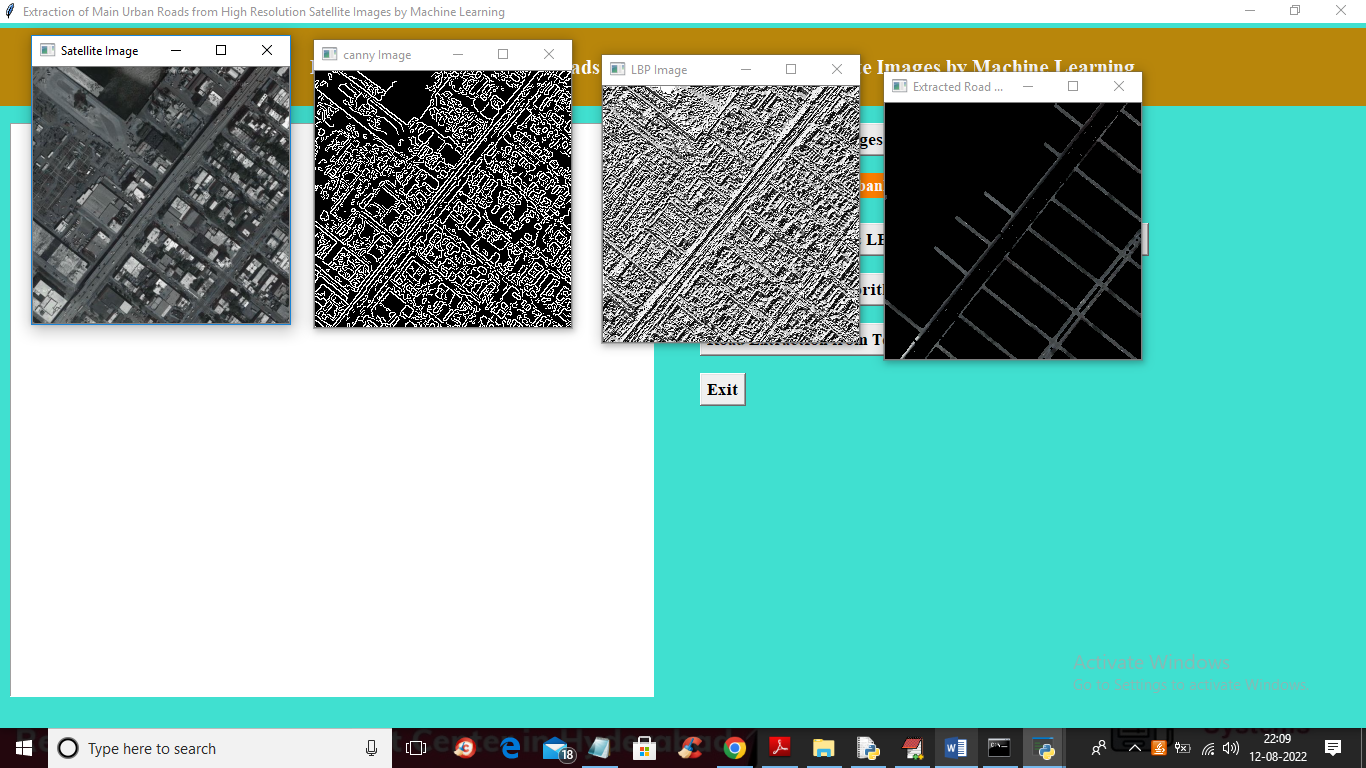
In above screen we can see application found 190 images in dataset and then extract 2352 features from each images and then generate features extracted training array and now click on ‘Train AdaBoost Algorithm’ button to train AdaBoost and get below output



In above screen AdaBoost training completed and we got Completeness (refers to correct prediction %) value as 0.99% and we got Correctness (wrong prediction %) as 0.0026 and now click on ‘Road Extraction from Test Images’ button to upload Satellite image and then AdaBoost will extract road from it

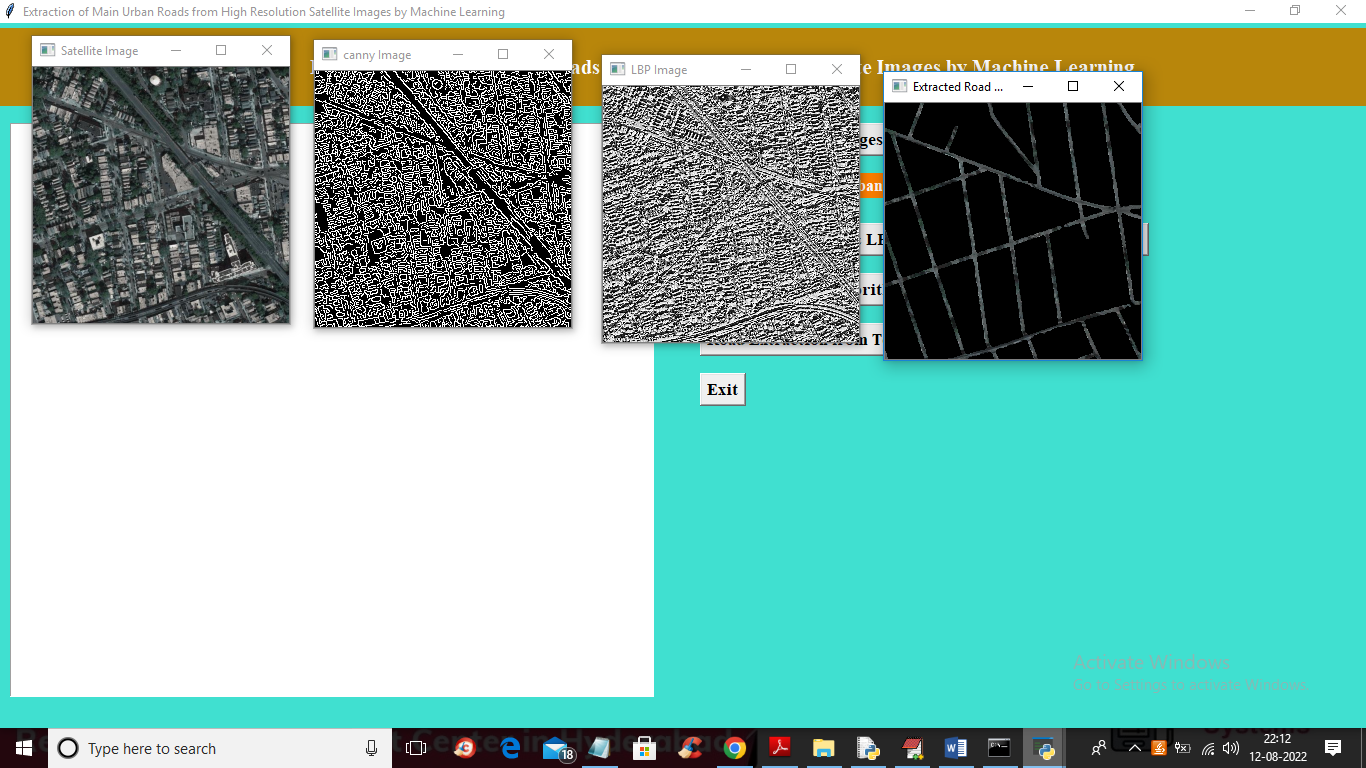


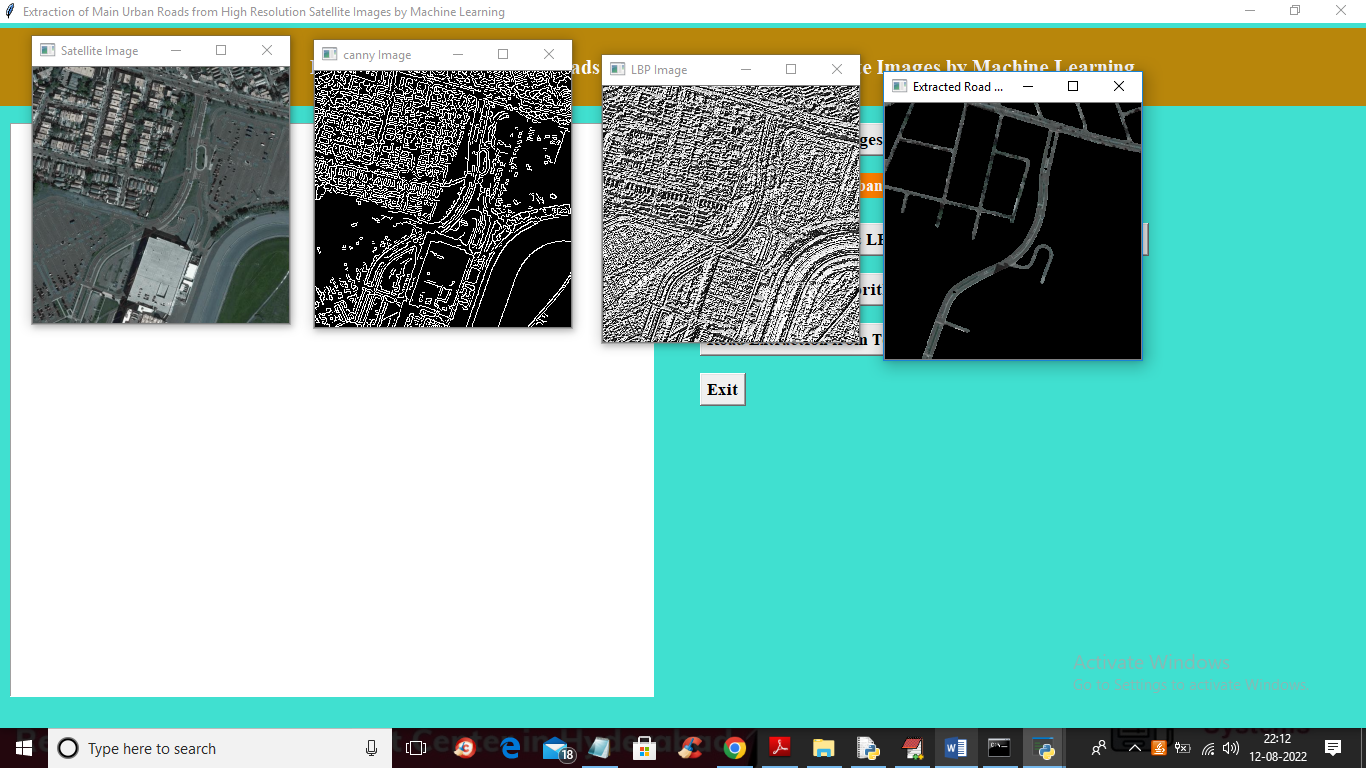
In above screen selecting and uploading ‘img1.png’ file and then click on ‘Open’ button to get below output

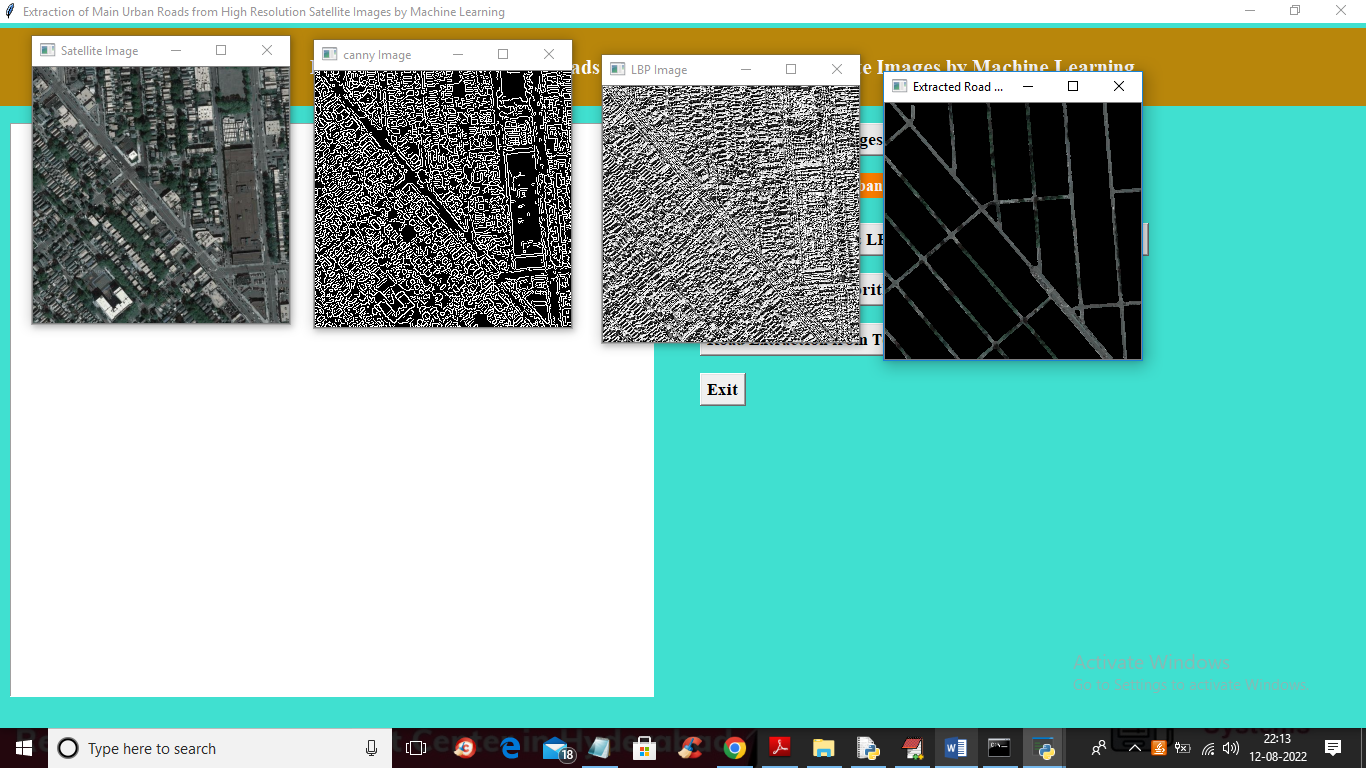


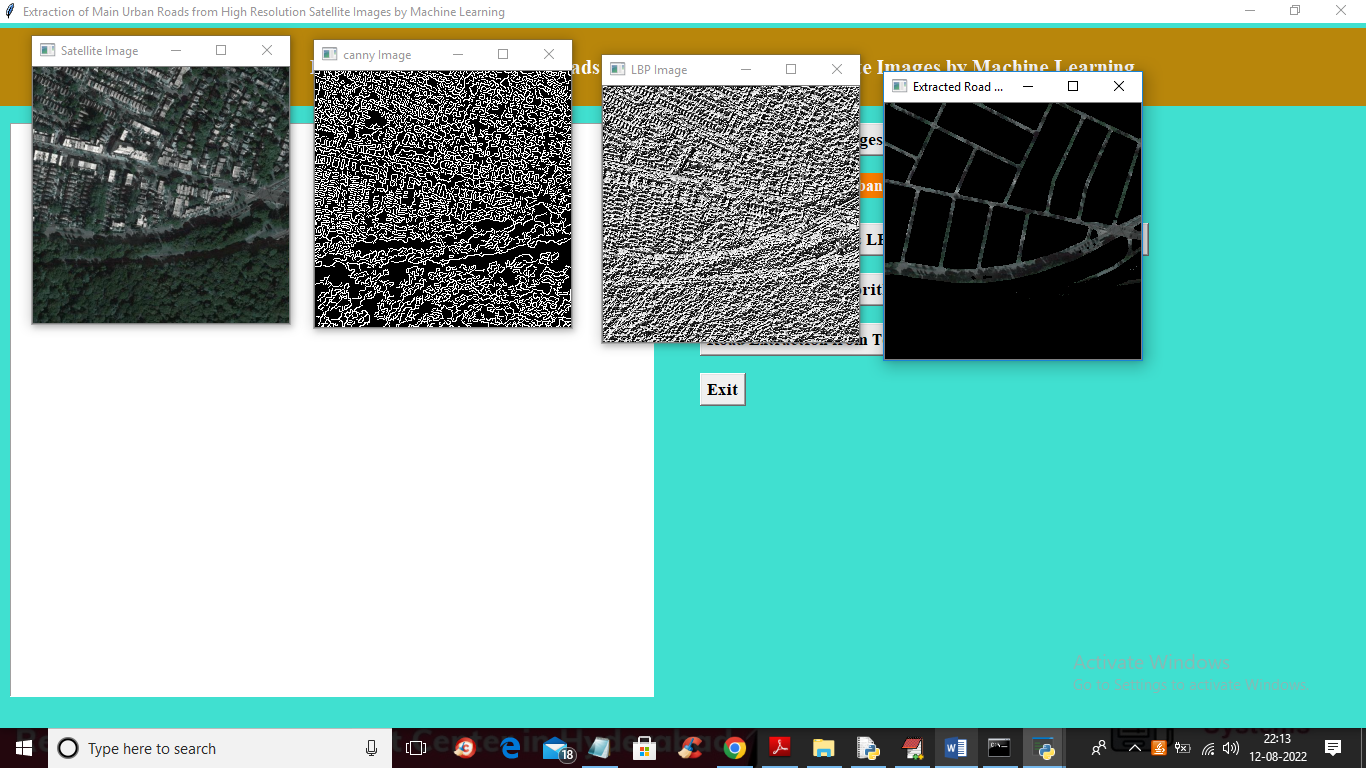
In above screen first image is the uploaded Satellite image and second image is Canny Edge detected image and 3rd image is the LBP image and in LBP image we can see straight ROAD line clearly and this line will extract by AdaBoost and give output as 4th image and in 4th image we can see extracted road clearly and in 4th image we can see small white colour dots as vehicles. Similarly you can upload and test other images

Below is the extracted road from other satellite images









**CONCLUSION:**

This article summarizes the methods extracting roads from HRSI in the past two decades, which are classified into heuristic and data-driven. For heuristic algorithms, we further divide the algorithms into two types (i.e., automatic and semiautomatic) and introduce the principles and inspirations of them. For data- driven methods, we focus on the methods based on deep learning, and survey how the different deep learning technologies and frameworks are applied in road extraction methods. Compre- hensive comparisons are described inner and inter different type of methods. The hand-crafted feature engineering required in the heuristic algorithms decreases the algorithms’ general- ization, makes them difficult to apply to large-scale datasets. Comparisons of performances achieved by heuristic methods and data-driven methods in the same dataset show the advance of the data-driven method. This article cannot elaborate on all kinds of heuristic algorithms due to the diversity of them and fails to compare their performances objectively for the lack of benchmark dataset used by them. In general, the existing road extraction algorithms are not smart enough to fulfill practical applications. Interventions, such as adjusting some parameters, specifying the road type or manually placing the seeds, etc., are required by most of the algorithms. Even by manual intervention, satisfactory results are sometimes difficult to obtain. In addition, different extraction methods have their own advantages and disadvantages. There- fore, the intelligence of road extraction for RS imagery still requires further study. Future research may continue from the following aspects.

#### URL listing

|  |  |
| --- | --- |
| Websites | Data collected |
| <https://wikipedia.org> | Searching of any information that will be used in documentation. |
| <https://dev.sqlserver.com/doc> | SQL server it performing in mainly depending on the one of the database using. |
| <https://www.answers.com> | Answers.com, online dictionary, encyclopedia and much more. |
| <https://google.co.in> | Any information searching and downloading. |
| <https://training-classes.com> | Designing part information as gathered |

#### 

**Scope :**

Road detection and extraction from multispectral satellite images has been considered an essential area of research in remote sensing and computer vision. Road detection and extraction is more effective in rural areas than urban areas where man-made objects are less and possible to detect road more easily, and minimized the human labour in some extent. In this study, we proposed an efficient road extraction method that can successfully extract road from multispectral satellite images having different size and shape with minimal human intervention. The effect of the input road performance parameters for neural network high resolution satellite tested on multispectral image. The back propagation algorithm was introduce with different size of hidden layers equipped with different iteration to avoid overtraining issues. Proposed methodology can be further used to extract other natural and man-made object from multispectral satellite images.

**REFERENCES :**

[1] H. R. R. Bakhtiari, A. Abdollahi, and H. Rezaeian, “Semi automatic road extraction from digital images,” Egypt. J. Remote Sens. Space Sci., vol. 20, no. 1, pp. 117–123, 2017.

[2] G. Panteras and G. Cervone, “Enhancing the temporal resolution of satellite-based flood extent generation using crowdsourced data for dis- aster monitoring,” Int. J. Remote Sens., vol. 39, no. 5, pp. 1459–1474, 2018.

[3] Z. Zhu, S. Yang, G. Xu, X. Lin, and D. Shi, “Fast road classification and orientation estimation using omni-view images and neural networks,” IEEE Trans. Image Process., vol. 7, no. 8, pp. 1182–1197, Aug. 1998.

[4] J. Wang, J. Song, M. Chen, and Y. Zhi, “Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine,” Int. J. Remote Sens., vol. 36, no. 12, pp. 3144–3169, 2015.

[5] W. Shi, Z. Miao, and J. Debayle, “An integrated method for urban main- road centerline extraction from optical remote sensed imagery,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 6, pp. 3359–3372, Jun. 2014.

[6] S. Saito, T. Yamashita, and Y. Aoki, “Multiple object extraction from aerial imagery with convolutional neural networks,” J. Imag. Sci. Tech- nol., vol. 60, no. 1, pp. 10402–10409, 2016.

[7] A. O. Ok, “Automated extraction of buildings and roads in a graph partitioning framework,” ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., vol. 3, pp. 79–84, 2013.

[8] W. Wang, N. Yang, Y. Zhang, F. Wang, T. Cao, and P. Eklund, “A review of road extraction from remote sensing images,” J. Traffic Transp. Eng. (Engl. Ed.), vol. 3, no. 3, pp. 271–282, 2016.

[9] Y. Wei, Z. Wang, and M. Xu, “Road structure refined CNN for road extraction in aerial image,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 5, pp. 709–713, May 2017.

[10] T. Perciano, F. Tupin, R. Hirata Jr., and R. M. Cesar Jr., “A two-level Markov random field for road network extraction and its application with optical, SAR, and multitemporal data,” Int. J. Remote Sens., vol. 37, no. 16, pp. 3584–3610, 2016.

[11] R. P. Krish, J. Fierrez, D. Ramos, F. Alonso-Fernandez, and J. Bigun, “Improving automated latent fingerprint identification using extended minutia types,” Inf. Fusion, vol. 50, pp. 9–19, 2019.

[12] H. S. Dadi and G. K. M. Pillutla, “Improved face recognition rate using HOG features and SVM classifier,” IOSR J. Electron. Commun. Eng., vol. 11, no. 4, pp. 34–44, 2016. [

13] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4651–4659.

[14] R. Liu, J. Song, Q. Miao, P. Xu, and Q. Xue, “Road centerlines extraction from high resolution images based on an improved directional segmenta- tion and road probability,” Neurocomputing, vol. 212, no. C, pp. 88–95, 2016.

[15] Z. Zhang, Q. Liu, and Y. Wang, “Road extraction by deep residual U-Net,” IEEE Geosci. Remote Sens. Lett., vol. 15, no. 5, pp. 749–753, May 2018.

[16] D. Marr, “Vision: A computational investigation into the human repre- sentation and processing of visual information,” Quart. Rev. Biol., vol. 8, pp. 9–37, 1982.

[17] Y. Trinder, C. John, and Wang, “Knowledge-based road interpreta- tion in aerial images,” Int. Arch. Photogramm. Remote Sens., vol. 32, pp. 635–640, 1998.

[18] Y. Zang, C. Wang, L. Cao, Y. Yu, and J. Li, “Road network extraction via aperiodic directional structure measurement,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 6, pp. 3322–3335, Jun. 2016.

[19] H. Pan, Y. Jia, and Z. Lv, “An adaptive multifeature method for semi- automatic road extraction from high-resolution stereo mapping satellite images,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 2, pp. 201–205, Feb. 2019.

[20] A. Baumgartner, C. Steger, H. Mayer, W. Eckstein, and H. Ebner, “Automatic road extraction based on multi-scale, grouping, and context,” Photogramm. Eng. Remote Sens., vol. 65, no. 7, pp. 777–785, 1999.

[21] S. Hinz and A. Baumgartner, “Automatic extraction of urban road net- works from multiview aerial imagery,” ISPRS J. Photogramm. Remote Sens., vol. 58, pp. 83–98, 2003.

[22] M. Maboudi, J. Amini, S. Malihi, and M. Hahn, “Integrating fuzzy object based image analysis and ant colony optimization for road extraction from remotely sensed images,” ISPRS J. Photogramm. Remote Sens., vol. 138, pp. 151–163, 2018.