

UNIT-5

5.1 Natural Language Processing

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be –

- Speech
- Written Text

5.1.1 Components of NLP

There are two components of NLP as given –

Natural Language Understanding (NLU)

Understanding involves the following tasks –

- Mapping the given input in natural language into useful representations.
- Analyzing different aspects of the language.

Natural Language Generation (NLG)

It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

It involves –

- **Text planning** – It includes retrieving the relevant content from knowledge base.
- **Sentence planning** – It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
- **Text Realization** – It is mapping sentence plan into sentence structure.

The NLU is harder than NLG.

Difficulties in NLU

NL has an extremely rich form and structure.

It is very ambiguous. There can be different levels of ambiguity –

- **Lexical ambiguity** – It is at very primitive level such as word-level.
- For example, treating the word “board” as noun or verb?
- **Syntax Level ambiguity** – A sentence can be parsed in different ways.
- For example, “He lifted the beetle with red cap.” – Did he use cap to lift the beetle or he lifted a beetle that had red cap?
- **Referential ambiguity** – Referring to something using pronouns. For example, Rima went to Gauri. She said, “I am tired.” – Exactly who is tired?
- One input can mean different meanings.
- Many inputs can mean the same thing.

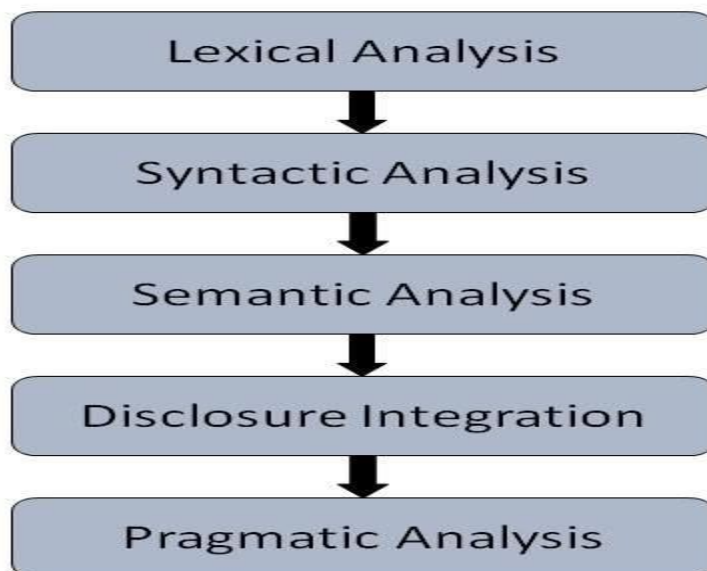
5.1.2 NLP Terminology

- **Phonology** – It is study of organizing sound systematically.
- **Morphology** – It is a study of construction of words from primitive meaningful units.
- **Morpheme** – It is primitive unit of meaning in a language.
- **Syntax** – It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.
- **Semantics** – It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.
- **Pragmatics** – It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.
- **Discourse** – It deals with how the immediately preceding sentence can affect the interpretation of the next sentence.
- **World Knowledge** – It includes the general knowledge about the world.

5.1.3 Steps in NLP

There are general five steps –

- **Lexical Analysis** – It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
- **Syntactic Analysis (Parsing)** – It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.



- **Semantic Analysis** – It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as “hot ice-cream”.
- **Discourse Integration** – The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
- **Pragmatic Analysis** – During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Implementation Aspects of Syntactic Analysis

There are a number of algorithms researchers have developed for syntactic analysis, but we consider only the following simple methods –

- Context-Free Grammar
- Top-Down Parser

Let us see them in detail –

Context-Free Grammar

It is the grammar that consists rules with a single symbol on the left-hand side of the rewrite rules. Let us create grammar to parse a sentence –

“The bird pecks the grains”

Articles (DET) – a | an | the

Nouns – bird | birds | grain | grains

Noun Phrase (NP) – Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

Verbs – pecks | pecking | pecked

Verb Phrase (VP) – NP V | V NP

Adjectives (ADJ) – beautiful | small | chirping

The parse tree breaks down the sentence into structured parts so that the computer can easily understand and process it. In order for the parsing algorithm to construct this parse tree, a set of rewrite rules, which describe what tree structures are legal, need to be constructed.

These rules say that a certain symbol may be expanded in the tree by a sequence of other symbols. According to first order logic rule, if there are two strings Noun Phrase (NP) and Verb Phrase (VP), then the string combined by NP followed by VP is a sentence. The rewrite rules for the sentence are as follows –

S → NP VP

NP → DET N | DET ADJ N

VP → V NP

Lexocon –

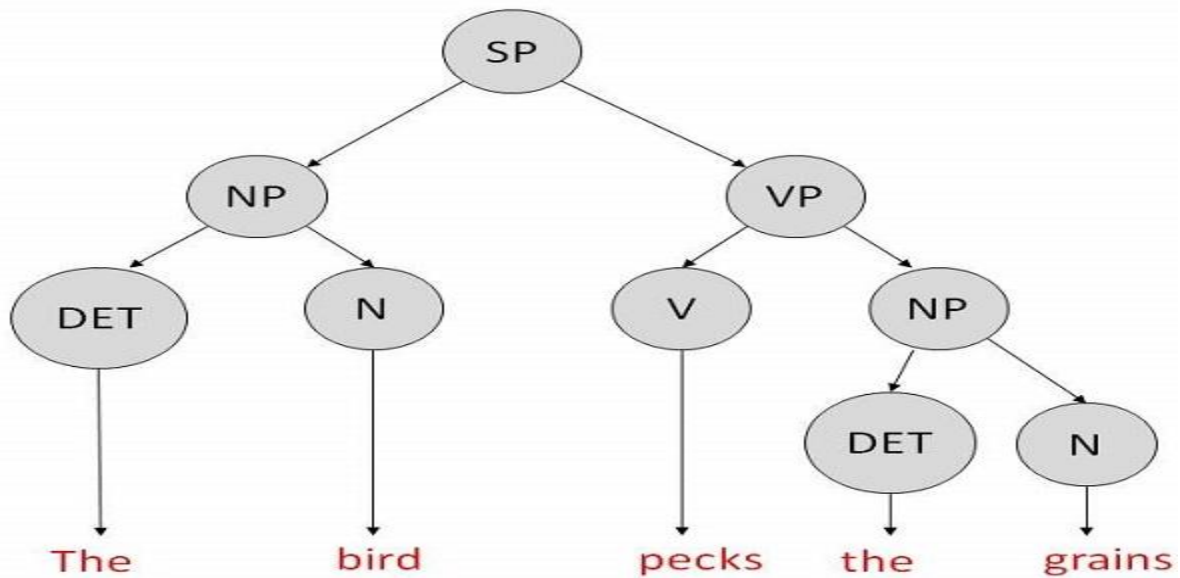
DET → a | the

ADJ → beautiful | perching

N → bird | birds | grain | grains

V → peck | pecks | pecking

The parse tree can be created as shown –



Now consider the above rewrite rules. Since V can be replaced by both, "peck" or "pecks", sentences such as "The bird peck the grains" can be wrongly permitted. i. e. the subject-verb agreement error is approved as correct.

Merit – The simplest style of grammar, therefore widely used one.

Demerits –

- They are not highly precise. For example, "The grains peck the bird", is a syntactically correct according to parser, but even if it makes no sense, parser takes it as a correct sentence.
- To bring out high precision, multiple sets of grammar need to be prepared. It may require a completely different sets of rules for parsing singular and plural variations, passive sentences, etc., which can lead to creation of huge set of rules that are unmanageable.

Top-Down Parser

Here, the parser starts with the S symbol and attempts to rewrite it into a sequence of *terminal symbols* that matches the classes of the words in the input sentence until it consists entirely of terminal symbols.

These are then checked with the input sentence to see if it matched. If not, the process is started over again with a different set of rules. This is repeated until a specific rule is found which describes the structure of the sentence.

Merit – It is simple to implement.

Demerits –

- It is inefficient, as the search process has to be repeated if an error occurs.
- Slow speed of working.

Expert systems (ES) are one of the prominent research domains of AI. It is introduced by the researchers at Stanford University, Computer Science Department.

5.2 Expert Systems

What are Expert Systems?

The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

5.2.1 Characteristics of Expert Systems

- High performance
- Understandable
- Reliable
- Highly responsive

5.2.2 Capabilities of Expert Systems

The expert systems are capable of –

- Advising
- Instructing and assisting human in decision making
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

They are incapable of –

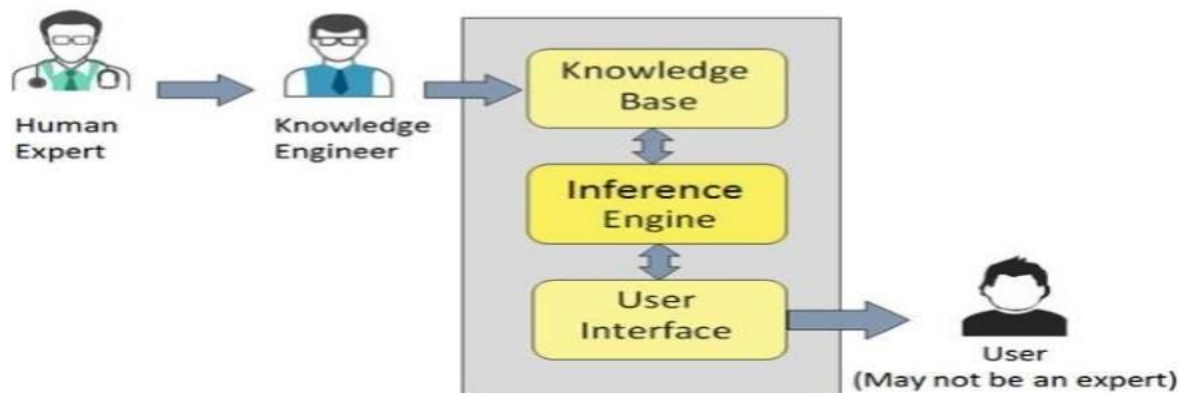
- Substituting human decision makers
- Possessing human capabilities
- Producing accurate output for inadequate knowledge base
- Refining their own knowledge

5.2.3 Components of Expert Systems

The components of ES include –

- Knowledge Base
- Inference Engine
- User Interface

Let us see them one by one briefly –



Knowledge Base

It contains domain-specific and high-quality knowledge.

Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge.

What is Knowledge?

The data is collection of facts. The information is organized as data and facts about the task domain. **Data, information, and past experience** combined together are termed as knowledge.

Components of Knowledge Base

The knowledge base of an ES is a store of both, factual and heuristic knowledge.

- **Factual Knowledge** – It is the information widely accepted by the Knowledge Engineers and scholars in the task domain.
- **Heuristic Knowledge** – It is about practice, accurate judgement, one's ability of evaluation, and guessing.

Knowledge representation

It is the method used to organize and formalize the knowledge in the knowledge base. It is in the form of IF-THEN-ELSE rules.

Knowledge Acquisition

The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base.

The knowledge base is formed by readings from various experts, scholars, and the **Knowledge Engineers**. The knowledge engineer is a person with the qualities of empathy, quick learning, and case analyzing skills.

He acquires information from subject expert by recording, interviewing, and observing him at work, etc. He then categorizes and organizes the information in a meaningful way, in the form of IF-THEN-ELSE rules, to be used by inference machine. The knowledge engineer also monitors the development of the ES.

Inference Engine

Use of efficient procedures and rules by the Inference Engine is essential in deducting a correct, flawless solution.

In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.

In case of rule based ES, it –

- Applies rules repeatedly to the facts, which are obtained from earlier rule application.
- Adds new knowledge into the knowledge base if required.
- Resolves rules conflict when multiple rules are applicable to a particular case.

To recommend a solution, the Inference Engine uses the following strategies –

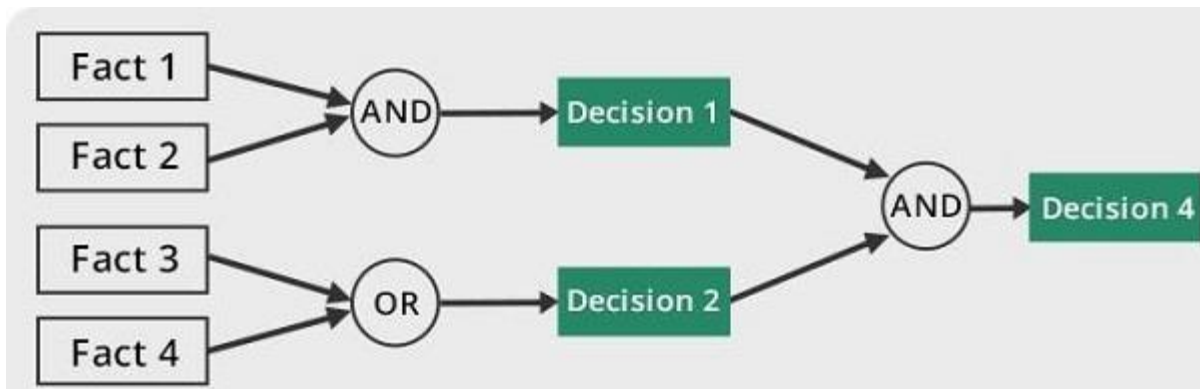
- Forward Chaining
- Backward Chaining

Forward Chaining

It is a strategy of an expert system to answer the question, **“What can happen next?”**

Here, the Inference Engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution.

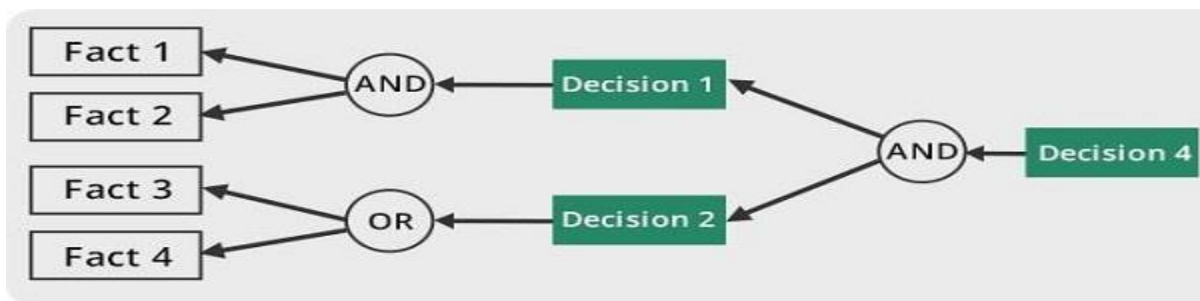
This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.



Backward Chaining

With this strategy, an expert system finds out the answer to the question, “Why this happened?”

On the basis of what has already happened, the Inference Engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.



User Interface

User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence.

It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms –

- Natural language displayed on screen.
- Verbal narrations in natural language.
- Listing of rule numbers displayed on the screen.

The user interface makes it easy to trace the credibility of the deductions.

Requirements of Efficient ES User Interface

- It should help users to accomplish their goals in shortest possible way.
- It should be designed to work for user’s existing or desired work practices.
- Its technology should be adaptable to user’s requirements; not the other way round.
- It should make efficient use of user input.

5.2.4 Expert Systems Limitations

No technology can offer easy and complete solution. Large systems are costly, require significant development time, and computer resources. ESs have their limitations which include –

- Limitations of the technology
- Difficult knowledge acquisition
- ES are difficult to maintain
- High development costs

5.2.5 Applications of Expert System

The following table shows where ES can be applied.

Application	Description
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis Systems to deduce cause of disease from observed data, conduction medical operations on humans.
Monitoring Systems	Comparing data continuously with observed system or with prescribed behavior such as leakage monitoring in long petroleum pipeline.
Process Control Systems	Controlling a physical process based on monitoring.
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

Expert System Technology

There are several levels of ES technologies available. Expert systems technologies include –

- **Expert System Development Environment** – The ES development environment includes hardware and tools. They are –
 - Workstations, minicomputers, mainframes.
 - High level Symbolic Programming Languages such as **LIS**t Programming (LISP) and **PRO**grammation en **LOG**ique (PROLOG).
 - Large databases.
- **Tools** – They reduce the effort and cost involved in developing an expert system to large extent.
 - Powerful editors and debugging tools with multi-windows.
 - They provide rapid prototyping
 - Have Inbuilt definitions of model, knowledge representation, and inference design.
- **Shells** – A shell is nothing but an expert system without knowledge base. A shell provides the developers with knowledge acquisition, inference engine, user interface, and explanation facility. For example, few shells are given below –
 - Java Expert System Shell (JESS) that provides fully developed Java API for creating an expert system.
 - *Vidwan*, a shell developed at the National Centre for Software Technology, Mumbai in 1993. It enables knowledge encoding in the form of IF-THEN rules.

5.2.6 Development of Expert Systems: General Steps

The process of ES development is iterative. Steps in developing the ES include –

Identify Problem Domain

- The problem must be suitable for an expert system to solve it.
- Find the experts in task domain for the ES project.
- Establish cost-effectiveness of the system.

Design the System

- Identify the ES Technology
- Know and establish the degree of integration with the other systems and databases.
- Realize how the concepts can represent the domain knowledge best.

Develop the Prototype

From Knowledge Base: The knowledge engineer works to –

- Acquire domain knowledge from the expert.

- Represent it in the form of If-THEN-ELSE rules.

Test and Refine the Prototype

- The knowledge engineer uses sample cases to test the prototype for any deficiencies in performance.
- End users test the prototypes of the ES.

Develop and Complete the ES

- Test and ensure the interaction of the ES with all elements of its environment, including end users, databases, and other information systems.
- Document the ES project well.
- Train the user to use ES.

Maintain the System

- Keep the knowledge base up-to-date by regular review and update.
- Cater for new interfaces with other information systems, as those systems evolve.

Benefits of Expert Systems

- **Availability** – They are easily available due to mass production of software.
- **Less Production Cost** – Production cost is reasonable. This makes them affordable.
- **Speed** – They offer great speed. They reduce the amount of work an individual puts in.
- **Less Error Rate** – Error rate is low as compared to human errors.
- **Reducing Risk** – They can work in the environment dangerous to humans.
- **Steady response** – They work steadily without getting motional, tensed or fatigued.

Robotics is a domain in artificial intelligence that deals with the study of creating intelligent and efficient robots.

5.3 Robotics

5.3.1 What are Robots?

Robots are the artificial agents acting in real world environment.

Objective

Robots are aimed at manipulating the objects by perceiving, picking, moving, modifying the physical properties of object, destroying it, or to have an effect thereby freeing manpower from doing repetitive functions without getting bored, distracted, or exhausted.

What is Robotics?

Robotics is a branch of AI, which is composed of Electrical Engineering, Mechanical Engineering, and Computer Science for designing, construction, and application of robots.

5.3.2 Aspects of Robotics

- The robots have **mechanical construction**, form, or shape designed to accomplish a particular task.
- They have **electrical components** which power and control the machinery.
- They contain some level of **computer program** that determines what, when and how a robot does something.

5.3.3 Difference in Robot System and Other AI Program

Here is the difference between the two –

AI Programs

They usually operate in computer-stimulated worlds.

The input to an AI program is in symbols and rules.

They need general purpose computers to operate on.

Robots

They operate in real physical world

Inputs to robots is analog signal in the form of speech waveform or images

They need special hardware with sensors and effectors.

Robot Locomotion

Locomotion is the mechanism that makes a robot capable of moving in its environment. There are various types of locomotions –

- Legged
- Wheeled
- Combination of Legged and Wheeled Locomotion
- Tracked slip/skid

Legged Locomotion

- This type of locomotion consumes more power while demonstrating walk, jump, trot, hop, climb up or down, etc.
- It requires more number of motors to accomplish a movement. It is suited for rough as well as smooth terrain where irregular or too smooth surface makes it consume more power for a wheeled locomotion. It is little difficult to implement because of stability issues.
- It comes with the variety of one, two, four, and six legs. If a robot has multiple legs then leg coordination is necessary for locomotion.

The total number of possible **gaits** (a periodic sequence of lift and release events for each of the total legs) a robot can travel depends upon the number of its legs.

If a robot has k legs, then the number of possible events $N = (2k-1)!$.

In case of a two-legged robot ($k=2$), the number of possible events is $N = (2k-1)! = (2*2-1)! = 3! = 6$.

Hence there are six possible different events –

- Lifting the Left leg
- Releasing the Left leg
- Lifting the Right leg
- Releasing the Right leg
- Lifting both the legs together
- Releasing both the legs together

In case of $k=6$ legs, there are 39916800 possible events. Hence the complexity of robots is directly proportional to the number of legs.



Wheeled Locomotion

It requires fewer number of motors to accomplish a movement. It is little easy to implement as there are less stability issues in case of more number of wheels. It is power efficient as compared to legged locomotion.

- **Standard wheel** – Rotates around the wheel axle and around the contact
- **Castor wheel** – Rotates around the wheel axle and the offset steering joint.
- **Swedish 45° and Swedish 90° wheels** – Omni-wheel, rotates around the contact point, around the wheel axle, and around the rollers.
- **Ball or spherical wheel** – Omnidirectional wheel, technically difficult to implement.



Slip/Skid Locomotion

In this type, the vehicles use tracks as in a tank. The robot is steered by moving the tracks with different speeds in the same or opposite direction. It offers stability because of large contact area of track and ground.



Components of a Robot

Robots are constructed with the following –

- **Power Supply** – The robots are powered by batteries, solar power, hydraulic, or pneumatic power sources.
- **Actuators** – They convert energy into movement.
- **Electric motors (AC/DC)** – They are required for rotational movement.
- **Pneumatic Air Muscles** – They contract almost 40% when air is sucked in them.
- **Muscle Wires** – They contract by 5% when electric current is passed through them.
- **Piezo Motors and Ultrasonic Motors** – Best for industrial robots.
- **Sensors** – They provide knowledge of real time information on the task environment. Robots are equipped with vision sensors to be to compute the depth in the environment. A tactile sensor imitates the mechanical properties of touch receptors of human fingertips.

Computer Vision

This is a technology of AI with which the robots can see. The computer vision plays vital role in the domains of safety, security, health, access, and entertainment.

Computer vision automatically extracts, analyzes, and comprehends useful information from a single image or an array of images. This process involves development of algorithms to accomplish automatic visual comprehension.

Hardware of Computer Vision System

This involves –

- Power supply
- Image acquisition device such as camera
- A processor
- A software
- A display device for monitoring the system
- Accessories such as camera stands, cables, and connectors

Tasks of Computer Vision

- **OCR** – In the domain of computers, Optical Character Reader, a software to convert scanned documents into editable text, which accompanies a scanner.
- **Face Detection** – Many state-of-the-art cameras come with this feature, which enables to read the face and take the picture of that perfect expression. It is used to let a user access the software on correct match.
- **Object Recognition** – They are installed in supermarkets, cameras, high-end cars such as BMW, GM, and Volvo.
- **Estimating Position** – It is estimating position of an object with respect to camera as in position of tumor in human's body.

Application Domains of Computer Vision

- Agriculture
- Autonomous vehicles
- Biometrics
- Character recognition
- Forensics, security, and surveillance
- Industrial quality inspection
- Face recognition
- Gesture analysis
- Geoscience
- Medical imagery
- Pollution monitoring
- Process control
- Remote sensing
- Robotics
- Transport

5.3.4 Applications of Robotics

The robotics has been instrumental in the various domains such as –

- **Industries** – Robots are used for handling material, cutting, welding, color coating, drilling, polishing, etc.
- **Military** – Autonomous robots can reach inaccessible and hazardous zones during war. A robot named *Daksh*, developed by Defense Research and Development Organization (DRDO), is in function to destroy life-threatening objects safely.
- **Medicine** – The robots are capable of carrying out hundreds of clinical tests simultaneously, rehabilitating permanently disabled people, and performing complex surgeries such as brain tumors.
- **Exploration** – The robot rock climbers used for space exploration, underwater drones used for ocean exploration are to name a few.
- **Entertainment** – Disney's engineers have created hundreds of robots for movie making.