A PROJECT REPORT

ON

"CANE INTELLIGENCE"

Submitted by

KAZI IRSHAD 16ME24

AWEES SANADI 15ME94

KAZI ZAMMAM 16ME25

SIDDIQUE ZAHOOR 18DME54

In partial fulfillment for the award of the Degree

Of

BACHELOR OF ENGINEERING

IN

MECHANICAL ENGINEERING

UNDER THE GUIDANCE

Of

Prof. UBAID SHAH

DEPARTMENT OF MECHANICAL ENGINEERING

ANJUMAN-I-ISLAM

KALSEKAR TECHNICAL CAMPUS NEW PANVEL,

NAVI MUMBAI – 410206

UNIVERSITY OF MUMBAI

ACADEMIC YEAR 2020 -2021

Internal Examiner   Project coordinator

ANJUMAN-I-ISLAM

KALSEKAR TECHNICAL CAMPUS NEW PANVEL

(Approved by AICTE, recg. By Maharashtra Govt. DTE,

Affiliated to Mumbai University)

PLOT #2&3, SECTOR 16, NEAR THANA NAKA, KHANDAGAON, NEW PANVEL,NAVI MUMBAI-410206,
Tel.: +91 22 27481247/48 * Website: www.aiktc.org

CERTIFICATE

This is to certify that the project entitled

"CANE INTELLIGENCE"

Submitted by

KAZI IRSHAD - 16ME24

KAZI ZAMMAM – 16ME25

SANADI AWEES – 15ME94

SIDDIQUI ZAHOOR – 18DME54

To the Kalsekar Technical Campus, New Panvel is a record of bonafide work carried out by them under our supervision and guidance, for partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Mechanical Engineering as prescribed by **University of Mumbai**, is approved.

PROF.UBAID SHAH
PROGECT GUIDE

PROF.RAHUL THAVAI/
PROF.RIZWAN SHAIKH
PROJECT COORDINATOR

PROF. ZAKIR ANSARI
HOD,MECHANICAL DEPARTMANT

DR.ABDUL RAZZAK HONNUTAGI
DIRECTOR

ANJUMAN-I-ISLAM

KALSEKAR TECHNICAL CAMPUS NEW PANVEL

(Approved by AICTE, recg. By Maharashtra Govt. DTE,

Affiliated to Mumbai University)

PLOT #2&3, SECTOR 16, NEAR THANA NAKA, KHANDAGAON, NEW PANVEL,NAVI MUMBAI-410206, Tel.: +91 22 27481247/48 * Website: www.aiktc.org

APPROVAL OF DISSERTATION

This is to certify that the thesis entitled

"CANE INTELIGENCE"

Submitted by

KAZI IRSHAD - 16ME24

KAZI ZAMMAM – 16ME25

SANADI AWEES – 15ME94

SIDDIQUI ZAHOOR – 18DME54

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Mechanical Engineering, as prescribed by University of Mumbai approved.

(Internal Examiner) **(External Examiner)**

__________________ __________________

Date: ___________

DECLARATION:

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

KAZI IRSHAD-16ME24

(Signature)

KAZI ZAMMAM- 16ME25

(Signature)

SANADI AWEES- 15ME94

(Signature)

SIDDIQUI ZAHOOR-18DME54

Date:

ABSTRACT:

The Cane is a hand-held device that assists blind or visually impaired people to navigate independently. The smart walking stick helps the blind people toperform navigation and to do their work easily and comfortably. In normal stick, the detection of the obstacle is not done and normal stick is not efficient for visually impaired person, because the blind person does not know what type of things or what type of the objects come in front of them. The person cannot recognize what is the size of that object and how far it is from the object. It is difficult for the blind person to move here and there. In smart walking stick, ultrasonic sensor is used to detect the obstacle. When the obstacle becomes detected camera becomes on and the object is detected with the help of a camera. If any obstacle comes in front of blind person, they can know about the obstacle by hearing the sound generated by the microphone. A person with blindness can easily detect obstacles on ground, surface textures, pot holes etc. while travelling with a standard white cane. However cane cannot detect overhanging objects like tree branches, sign boards, open glass windows, etc. Also, at times using a white cane could result in scratching a parked vehicle with a cane, bumping into another person, etc. This Cane is designed to vibrate when detecting obstacles and identify objects with artificial intelligence, allowing users the ability to understand what objects are around them, where they are located, and even how they appear.

1.1 INTRODUCTION:

Visual impairment, also known as vision impairment or vision loss, is a decreased ability to see to a degree that causes problems not fixable by usual means, such as glasses. Some also include those who have a decreased ability to see because they do not have access to glasses or contact lenses. Visual impairment is often defined as a best corrected visual acuity of worse than either 20/40 or 20/60. The term blindness is used for complete or nearly complete vision loss. Visual impairment may cause difficulties with normal daily activities such as reading and walking without adaptive training and equipment.

The most common causes of visual impairment globally are uncorrected refractive errors (43%), cataracts (33%). Refractive errors include near-sightedness, far-sightedness, presbyopia, and astigmatism. Cataracts are the most common cause of blindness. Other disorders that may cause visual problems include age-related macular degeneration, diabetic retinopathy, corneal clouding, childhood blindness, and a number of infections. Visual impairment can also be caused by problems in the brain due to stroke, premature birth, or trauma, among others. These cases are known as cortical visual impairment. Screening for vision problems in children may improve future vision and educational achievement. Screening adults without symptoms is of uncertain benefit.] Diagnosis is by an eye exam.

The World Health Organization (WHO) estimates that 80% of visual impairment is either preventable or curable with treatment. This includes cataracts, the infections river blindness and trachoma, glaucoma, diabetic retinopathy, uncorrected refractive errors, and some cases of childhood blindness. Many people with significant visual impairment benefit from vision rehabilitation, changes in their environment, and assistive devices.

As of 2015 there were 940 million people with some degree of vision loss. 246 million had low vision and 39 million were blind. The majority of people with poor vision are in the developing world and are over the age of 50 years. Rates of visual impairment have decreased since the 1990s. Visual impairments have considerable economic costs both directly due to the cost of treatment and indirectly due to decreased ability to work.

1.1.1 CAUSES:

The most common causes of visual impairment globally in 2010 were:

1. Refractive error (42%)
2. Cataract (33%)
3. Glaucoma (2%)
4. Age-related macular degeneration (1%)
5. Corneal opacification (1%)
6. Diabetic retinopathy (1%)
7. Childhood blindness
8. Trachoma (1%)
9. Undetermined (18%)

The most common causes of blindness worldwide in 2010 were:

1. Cataracts (51%)
2. Glaucoma (8%)
3. Age-related macular degeneration (5%)
4. Corneal opacification (4%)
5. Childhood blindness (4%)
6. Refractive errors (3%)
7. Trachoma (3%)
8. Diabetic retinopathy (1%)
9. Undetermined (21%)

About 90% of people who are visually impaired live in the developing world. Age-related macular degeneration, glaucoma, and diabetic retinopathy are the leading causes of blindness in the developed world.

1.1.2 Cataracts

Cataracts are the congenital and pediatric pathology that describes the greying or opacity of the crystalline lens, which is most commonly caused by intrauterine infections, metabolic disorders, and genetically transmitted syndromes .Cataracts are the leading cause of child and adult blindness that doubles in prevalence with every ten years after the age of 40.[Consequently, today cataracts are more common among adults than in children.[That is, people face higher chances of developing cataracts as they age. Nonetheless, cataracts tend to have a greater financial and emotional toll upon children as they must undergo expensive diagnosis, long term rehabilitation, and visual assistance. Also, according to the Saudi Journal for Health Sciences, sometimes patients experience irreversible amblyopia after pediatric cataract surgery because the cataracts prevented the normal maturation of vision prior to operation. Despite the great progress in treatment, cataracts remain a global problem in both economically developed and developing countries. At present, with the variant outcomes as well as the unequal access to cataract surgery, the best way to reduce the risk of developing cataracts is to avoid smoking and extensive exposure to sun light (i.e. UV-B rays).

1.1.3 Glaucoma

Glaucoma is a congenital and pediatric eye disease characterized by increased pressure within the eye or intraocular pressure (IOP). Glaucoma causes visual field loss as well as severs the optic nerve. Early diagnosis and treatment of glaucoma in patients is imperative because glaucoma is triggered by non-specific levels of IOP. Also, another challenge in accurately diagnosing glaucoma is that the disease has four causes: 1) inflammatory ocular hypertension syndrome (IOHS); 2) severe uveitic angle closure; 3) corticosteroid-induced; and 4) a heterogonous mechanism associated with structural change and chronic inflammation. In addition, often pediatric glaucoma differs greatly in cause and management from the glaucoma developed by adults. Currently, the best sign of pediatric glaucoma is an IOP of 21 mm Hg or greater present within a child. One of the most common causes of pediatric glaucoma is cataract removal surgery, which leads to an incidence rate of about 12.2% among infants and 58.7% among 10-year-olds.

1.1.4 INFECTIONS

The burden of onchocerciasis: children leading blind adults in Africa

Childhood blindness can be caused by conditions related to pregnancy, such as congenital rubella syndrome and retinopathy of prematurity. Leprosy and onchocerciasis each blind approximately 1 million individuals in the developing world.

The number of individuals blind from trachoma has decreased in the past 10 years from 6 million to 1.3 million, putting it in seventh place on the list of causes of blindness worldwide.

Central corneal ulceration is also a significant cause of monocular blindness worldwide, accounting for an estimated 850,000 cases of corneal blindness every year in the Indian subcontinent alone. As a result, corneal scarring from all causes is now the fourth greatest cause of global blindness.

1.1.5 INJURIES

Re-educating wounded. Blind French soldiers learning to make baskets, World War I.

Eye injuries, most often occurring in people under 30, are the leading cause of monocular blindness (vision loss in one eye) throughout the United States. Injuries and cataracts affect the eye itself, while abnormalities such as optic nerve hypoplasia affect the nerve bundle that sends signals from the eye to the back of the brain, which can lead to decreased visual acuity.

Cortical blindness results from injuries to the occipital lobe of the brain that prevent the brain from correctly receiving or interpreting signals from the optic nerve. Symptoms of cortical blindness vary greatly across individuals and may be more severe in periods of exhaustion or stress. It is common for people with cortical blindness to have poorer vision later in the day.

Blinding has been used as an act of vengeance and torture in some instances, to deprive a person of a major sense by which they can navigate or interact within the world, act fully independently, and be aware of events surrounding them. An example from the classical realm is Oedipus, who gouges out his own eyes after realizing that he fulfilled the awful prophecy spoken of him. Having crushed the Bulgarians, the Byzantine Emperor Basil II blinded as many as 15,000 prisoners taken in the battle, before releasing them. Contemporary examples include the addition of methods such as acid throwing as a form of disfigurement.

1.1.6 GENETIC DEFECTS

People with albinism often have vision loss to the extent that many are legally blind, though few of them actually cannot see. Leber congenital Amaurosis can cause total blindness or severe sight loss from birth or early childhood.

Recent advances in mapping of the human genome have identified other genetic causes of low vision or blindness. One such example is Bardet Biedl syndrome.

1.1.7 POISONING

Rarely, blindness is caused by the intake of certain chemicals. A well-known example is methanol, which is only mildly toxic and minimally intoxicating, and breaks down into the substances formaldehyde and formic acid which in turn can cause blindness, an array of other health complications, and death. When competing with ethanol for metabolism, ethanol is metabolized first, and the onset of toxicity is delayed. Methanol is commonly found in methylated spirits, denatured ethyl alcohol, to avoid paying taxes on selling ethanol intended for human consumption. Methylated spirits are sometimes used by alcoholics as a desperate and cheap substitute for regular ethanol alcoholic beverages.

1.1.8 OTHER

- Amblyopia: is a category of vision loss or visual impairment that is caused by factors unrelated to refractive errors or coexisting ocular diseases.] Amblyopia is the condition when a child's visual systems fail to mature normally because the child either suffers from a premature birth, measles, congenital rubella syndrome, vitamin A deficiency, or meningitis. If left untreated during childhood, amblyopia is currently incurable in adulthood because surgical treatment effectiveness changes as a child matures. Consequently, amblyopia is the world's leading cause of child monocular vision loss, which is the damage or loss of vision in one eye. In the best case scenario, which is very rare, properly treated amblyopia patients can regain 20/40 acuity.
- Corneal opacification
- Degenerative myopia
- Diabetic retinopathy: is one of the manifestation microvascular complications of diabetes, which is characterized by blindness or reduced acuity. That is, diabetic retinopathy

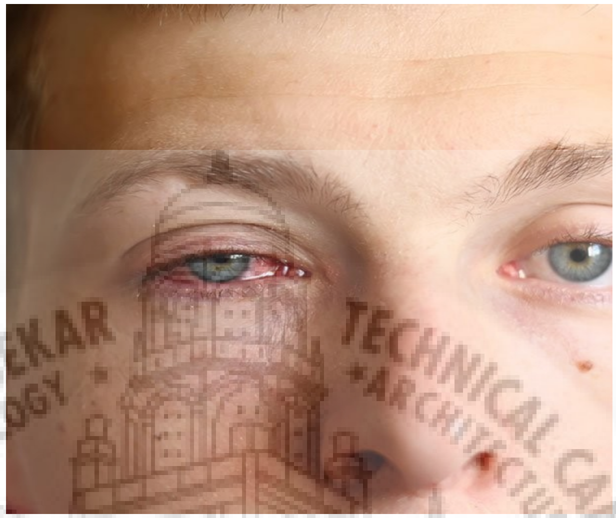
describes the retinal and vitreous hemorrhages or retinal capillary blockage caused by the increase of A1C, which a measurement of blood glucose or sugar level. In fact, as A1C increases, people tend to be at greater risk of developing diabetic retinopathy than developing other microvascular complications associated with diabetes (e.g. chronic hyperglycemia, diabetic neuropathy, and diabetic nephropathy). Despite the fact that only 8% of adults 40 years and older experience vision-threatening diabetic retinopathy (e.g. Non proliferative diabetic retinopathy or NPDR and proliferative diabetic retinopathy or PDR), this eye disease accounted for 17% of cases of blindness in 2002.

1.1.9BLINDNESS FACTS:

Eye infectious causes in underdeveloped areas of the world include trachoma, onchocerciasis (river blindness), and leprosy.

- Blindness is strictly defined as the state of being totally sightless in both eyes. A completely blind individual is unable to see at all. The word blindness , however, is commonly used as a relative term to signify visual impairment, or low vision, meaning that even with eyeglasses, contact lenses, medicine or surgery, a person does not see well. Vision impairment can range from mild to severe.

- Worldwide, between 300 million and 400 million people are visually impaired due to various causes. Of this group, approximately 50 million people are totally blind, unable to see light in either eye. Eighty percent of blindness occurs in people over 50 years old.



- Common causes of blindness include diabetes, macular degeneration, traumatic injuries, infections of the cornea or retina, glaucoma, and inability to obtain any glasses.
- Less common causes of blindness include vitamin A deficiency, retinopathy of prematurity, vascular disease involving the retina or optic nerve including stroke ocular inflammatory disease, retinitis pigmentosa, primary or secondary malignancies of the eye, congenital abnormalities, hereditary diseases of the eye, and chemical poisoning from toxic agents such as methanol.
- Temporary blindness differs in causes from permanent blindness.
- The diagnosis of blindness is made by examination of all parts of the eye by an ophthalmologist.
- The universal symptom of blindness or visual impairment is difficulty with seeing. People who lose their vision suddenly, rather than over a period of years, are more symptomatic regarding their visual loss.
- The treatment of blindness depends on the cause of blindness.
- The prognosis for blindness is dependent on its cause.

- Legal blindness is defined by lawmakers in nations or states in order to either limit allowable activities, such as driving, of individuals who are "legally blind" or to provide preferential governmental benefits to those people in the form of special educational services, assistance with daily functions or monetary assistance. It is estimated that approximately 700,000 people in the United States meet the legal definition of blindness.
- In most states in the United States, "legal blindness" is defined as the inability to see at least 20/200 in either eye with best optical correction.
- Between 80%-90% of the blindness in the world is preventable through a combination of education, access to good medical care, and provision of glasses.
- Patients who have untreatable blindness require reorganization of their habits and re-education to allow them to do everyday tasks in different ways. In the United States and most other developed nations, financial assistance through various agencies can pay for the training and support necessary to allow a blind person to function.
- There are countless individuals with blindness, who, despite significant visual handicaps, have had full lives and enriched the lives of those who have had contact with them.

WHAT IS ARTIFICIAL INTELLIGENCE?

Artificial intelligence (**AI**) is intelligence demonstrated by machines, unlike the **natural intelligence** displayed by humans and animals, which involves consciousness and emotionality. The distinction between the former and the latter categories is often revealed by the acronym chosen. 'Strong' AI is usually labelled as artificial general intelligence (AGI) while attempts to emulate 'natural' intelligence have been called artificial biological intelligence (ABI). Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. A quip in Tesler's Theorem says "AI is whatever hasn't been done yet." For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology. Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), [and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations.

Artificial intelligence was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding. After AlphaGo successfully defeated a professional Go player in 2015, artificial intelligence once again attracted widespread global attention. For most of its history, AI research has been divided into sub-fields that often fail to communicate with each other. These sub-fields are based on technical considerations, such as particular goals (e.g. "robotics" or "machine learning"), the use of particular tools ("logic" or artificial neural networks), or deep philosophical differences. Sub-fields have also been based on social factors (particular institutions or the work of particular researchers).

The traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects. AGI is among the field's long-term goals. Approaches include statistical methods, computational intelligence, and traditional symbolic AI. Many tools are used in AI, including versions of search and mathematical optimization, artificial neural networks, and methods based on statistics, probability and economics. The AI field draws upon computer science, information engineering, mathematics, psychology, linguistics, philosophy, and many other fields.

The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence. These issues have been explored by myth, fiction and philosophy since antiquity. Some people also consider AI to be a danger to humanity if it progresses unabated. Others believe that AI, unlike previous technological revolutions, will create a risk of mass unemployment.

In the twenty-first century, AI techniques have experienced a resurgence following concurrent advances in computer power, large amounts of data, and theoretical understanding; and AI techniques have become an essential part of the technology industry, helping to solve many challenging problems in computer science, software engineering and operations research.

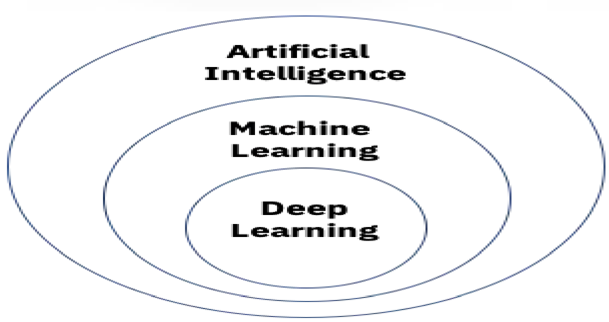
TYPES OF ARTIFICIAL INTELLIGENCE -WEAK AI VS. STRONG AI

Weak AI—also called Narrow AI or Artificial Narrow Intelligence (ANI)—is AI trained and focused to perform specific tasks. Weak AI drives most of the AI that surrounds us today. 'Narrow' might be a more accurate descriptor for this type of AI as it is anything but weak; it enables some very robust applications, such as Apple's Siri, Amazon's Alexa, IBM Watson, and autonomous vehicles.

Strong AI is made up of Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI). Artificial general intelligence (AGI), or general AI, is a theoretical form of AI where a machine would have an intelligence equaled to humans; it would have a self-aware consciousness that has the ability to solve problems, learn, and plan for the future. Artificial Super Intelligence (ASI)—also known as superintelligence—would surpass the intelligence and ability of the human brain. While strong AI is still entirely theoretical with no practical examples in use today, that doesn't mean AI researchers aren't also exploring its development. In the meantime, the best examples of ASI might be from science fiction, such as HAL, the superhuman, rogue computer assistant in *2001: A Space Odyssey.*

1.1.2 DEEP LEARNING VS. MACHINE LEARNING

Since deep learning and machine learning tend to be used interchangeably, it's worth noting the nuances between the two. As mentioned above, both deep learning and machine learning are sub-fields of artificial intelligence, and deep learning is actually a sub-field of machine learning.

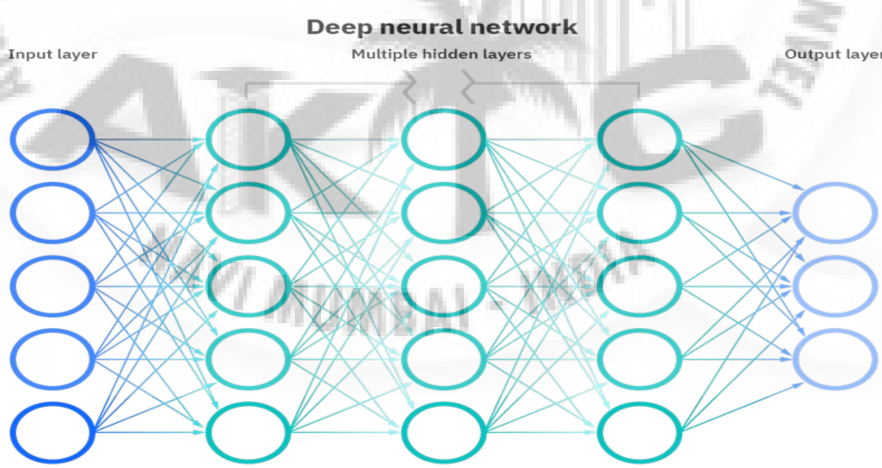


SUBSET OF AI

Deep learning is actually comprised of neural networks. “Deep” in deep learning refers to a neural network comprised of more than three layers—which would be inclusive of the inputs and the output—can be considered a deep learning algorithm.

The way in which deep learning and machine learning differ is in how each algorithm learns. Deep learning automates much of the feature extraction piece of the process, eliminating some of the manual human intervention required and enabling the use of larger data sets. You can think of deep learning as "scalable machine learning" as Lex Fridman noted in same MIT lecture from above. Classical, or "non-deep", machine learning is more dependent on human intervention to learn. Human experts determine the hierarchy of features to understand the differences between data inputs, usually requiring more structured data to learn.

"Deep" machine learning can leverage labeled datasets, also known as supervised learning, to inform its algorithm, but it doesn’t necessarily require a labeled dataset. It can ingest unstructured data in its raw form (e.g. text, images), and it can automatically determine the hierarchy of features which distinguish different categories of data from one another. Unlike machine learning, it doesn't require human intervention to process data, allowing us to scale machine learning in more interesting.



NEURAL NETWORK

ARTIFICIAL INTELLIGENCE APPLICATIONS

There are numerous, real-world applications of AI systems today. Below are some of the most common examples:

- **Speech Recognition:** It is also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, and it is a capability which uses natural language processing (NLP) to process human speech into a written format. Many mobile devices incorporate speech recognition into their systems to conduct voice search—e.g. Siri—or provide more accessibility around texting.

- **Customer Service:** Online chatbots are replacing human agents along the customer journey. They answer frequently asked questions (FAQs) around topics, like shipping, or provide personalized advice, cross-selling products or suggesting sizes for users, changing the way we think about customer engagement across websites and social media platforms. Examples include messaging bots on e-commerce sites with virtual agents, messaging apps, such as Slack and Facebook Messenger, and tasks usually done by virtual assistants and voice assistants.

- **Computer Vision:** This AI technology enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks. Powered by convolutional neural networks, computer vision has applications within photo tagging in social media, radiology imaging in healthcare, and self-driving cars within the automotive industry.

- **Recommendation Engines:** Using past consumption behavior data, AI algorithms can help to discover data trends that can be used to develop more effective cross-selling strategies. This is used to make relevant add-on recommendations to customers during the checkout process for online retailers.

- **Automated stock trading:** Designed to optimize stock portfolios, AI-driven high-frequency trading platforms make thousands or even millions of trades per day without human intervention.

HISTORY OF ARTIFICIAL INTELLIGENCE: KEY DATES AND NAMES

The idea of 'a machine that thinks' dates back to ancient Greece. But since the advent of electronic computing (and relative to some of the topics discussed in this article) important events and milestones in the evolution of artificial intelligence include the following:

- **1950:** Alan Turing publishes *Computing Machinery and Intelligence.* In the paper, Turing —famous for breaking the Nazi's ENIGMA code during WWII—proposes to answer the question 'can machines think?' and introduces the Turing Test to determine if a computer can demonstrate the same intelligence (or the results of the same intelligence) as a human. The value of the Turing test has been debated ever since.

- **1956:** John McCarthy coins the term 'artificial intelligence' at the first-ever AI conference at Dartmouth College. (McCarthy would go on to invent the Lisp language.) Later that year, Allen Newell, J.C. Shaw, and Herbert Simon create the Logic Theorist, the first-ever running AI software program.

- **1967:** Frank Rosenblatt builds the Mark 1 Perceptron, the first computer based on a neural network that 'learned' though trial and error. Just a year later, Marvin Minsky and Seymour Papert publish a book titled *Perceptrons*, which becomes both the landmark work on neural networks and, at least for a while, an argument against future neural network research projects.

- **1980s:** Neural networks which use a backpropagation algorithm to train itself become widely used in AI applications.

- **1997:** IBM's Deep Blue beats then world chess champion Garry Kasparov, in a chess match (and rematch).

- **2011:** IBM Watson beats champions Ken Jennings and Brad Rutter at *Jeopardy!*

- **2015:** Baidu's Minwa supercomputer uses a special kind of deep neural network called a convolutional neural network to identify and categorize images with a higher rate of accuracy than the average human.
- **2016:** DeepMind's AlphaGo program, powered by a deep neural network, beats Lee Sodol, the world champion Go player, in a five-game match. The victory is significant given the huge number of possible moves as the game progresses (over 14.5 trillion after just four moves!). Later, Google purchased DeepMind for a reported $400 million.

ARTIFICIAL INTELLIGENCE AND VISUAL LOSS:

Artificial intelligence is the study and creation of computer systems that can perceive, reason and act. The goal is to develop an intelligent machine. The intelligence must be represented by learning, thinking, making decision, solving problems. AI is a collaborative field which requires knowledge in all fields such as psychology, computer science, engineering, logic, mathematics, ethics and so on. It is a new field which is expanding meteorically. AI can store large amounts of information and can process at high speed. It is a representation or duplication of human intelligence, AI systems are supposed to learn from the previous experience and self-correct through deep learning. As there is an increase in the growth of technology, these intelligent systems are also useful for the humans in the everyday life. It can be our companion, do our daily task, understand our emotions and help accordingly. The vision impaired individual find it difficult to self-navigate outdoor in the well known environment and even simply walking down the crowded street. They are unable to judge what's happening around them. Many blind individuals rely on their dogs to get around from place to place since the dogs have the ability to sense danger to act accordingly and guide them too, unfortunately the dog itself gets attacked brutally. Humans mostly communicate through non-verbal activities such as facial expressions, gestures, body language and tone of voice, to communicate their emotions. Likewise, there are hundreds of thousands of emotions. To collect all these emotion we use big data as normal storage device will not be able to handle so much data. And when the machines find something new it needs to store that data too. Emotional intelligence (EI) is the area of cognitive ability that facilitates interpersonal behavior. Five components of emotional intelligence are selfawareness, self, motivation, empathy and social skills. Emotional intelligence (EI), also known as Emotional quotient (EQ) and Emotional Intelligence Quotient (EIQ), is the capability of individuals to recognize their own emotions and those of others, discern between different feelings and label them appropriately, use emotional information to guide thinking and behavior , and manage and/or adjust emotions to adapt to environments or achieve one's goal. Although the term first appeared in a 1964 paper by Michael Beldoch, it gained popularity in the 1995 book by that title, written by the author, and science journalist Daniel Goleman. Since this time, Goleman's 1995 analysis of EI has been criticized within the scientific community despite prolific reports of its usefulness in the popular press.

There are currently several models of EI. Studies have shown that people with high EI have greater mental health job performance, and leadership skills although no causal relationships have been shown and such findings are likely to be attributable to general intelligence and specific personality traits rather than emotional intelligence as a construct. For example, Goleman indicated that EI accounted for 67% of the abilities deemed necessary for superior performance in leaders, and mattered twice as much as technical expertise or IQ. Other research finds that the effect of EI on leadership and managerial performance is non-significant when ability and personality are controlled for, and that general intelligence correlates very closely with leadership.

As the blind people are more vulnerable to the attackers than the sighted people, an assistant is required when they want to go outdoor, as we have discussed earlier it seems quite challenging to take an animal assistance as there is a high chance of the animal to be harm also. So it is better to use technology as it can protect the user and does not get hurt. When people display a facial expression of emotion, we make judgments not only about their affective state, but also about their behavioural tendencies and traits. For example, when people display happy faces we perceive them a shaving traits associated with high affiliation and high dominance. When they display angry faces we perceive them as having traits associated with low affiliation and high dominance. Facial expressions of sadness and fear elicit impressions of traits associated with moderate affiliation and low dominance. The prototype is inbuilt with a camera, microphone and a speaker. It is a small device which can fit into the user's shirt. When the user is walking down the street, the device detects each and every person who crosses nearby. Depending on the facial expression the device will be able to detect if the nearby person is about to attack or not. How the device is able to identify the attacker is that, data of different facial expressions will be stored in thedevice, according to the matching expression the device will be able to detect who is innocent and who is not. Some of the AI technologies used are image matching, image recognition, voice recognition, voice output, speech recognition, face recognition. Can AI read exact the features and distinguish an attacker and non-attacker? Well, the device may not be able to give 100 per cent accuracy however the registered features in the system may match much up to most of the people like discussed in [1]. They are few characteristics that can distinguish from a neutral face to the aggressive face. People with neutral. How the test be carried out by assembling some people who can express their emotions very well. After they have been assembled, test may be conducted. So when the person frowns and expresses few gestures then he may be assumed as an attacker. And the necessary will be given to blind person. Image Recognition It is the ability of a software to identify objects, place, person, writing and actions in an image. Computers use technologies in combination of camera and artificial intelligence software to achieve image recognition. The image recognition can be helpful to recognize the nearby persons face .Image Matching Image Matching is a technique used to match the feature of the object seen through the camera and the registered image in the system. For example, if the camera in the device recognizes the persons face or features, the system checks the available features to determine if he is the attacker or an innocent person. Voice Recognition/Voice Matching It is an ability of the system to determine recognize the tone of the voice. The device inputs the given voice and then it matches its voice with the modulation of the speaker if the speaker is rude or normal. This lets the system to know how the speaker is speaking. And the system warns the user that the speaker is rude. Big data is used to store huge amount of image and voice output and voice recognitions. It is often characterized by 3V's, the extreme volume of data, the wide variety of data types and the velocity at which the data must be processed. Although big data doesn't quite equate to any specific volume of data, the term is often used to describe terabytes, petabytes and even Exa bytes of data. Integration the deep learning, emotional intelligence and machine learning techniques it is possible to acquire a system that can distinguish the emotions of every person from other.

1.2 LITERATURE REVIEW:

A.Nivedita, M.Sindhuja, G.Asha, R.S.Subasree, S.Monisha, they proposed. The smart walking stick helps the blind people to perform navigation and to do their work easily and comfortably. In normal stick, the detection of the obstacle is not done and normal stick is not efficient for visually impaired person, because the blind person does not know what type of things or what type of the objects come in front of them. The person cannot recognize what is the size of that object and how far it is from the object. It is difficult for the blind person to move here and there. In smart walking stick, ultrasonic sensor is used to detect the obstacle. When the obstacle becomes detected camera becomes on and the object is detected with the help of a camera. If any obstacle comes in front of blind person, they can know about the obstacle by hearing the sound generated by the microphone. This is implemented using RaspBerry Pi 3 Model B.LDR sensor is used to detect the brightness of the environment i.e to detect whether the environment is bright or dark. The system is very useful for people who are visually impaired and are often need help from others.

R.Dhanuja1, F.Farhana2, G.Savitha3 ultrasonic blind walking stick with the use of arduino. According to WHO, 30 million peoples are permanently blind and 285 billion peoples with vision impairment. If u notice them , you can very well know about it they can't walk without the help of other. One has to ask guidance to reach their destination. They have to face more struggles in their life daily life. Using this blind stick, a person can walk more confidently. This stick detects the object in front of the person and give response to the user either by vibrating or through command. So, the person can walk without any fear. This device will be best solution to overcome their difficulties.

Muriel Pinto, **Rose Denzil Stanley**, **Sheetal Malagi**, **Veena Parvathi K.**, **Ajithanjaya Kumar M. K.**
Currently, visually impaired people use a traditional cane as a tool for directing them when they move from one place to another. Although, the traditional cane is the most widespread means that is used today by the visually impaired people, it could not help them to detect dangers from all levels of obstacles. In this context, we propose a new intelligent system for guiding individuals who are visually impaired or partially sighted. The system is used to enable visually impaired people to move with the same ease and confidence as a sighted people. Also the system helps in detecting the potholes. The system is linked with a GSM-GPS module to pin-point the location of the visually impaired person and to establish a two way communication path in a wireless fashion. Moreover, it provides the direction information as well as information to avoid obstacles based on ultrasonic sensors. A buzzer and vibrator motor are also added to the system. The whole system is designed to be small, light and is used in conjunction with the white cane. The results have shown that the blinds that used this system could move independently and safely.

1.3 CONCEPT KNOWLEDGE

1.3.1 Lidar 360

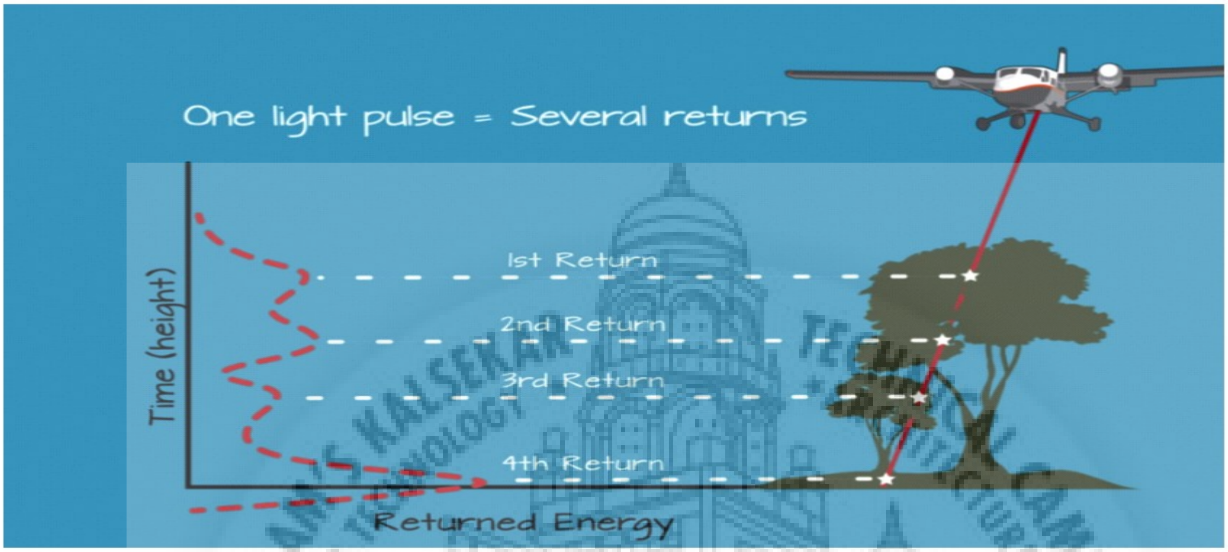
LiDAR, or "light detection and ranging," is a remote-sensing method that uses light in laser form to measure distances or ranges. LiDAR has been used for decades although there are more recent advancements in the technology.

These include laser scanning and processing speeds that make 3D mapping possible and accessible. This technique uses light (precise laser pulses) to map an underlying object or the terrain (the earth's surface). The laser is pointed on an object's surface or the terrain and, after hitting the surface, reflects back.

Mapping and 3D drawings of an object's surface or terrain is quite simple. Every reflection of light is composed of thousands of individual beams. These beams, or patterns, vary in distance — each one is nearly 10 cm apart. A scanner picks these up.

The time and distance it takes for these beams to occur are recorded by the scanner. Additionally, the distance between the object and the light source is calculated by using the recorded time and speed of light.

If we place dots representing each beam at its distance and then view these dots from the opposite axis, it's possible to see a 3D image of the ground. Let's check out this phenomenon in an image.



Instead of taking a step-and-read approach, the RPLIDAR drives the rotating scanner with a DC motor, continuously taking readings and each possible angle, but over several rotations a full scan can be assembled.

The animation below shows a different mechanical approach (using a rotating mirror instead of a rotating sensor) and only covers 180 degrees, but the concept is identical. It shows the general theory of operation: readings taken at different angles, combined to provide a map of distances from the sensor. Using this data, a system can get a sense of the space around it.

1.3.1 a. CPython on Raspberry Pi

The simplest way to use the RPLIDAR with a Raspberry Pi is to interface through the bundled USB adapter. Connect a USB cable on the Raspberry Pi and then plug the other end into the LIDAR USB adapter board. Ensure the adapter board is fitted correctly onto the sensor per the instructions in the manual.

1.3.1 b. PyGame

We'll be using the 2.8" PiTFT display. See this guide for instructions on installing and configuring the required support software. We'll also need pyGame to provide that data plotting capabilities. See the pyGame site for installation instructions.

To use the RPLIDAR, you need to start the motor spinning and tell it to start taking readings. It will then stream each reading it takes until you tell it to stop. Each reading includes the distance sensed, and the angle of the reading. To make things easier, the Skoltech library provides a convenient wrapper for this functionality, and that's what we'll use for this guide.

Next, we create a buffer in which to keep distance data. Each item in the list stores the distance measured at each degree for a complete rotation. We use this to maintain the most recent set of measurements even though each scan will be incomplete.

We use the interscan function of the RPLIDAR object. This starts the motor and the scanning for us so we can skip that in our code. Interscan accumulate measurements for a single rotation and returns a list of tuples, from which we are interested in the second and third items. These are the angle and the distance measured at that angle. Both are floating point values.

Once we have a scan, we step through each data point. The floor of the angle is taken and used as the index at which to store the measurement value. Because it is possible that multiple consecutive measurements in a scan could be close to the same angle (and thus having the same floor value) there is a small chance that some measurements are lost. But there is no real disadvantage to this, so it can be disregarded.

The process_data function can do anything from finding the closest object, to choosing the best direction to move in. The only requirement is that it be as fast as possible. If it takes too long to process a scan, data from the RPLIDAR will eventually be dropped. Since we're constantly reading fresh data, dropping an occasional scan shouldn't be an issue.

In this example we display the distance data on a PiTFT.

The first step of a pass is to clear the display. Then it looks at each angle: 0-359. If the distance recorded for that angle is zero, it's because a reading hasn't been acquired for that angle yet, and we can safely ignore it. If there is a distance recorded for the angle, max_distance is adjusted as necessary to keep the display scaled to fit on the screen. The angle is converted to radians (from degrees) and the vector for the reading is converted to a cartesian coordinate which is then used to plot the data point on the display.

After all measurements have been plotted, the display is updated.There are several possible ways forward. Continuing on the Pi, different applications can be explored using the LIDAR data. Robot navigation would be a fun one.

In a different direction, the library can be ported to CircuitPython and run on SAMD boards. Look forward to seeing that added to this guide in the near future.

LiDAR360 is the flagship LiDAR post-processing software, which provides comprehensive LiDAR data management and analysis functions. It has over 10 peer-reviewed point cloud processing algorithms which enables it to process more than 300 GB of LiDAR data simultaneously. The software platform also includes versatile editing tools and automatic strip adjustment for increased workflow productivity in applications of terrain, forestry and power line surveying(See the LiPowerline software).

The Terrain module is a suite of GIS tools used primarily for the generation of industry-standard topographic products. Our point cloud filtering algorithm can precisely extract ground points under complex landscapes and therefore improve the terrain surveying accuracy. The module also generates more advanced topographic products, such as TDOM, through the fusion of LiDAR and photogrammetry.

The Forestry module brings important technological innovations to forest inventory and analysis. It provides a unique toolset for manipulating point cloud collected from aerial and terrestrial LiDAR scanners. Defining individual tree level parameters, such as tree height, DBH, LAI, and crown diameter, are made possible through our segmentation algorithms. A range of regression models for predicting forest structures from LiDAR variables are also provided.

1.3.1 c. SPECIFICALLY, LIDAR 360 HAS THE FOLLOWING MODULES:

Strip Alignment: Point clouds from overlapped strips can be automatically aligned based on strict geometric model. The software can display aligned strips in real-time and generate aligned point clouds with high accuracy. The module also provides a series of tools to check and analyze data quality.

Data Management: The module provides management tools for point cloud and raster data, which include format conversion, point cloud de-noising, normalization, raster band calculation and so on.

Statistics: Based on the number, density and elevation of LiDAR points, statistics can be calculated for data quality evaluation.

Classify: The module provides various classification methods for point cloud classification including ground classification, ground key point classification, interactive classification, machine-learning-based classification (for building, vegetation, and custom classes) etc.

Terrain: The module can generate DEM (Digital Elevation Model), DSM(Digital Surface Model), CHM(Canopy Height Model) for geospatial applications. Other related topographic products such as Hillshade, Slope, Aspect, Roughness and Contour are also supported.

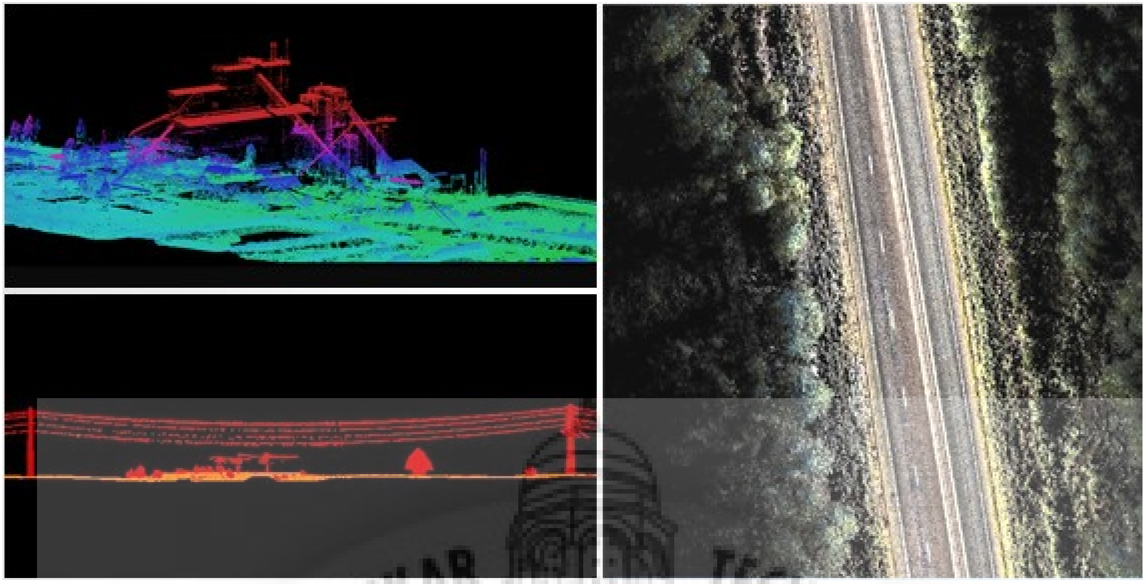
Vector Editor: The module provides vectorization functions for DLG (Digital Line Graphics) workflow. The software displays point cloud as a high-contrast base map, in which contours of objects (e.g., building, vegetation, roads, street lamps, water, bridges) can be clearly identified and vectorized.

ALS Forest: Based on ALS(Airborne Laser Scanning) data, the module can extract a series of forestry parameters (Elevation Metrics, Leaf Area Index, Canopy Cover, etc.), segment individual trees and calculate parameters of individual tree(position, height, canopy, etc.). Moreover, various regression functions are also provided which can be used in combination with field measurement to retrieve forest parameters that cannot be directly derived from ALS LiDAR data (such as biomass or stem volume).

TLS Forest: Based on TLS (Terrestrial Laser Scanning) data, the module can estimate the number of trees, extract DBH (Diameter at Breast Height) with batch processing, segment individual tree, and calculate parameters of individual tree(position, height, etc.).

Geology: Extract terrain features, geological structure surface features, etc. based on airborne lidar point cloud data.

Powerline: Obtaining a clearance distance analysis report based on airborne LiDAR point cloud data, including towers positioning, data classification, and danger point detection.



1.3.2 RASBERRY PI :

Now over four years old, the Raspberry Pi, a cheap credit-card sized computer, has taken the computing and DIY world by storm. Read on as we guide you through everything from buying to powering to running the tiny dynamo.

1.3.2 a. WHAT IS THE RASPBERRY PI?

The Raspberry Pi is a credit-card sized computer designed and manufactured by the Raspberry Pi Foundation, a non-profit organization dedicated to making computers and programming instruction as accessible as possible to the widest number of people.

Although the original mission of the Raspberry Pi project was to get inexpensive computers with programming capabilities into the hands of students, the Pi has been embraced by a diverse audience. Tinkers, programmers, and DIYers across the globe have adopted the tiny platform for projects ranging from recreating retro arcade cabinets to controlling robots to setting up cheap but powerful home media devices.

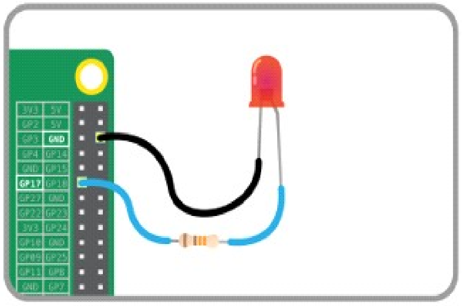
Introduced in 2012, the original Raspberry Pi (what we now refer to as the Raspberry Pi 1 Model A) featured a system-on-a-chip setup built around the Broadcom BCM2835 processor–a tiny but fairly powerful mobile processor commonly used in cellphones. It included a CPU, GPU, audio/video processing, and other functionality all on a low-power chip paired with a 700Mhz single core ARM processor. Over the intervening years the foundation has released multiple revisions (switching out the Broadcom chips for improved versions and upping the CPU power with a 1.2GHz quad-core chip).

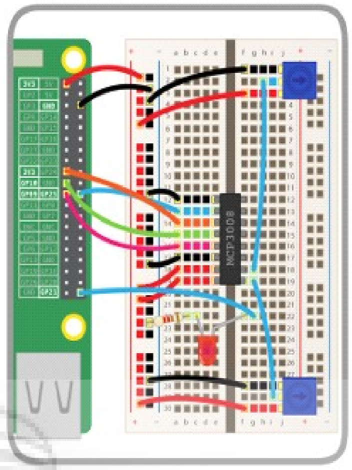
Although the Pi is an amazing little device that has grown by leaps and bounds since its introduction, it's important to emphasize what the Raspberry Pi is not. The Raspberry Pi is not an outright replacement for your desktop computer or laptop. You cannot run Windows on it (at least not the traditional version of Windows you know), although you can run many distributions of Linux—including distributions with desktop environments, web browsers, and other elements you would expect in a desktop computer.

The Raspberry Pi is, however, an astoundingly versatile device that packs a lot of hardware into a very inexpensive body and is perfect for hobby electronics, DIY projects, setting up an inexpensive computer forprogramming lessons and experiments, and other endeavors.

All over the world, people use the Raspberry Pi to learn programming skills, build hardware projects, do home automation, implement Kubernetes clusters and Edge computing, and even use them in industrial applications.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).





1.3.2. b. WHAT'S ON THE RASPBERRY PI BOARD?

In the early years of the Pi foundation, the Raspberry Pi came in two versions at two different price points: the Model A ($25) and Model B ($35). If you needed less hardware (the Model A had one less USB port, no Ethernet port, and half the RAM) you could save ten bucks.

As manufacturing costs fell and the Pi gained more and more attention, they were able to increase the hardware specs of the device significantly while keep the cost the same–eventually unifying all models with the introduction of the Raspberry Pi 2 in 2015 and the Raspberry Pi 3 in 2016. Put another way: the best Raspberry Pi available at any given time has always cost $35. So what do you get when you buy a $35 credit card sized computer?

The current generation Raspberry Pi 3, seen above, sports the following hardware:

1.2 Ghz ARM processor Systems-On-a-Chip (SoC) with integrated 1GB RAM.

1 HDMI port for digital audio/video output

1 3.5mm jack that offers both audio and composite video out (when paired with an appropriate cable).

4 USB 2.0 ports for connecting input devices and peripheral add-ons.

1 microSD card reader for loading the operating system.

1 Ethernet LAN port.

1 Integrated Wi-Fi/Bluetooth radio antenna.

1 microUSB power port.

1 GPIO (General Purpose Input/Output) interface.

What is a GPIO? The Raspberry Pi comes with a set of 26 exposed vertical pins on the board. These pins are a General Purpose Input/Output interface that is purposely not linked to any specific native function on the Raspberry Pi board.

Instead, the GPIO pins are there explicitly for the end user to have low-level hardware access directly to the board for the purposes of attaching other hardware boards, peripherals, LCD display screens, and other hardware devices to the Pi. For example, if you wanted to take an old arcade controller and wire it directly to your Raspberry Pi to give your arcade a more authentic feel, you could do so using the GPIO interface.

Although we will not be using the GPIO header in today's "getting started" tutorial, we do take advantage of it in other tutorials, like our Raspberry Pi LED indicator build that uses an LED breakout board attached to the GPIO header.

1.3.2 c.WHAT RASPBERRY PI MODELS HAVE BEEN RELEASED?

There have been many generations of the Raspberry Pi line: from Pi 1 to 4, and even a Pi 400. There has generally been a Model A and a Model B of most generations. Model A has been a less expensive variant, and tends to have reduced RAM and fewer ports (such as USB and Ethernet). The Pi Zero is a spinoff of the original (Pi 1) generation, made even smaller and cheaper. Here's the lineup so far:

Pi 1 Model B (2012)

Pi 1 Model A (2013)

Pi 1 Model B+ (2014)

Pi 1 Model A+ (2014)

Pi 2 Model B (2015)

Pi Zero (2015)

Pi 3 Model B (2016)

Pi Zero W (2017)

Pi 3 Model B+ (2018)

Pi 3 Model A+ (2019)

Pi 4 Model A (2019)

Pi 4 Model B (2020)

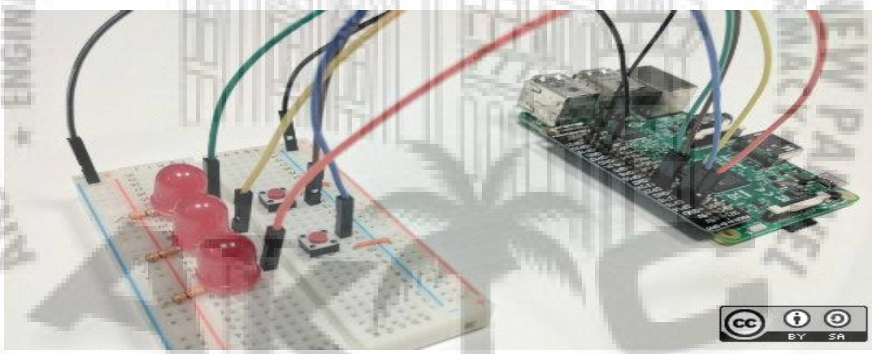
Pi 400 (2021)

1.3.2 d. WHAT'S THE RASPBERRY PI FOUNDATION?

The Raspberry Pi Foundation works to put the power of computing and digital making into the hands of people all over the world. It does this by providing low-cost, high-performance computers that people use to learn, solve problems, and have fun. It provides outreach and education to help more people access computing and digital making—it develops free resources to help people learn about computing and making things with computers and also trains educators who can guide other people to learn.

Code Club and CoderDojo are part of the Raspberry Pi Foundation, although these programs are platform-independent (they're not tied to Raspberry Pi hardware). The Raspberry Pi Foundation promotes these clubs and helps grow the network around the world in order to ensure every child has access to learning about computing. Similarly, Raspberry Jams are Raspberry Pi-focused events for people of all ages to come together to learn about Raspberry Pi and share ideas and projects.

The physical computing capabilities of the Raspberry Pi



While the Raspberry Pi is an excellent and affordable mini Linux computer with a stylish and functional desktop user interface, it has plenty of scope beyond that of a regular PC. Here's an overview of the physical computing capabilities of the Pi.

1.3.2 e.GPIO PINS

Since 2014, with the release of the Model B+, the Raspberry Pi's form factor has stayed the same, including the uniform 40-pin GPIO (General Purpose Input/Output) pin header.

These pins allow you to connect electronic components and program physical devices in the real world, such as inputs like sensors and outputs like lights. The pins include:

3V3 (a constant supply of 3.3 volts)

5V (a constant supply of 5 volts)

GND (ground pins, 0 volts)

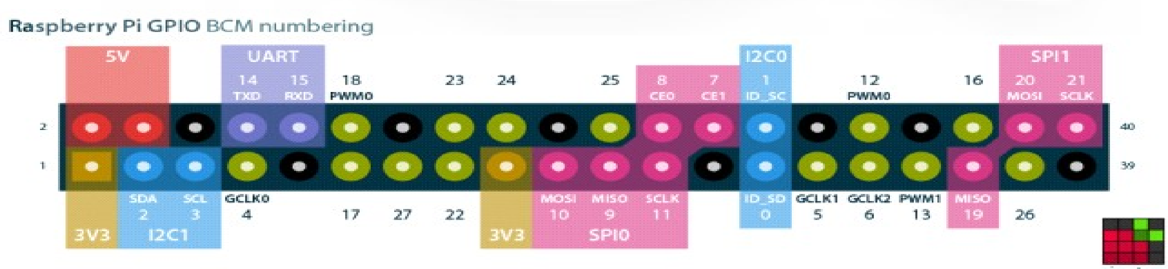
GPIO (general purpose input/output pins—these are essentially variable 3V3 pins controlled by the user)

SPI (Serial Peripheral Interface)

I2C (Inter-integrated Circuit)

UART (Universal Asynchronous Receiver/Transmitter)

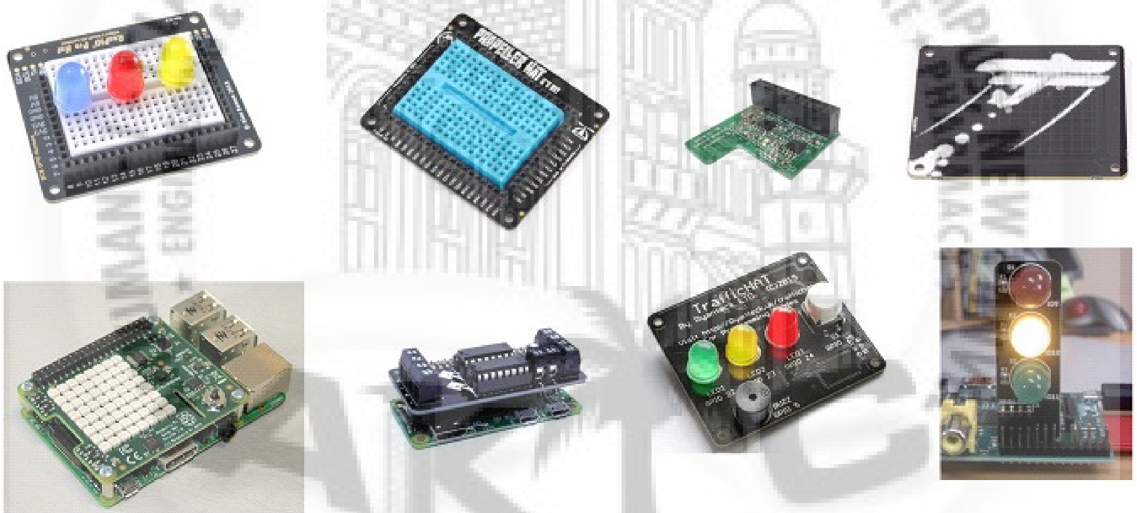
The layout of the pins is the same on all models since the B+, which includes the Pi 2, Pi 3, and Pi Zero



1.3.2 f. ADD-ON BOARDS / HATS:

You can connect simple components directly to the pins using jumper wires, or you can use a breadboard to hold everything in place and allow components to share use of some pins.

Alternatively, you can use add-on boards that provide embedded components on a PCB (printed circuit board), which sits on top of the Pi's GPIO pins. Soon after the B+ launch, the Raspberry Pi Foundation published the specification for a HAT (Hardware Attached on Top) standard, which determines the dimensions and other requirements for what can be considered a HAT. HATs are add-on boards that fit neatly on top of the Pi with aligned mounting holes, and some have slots for the camera and display cables. The community of Raspberry Pi accessory retailers have produced a wide range of HATs.



The Raspberry Pi Foundation has produced one official HAT of its own named the Sense HAT, which was made especially for the Astro Pi space mission aboard the International Space Station. It is also available for purchase. It encompasses a LED grid, a mini joystick, and a series of sensors. Read more about programming the Sense HAT.

HATs are a great way of extending the capabilities of your Raspberry Pi to use in a project without having to wire up or solder components.

1.3.2 g. GPIO WITH PYTHON

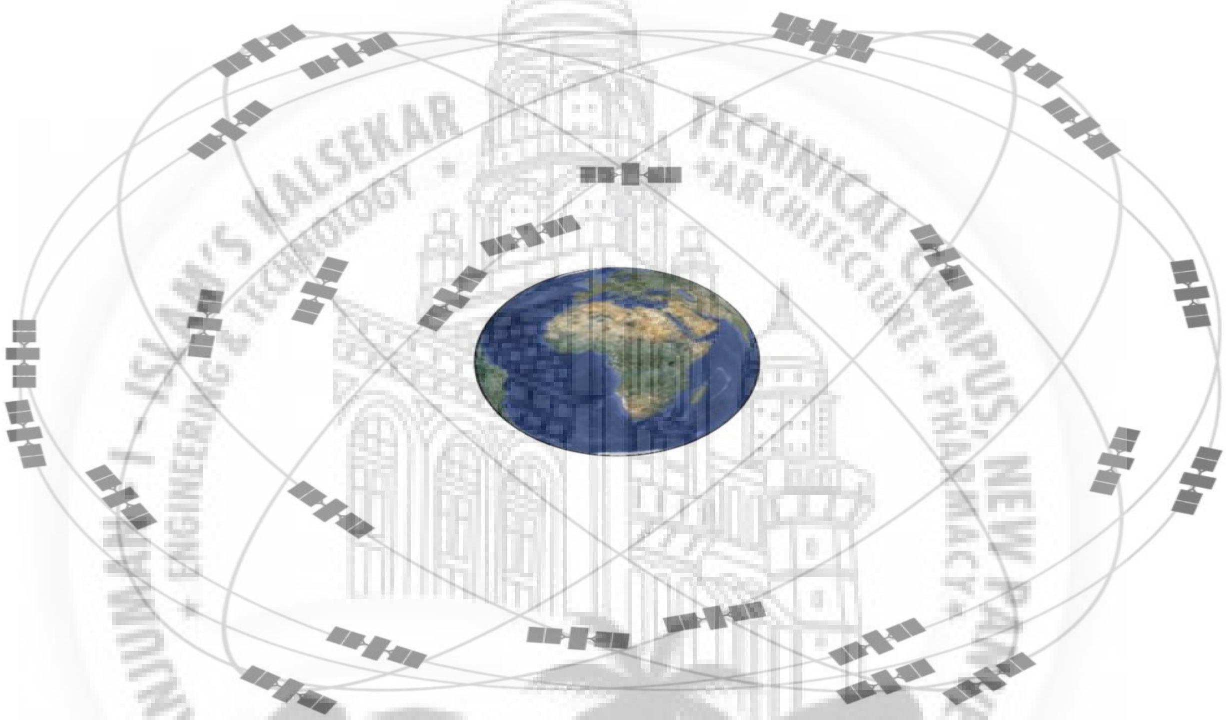
It's possible to control the GPIO pins from a wide range of programming languages, but the simplest and most popular way is to use Python. The GPIO Zero library provides a simple interface to GPIO devices and includes support for a range of components and add-on boards. With just a few lines of code you can flash an LED:

```
from gpiozero import LED
from time import sleep
led = LED(17)
while True:
    led.on()
    sleep(1)
led.off()
    sleep(1)
```

1.3.3 GPS MODULE

1.3.3 a. WHAT IS GPS?

The Global Positioning System (GPS) is the most widely used satellite navigation system around the world. It is one of the Global Navigation Satellite Systems (GNSS) that provides geolocation, time, and velocity information. GPS is operational since 1978 and globally available since 1994. The latest GPS receivers provide geolocation with an accuracy of 30 centimetres. The GPS system is owned by the United States government and is maintained by the United States Space Force.



GPS is a network of 30+ Medium Earth Orbit (MEO) satellites. These satellites continuously send signals over dedicated RF frequencies that a GPS receiver can listen to. By calculating its distance from four or more satellites, a GPS receiver can find out its position in latitude and longitude anywhere on earth. There are many kinds of GPS receivers with different capabilities. All GPS receivers essentially are capable of providing their geolocation, UTC time, and velocity information. The latest GPS receivers are ultra-compact and accurate than ever. These tiny devices provide geolocation, time, and velocity data for no cost and are now found in most of the smartphones. GPS receivers are also used in automobiles, for tracking commerce and in several tracking applications like in drones and UAVs.

1.3.4 b. Building blocks of GPS

consists of three segments:

Space segment – This segment consists of GPS navigation satellites. There are 30+ GPS satellites deployed on six orbits around the earth at an altitude of approximately 20,000 Km. These satellites revolve around the earth at an interval of 12 hours, such that they provide coverage all around the world all the time.

Control segment – This segment consists of Ground Control Stations. These stations are responsible for monitoring, controlling, and maintaining orbits of GPS satellites. The ground control stations ensure that deviation of satellites from their orbit and their timing are within tolerance level.

User segment – This segment consists of GPS receivers. A GPS receiver listens to RF signals from GPS satellites and calculates its distance with at least four satellites to get its position and time. At any time, at most, 12 GPS satellites are visible in the sky to a GPS receiver. The satellites transmit information over a band of radio frequency which ranges from 1.1 to 1.5 GHz.

1.3.3 c. GPS Positioning

The GPS satellites continuously send different pieces of information over the RF frequency bands. For example, the satellites have extremely accurate atomic clocks installed. A GPS satellite sends the time on its atomic clock and information about its orbital position and arrival times at different points of its orbit. A GPS receiver uses this information to calculate its distance with the satellites. It is possible to find the absolute position of an object if its distance to three fixed points in space is known. Therefore, a GPS receiver can find its geolocation by calculating its distance with three satellites. However, at least 4 GPS satellites are required to find the time and position accurately. The distance with the fourth satellite is used to compensate time error in the clock of the GPS receiver. So, a GPS receiver calculates its geolocation by calculating its distance with three GPS satellites and use distance with other GPS satellites for corrections and accuracy. If at least 4 GPS satellites are visible to a receiver, it is said to be in a lock or fix. By calculating its distance with at least four satellites, a GPS receiver can accurately find its geolocation (in latitude and longitude) and current (UTC) timestamp.

1.3.3 d. GPS Signals

GPS satellites broadcast RF signals in two carrier frequencies; L1 (1,575.42 MHz) and L2 (1,227.60 MHz). The beams that are accessible to the general public are encoded in C/A (Coarse/

Acquisition) code, and the beams that are used only by the US military force are encoded in P (Precise) code.

The C/A code includes identification codes of each satellite and navigation messages. L1 signal from the GPS satellites is phase-modulated in C/A code. The C/A code is a digital signal, in which 0 and 1 are coded by 1023 consecutive patterns of the digital pulse. The navigation message consists of 25 frames, each having five sub frames of 300 bits. Therefore each frame has 1500 bits (0 and 1). The data length of each bit is 20 msec (i.e., Total pulse width of the C/A signal for a bit is 20 msec).

The data of the orbit of each satellite is called the ephemeris. This data is used to generate the precise location of the satellite, which is required for calculating position information by a GPS receiver. The data of orbit of all satellites is called the almanac. It includes coarse orbit and status information for all satellites in the network. This data is used to locate available satellites to a GPS receiver so that it can find out its current time and position on earth.

1.3.3 e. GPS ACCURACY

The accuracy of GPS position depends on several factors. There can be position errors due to change in propagation velocity of GPS signal in the ionosphere; radio reflections in the troposphere; reflections from the ground, buildings, and other structures (called multipath propagation); and noise sources around the GPS receiver itself.

The accuracy of the GPS position is expressed by Dilution of Precision (DoP). DoP indicates the degradation of accuracy in the GPS position. Smaller is the value of DoP; higher is the accuracy of GPS position. The DoP mainly depends upon the position of GPS satellites engaged in tracking the receiver's position. If the tracked GPS satellites are evenly spaced over the sky, the position accuracy will be higher. If the tracked satellites are spaced disproportionately, position accuracy will be lower.

A GPS receiver requires tracking at least four satellites to provide the position and time accurately. The first lock or fix allows the GPS receiver to obtain the almanac information, and therefore it gets to know what other satellites it can listen for. If a higher number of satellites arevisible to a receiver, the signal strength will be high, and position accuracy will be better. At most, 12 satellites can be visible to a GPS receiver at a time.

1.3.3 f. SELECTION OF GPS MODULE:

SIZE

This is something you need to consider if your project is supposed to be pocket-sized. GPS modules are getting ever-smaller (Your tiny, tiny cell phone has one in it!) but remember that in general, the antenna has to shrink to fit the module which will affect things like lock time and accuracy.

UPDATE RATE

The update rate of a GPS module is basically how often it recalculates and reports its position. The standard for most devices is 1Hz (Only once per second). The fact is, unless you're on an airplane or something, you're probably not going fast enough to have changed position significantly in the past second. However, UAVs and other flying or fast vehicles may require faster update rates to stay on track. 5 and even 10Hz update rates are becoming more and more available for cheap. Keep in mind, though, that a fast update rate means that there's more NMEA sentences flying out of the module, some microprocessors will be quickly overwhelmed trying to parse that much data. On the plus side, if you have a module that runs at 5 or 10Hz, it can usually be configured to run at an easier pace.

POWER REQUIREMENTS

If someone asked you to crunch a bunch of numbers that you had to get from satellites in orbit around the Earth and use that information to figure out where you were, you'd flat out refuse. It's a lot of work, and yet that's exactly what these tiny GPS units are doing (multiple times per second!) so they can use a lot of power. On average, around 30mA at 3.3V. Keep in mind, also, that GPS antennas usually enlist the help of an amplifier that draws extra power. If a unit appears to have super-groovy-low power consumption, make sure there's an antenna attached.

NUMBER OF CHANNELS

Even though there are only so many GPS satellites in view at any given time, the number of channels that your module runs will affect your time to first fix. Since the module doesn't know which satellites are in view, the more frequencies that you can check at once, the faster you'll find a fix. After you get a lock, some modules will shut down the extra blocks of channels to save power. If you don't mind waiting a little longer for a lock, 12 or 14 channels will work just fine for tracking.

ANTENNAS

Many modules come with this chunk of something on top of it. What is that? That is a precisely made chunk of ceramic. Each antenna is finely trimmed to pickup the GPS L1 frequency of 1.57542 GHz. Sound expensive? Well, they make a lot of them. There are some other GPS antenna technologies (chip, helical), but they are not as common, a bit more expensive, and require significantly more amplification and filtering.

ACCURACY:

How accurate is GPS? Well it varies a bit, but you can usually find out where you are, anywhere in the world, within 30 seconds, down to +/- 10m. Amazing! I say +/- because it can vary between modules, time of day, clarity of reception, etc. Most modules can get it down to +/-3m, but if you need sub meter or centimeter accuracy, it gets really expensive. I've heard stories of such fabled GPS receivers, but I have never gotten to touch one. Someone please prove us wrong.

Different types of GPS module is available in market as per requirements but the one which is matched as per our requirement is REES52 Raspberry Pi GPS using L80-39 module program.

As we are going to use Raspberry Pi this is best compatible module with Raspberry Pi. Specifications of this module is show below in next slide.

REES52 Raspberry Pi GPS Module Using L80-39 Module Program

1.3.3 g. Description :

Support Raspberry Pi model A, B, A+, B+, Zero, 2, 3 with its' L80-39 GPS chip inside.

Communicates satellite with UART or USB.

CP2102 as USB to UART Bridge chip, stable and faster.

The L80-39 with 66 search channels and 22 simultaneous tracking channels, it acquires and tracks satellites in the shortest time at Outdoor.

PPS output can be used to coordinate the time with satellite.

Internal patch antenna which works quite well when used outdoors SMA connector for external active antenna for when used indoors

Fix status LED blinks to let you know when the GPS has determined the current coordinates

EASY™, advanced GPS technology without external memory

Support time service application which can be achieved by PPS sync NMEA feature

Built-in LNA for better sensitivity

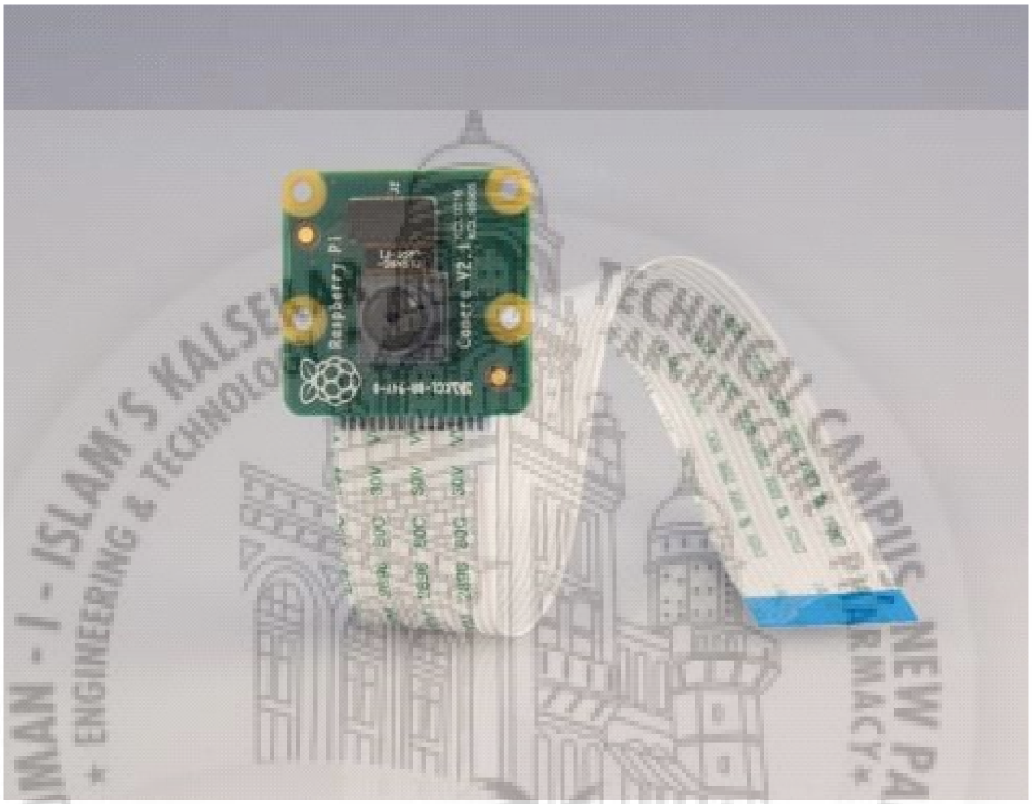
RTC battery-compatible

1x8 male headers, USB micro Interface

includes CP2102 Module Serial Converter USB 2.0 To TTL UART

1.3.4 CAMERA MODULE

The camera module is an official Raspberry Pi accessory and it comes in two variations, a visible light camera and an infrared camera. The current version of the camera (V2) has an 8-megapixel resolution, whereas the original camera was 5 megapixels.



You can control the camera module with the command-line tools raspistill and raspivid or with the Python library picamera:

from picamera import PiCamera

from time import sleep

camera = PiCamera()

camera.start_preview()

sleep(10)

camera.capture('/home/pi/image.jpg')

camera.stop_preview()

1.3.4 WHAT IS PYTHON?

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.

Python consistently ranks as one of the most popular programming languages. As of 2021, Python is the 2nd most popular language just behind JavaScript as of Stack Overflow's 2020 Developer Survey.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

1.3.4 a.APPLICATION OF PYTHON:

There are some application of python:

- Web Development. Python can be used to make web-applications at a rapid rate.
- Game Development. Python is also used in the development of interactive games.
- Machine Learning and Artificial Intelligence.
- Data Science and Data Visualization.
- Desktop GUI.
- Web Scraping Applications.
- Business Applications. .
- CAD Applications.

WEB DEVELOPMENT:

Python can be used to make web-applications at a rapid rate. Why is that? It is because of the frameworks Python uses to create these applications. There is common-backend logic that goes into making these frameworks and a number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more.

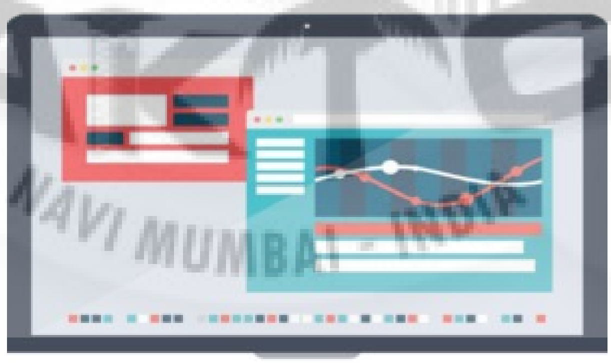
Some of the most well-known frameworks are Django, Flask, Pyramid . Why use a framework? The security, scalability, convenience that they provide is commendable if we compare it to starting the development of a website from scratch.

GAME DEVELOPMENT:

Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3, PyGame which provides functionality and a library for game development. Games such as Civilization-IV, Disney's Toontown Online, Vega Strike etc. have been built using Python.

DESKTOP GUI:

We use Python to program desktop applications. It provides the Tkinter library that can be used to develop user interfaces. There are some other useful toolkits such as the wxWidgets, Kivy, PYQT that can be used to create applications on several platforms.



You can start out with creating simple applications such as Calculators, To-Do apps and go ahead and create much more complicated applications.

WEB SCARPING APPILCATION:

Python is a savior when it comes to pull a large amount of data from websites which can then be helpful in various real-world processes such as price comparison, job listings, research and development and much more.

1.3.5 b. ARTIFICIAL INTELLIGENCE:

Machine Learning and Artificial Intelligence are the talks of the town as they yield the most promising careers for the future. We make the computer learn based on past experiences through the data stored or better yet, create algorithms which makes the computer learn by itself. The programming language that mostly everyone chooses? It's Python. Why? Support for these domains with the libraries that exist already such as Pandas, Scikit-Learn, NumPy and so many more.

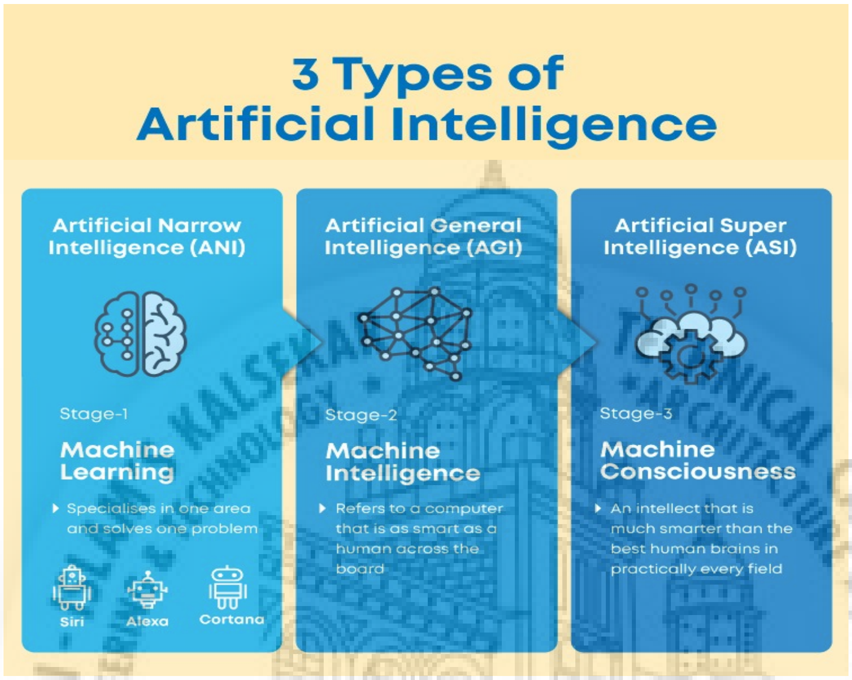
Learn the algorithm, use the library and you have your solution to the problem. It is that simple. But if you want to go the hardcore way, you can design your own code which yields a better solution, which still is much easier when we compare it to other languages.

What are the Types of Artificial Intelligence?

Not all types of AI all the above fields simultaneously. Different Artificial Intelligence entities are built for different purposes, and that's how they vary. AI can be classified based on Type 1 and Type 2 (Based on functionalities). Here's a brief introduction the first type.

3 Types of Artificial Intelligence

- Artificial Narrow Intelligence (ANI)
- Artificial General Intelligence (AGI)
- Artificial Super Intelligence (ASI)



Let's take a detailed look.

What is Artificial Narrow Intelligence (ANI)?

This is the most common form of AI that you'd find in the market now. These Artificial Intelligence systems are designed to solve one single problem and would be able to execute a single task really well. By definition, they have narrow capabilities, like recommending a product for an e-commerce user or predicting the weather. This is the only kind of Artificial Intelligence that exists today. They're able to come close to human functioning in very specific contexts, and even surpass them in many instances, but only excelling in very controlled environments with a limited set of parameters.

What is Artificial General Intelligence (AGI)?

AGI is still a theoretical concept. It's defined as AI which has a human-level of cognitive function, across a wide variety of domains such as language processing, image processing, computational functioning and reasoning and so on.
We're still a long way away from building an AGI system. An AGI system would need to

comprise of thousands of Artificial Narrow Intelligence systems working in tandem, communicating with each other to mimic human reasoning. Even with the most advanced computing systems and infrastructures, such as Fujitsu's K or IBM's Watson, it has taken them 40 minutes to simulate a single second of neuronal activity. This speaks to both the immense complexity and interconnectedness of the human brain, and to the magnitude of the challenge of building an AGI with our current resources.

What is Artificial Super Intelligence (ASI)?

We're almost entering into science-fiction territory here, but ASI is seen as the logical progression from AGI. An Artificial Super Intelligence (ASI) system would be able to surpass all human capabilities. This would include decision making, taking rational decisions, and even includes things like making better art and building emotional relationships.
Once we achieve Artificial General Intelligence, AI systems would rapidly be able to improve their capabilities and advance into realms that we might not even have dreamed of. While the gap between AGI and ASI would be relatively narrow (some say as little as a nanosecond, because that's how fast Artificial Intelligence would learn) the long journey ahead of us towards AGI itself makes this seem like a concept that lays far into the future.

Strong and Weak Artificial Intelligence

Extensive research in Artificial Intelligence also divides it into two more categories, namely Strong Artificial Intelligence and Weak Artificial Intelligence. The terms were coined by John Searle in order to differentiate the performance levels in different kinds of AI machines. Here are some of the core differences between them.

Weak AI	**Strong AI**
It is a narrow application with a limited scope.	It is a wider application with a more vast scope.
This application is good at specific tasks.	This application has an incredible human-level intelligence.
It uses supervised and unsupervised learning to process data.	It uses clustering and association to process data.
Example: Siri, Alexa.	Example: Advanced Robotics

1.3.5 c. What is the Purpose of Artificial Intelligence?

The purpose of Artificial Intelligence is to aid human capabilities and help us make advanced decisions with far-reaching consequences. That's the answer from a technical standpoint. From a philosophical perspective, Artificial Intelligence has the potential to help humans live more meaningful lives devoid of hard labour, and help manage the complex web of interconnected individuals, companies, states and nations to function in a manner that's beneficial to all of humanity.

Currently, the purpose of Artificial Intelligence is shared by all the different tools and techniques that we've invented over the past thousand years – to simplify human effort, and to help us make better decisions. Artificial Intelligence has also been touted as our Final Invention, a creation that would invent ground-breaking tools and services that would exponentially change how we lead our lives, by hopefully removing strife, inequality and human suffering.

That's all in the far future though – we're still a long way from those kinds of outcomes. Currently, Artificial Intelligence is being used mostly by companies to improve their process efficiencies, automate resource-heavy tasks, and to make business predictions based on hard data rather than gut feelings. As all technology that has come before this, the research and development costs need to be subsidised by corporations and government agencies before it becomes accessible to everyday laymen. To learn more about the purpose of artificial intelligence and where it is used, you can take up an AI course and understand the artificial intelligence course details and upskill today.

How Artificial Intelligence (AI) Works?

Building an AI system is a careful process of reverse-engineering human traits and capabilities in a machine, and using it's computational prowess to surpass what we are capable of.

To understand How Artificial Intelligence actually works, one needs to deep dive into the various sub domains of Artificial Intelligence and understand how those domains could be applied into the various fields of the industry. You can also take up an artificial intelligence course that will help you gain a comprehensive understanding.

- **Machine Learning**: ML teaches a machine how to make inferences and decisions based on past experience. It identifies patterns, analyses past data to infer the meaning of these data points to reach a possible conclusion without having to involve human experience.

This automation to reach conclusions by evaluating data, saves a human time for businesses and helps them make a better decision.

- **Deep Learning**: Deep Learning is an ML technique. It teaches a machine to process inputs through layers in order to classify, infer and predict the outcome.
- **Neural Networks**: Neural Networks work on the similar principles as of Human Neural cells. They are a series of algorithms that captures the relationship between various Underying Variabes and processes the data as a human brain does.
- **Natural Language Processing**: NLP is a science of reading, understanding, interpreting a language by a machine. Once a machine understands what the user intends to communicate, it responds accordingly.
- **Computer Vision**: Computer vision algorithms tries to understand an image by breaking down an image and studying different parts of the objects. This helps the machine classify and learn from a set of images, to make a better output decision based on previous observations.
- **Cognitive Computing**: Cognitive computing algorithms try to mimic a human brain by analysing text/speech/images/objects in a manner that a human does and tries to give the desired output.

1.3.5 d. COMPUTER VISION:

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do.

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g. in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that make sense to thought processes and can elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

The scientific discipline of computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, multi-dimensional data from a 3D scanner, or medical scanning device. The technological discipline of computer vision seeks to apply its theories and models to the construction of computer vision systems.

Sub-domains of computer vision include scene reconstruction, object detection, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, visual serving, 3D scene modeling, and image restoration.

Computer vision is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. "Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding." As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems

1.3.5 e. HOW DOES COMPUTER VISION WORK?

One of the major open questions in both Neuroscience and Machine Learning is: How exactly do our brains work, and how can we approximate that with our own algorithms? The reality is that there are very few working and comprehensive theories of brain computation; so despite the fact that Neural Nets are supposed to “mimic the way the brain works,” nobody is quite sure if that’s actually true.

The same paradox holds true for computer vision — since we’re not decided on how the brain and eyes process images, it’s difficult to say how well the algorithms used in production approximate our own internal mental processes.

On a certain level Computer vision is all about pattern recognition. So one way to train a computer how to understand visual data is to feed it images, lots of images thousands, millions if possible that have been labeled, and then subject those to various software techniques, or algorithms, that allow the computer to hunt down patterns in all the elements that relate to those labels

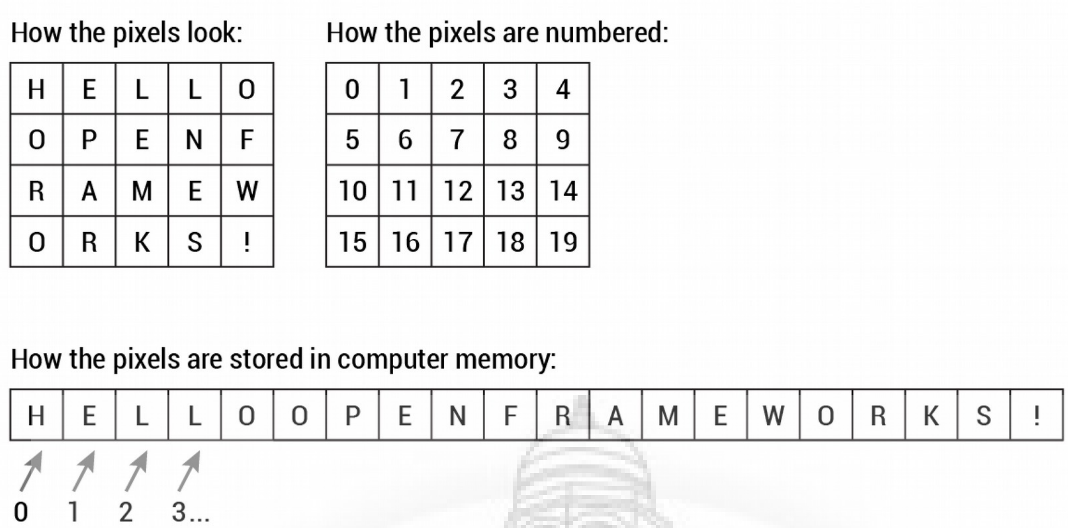
So, for example, if you feed a computer a million images of cats , it will subject them all to algorithms that let them analyze the colors in the photo, the shapes, the distances between the shapes, where objects border each other, and so on, so that it identifies a profile of what "cat" means. When it's finished, the computer will (in theory) be able to use its experience if fed other unlabeled images to find the ones that are of cat.

Let's leave our fluffy cat friends for a moment on the side and let's get more technical. Below is a simple illustration of the grayscale image buffer which stores our image of Abraham Lincoln. Each pixel's brightness is represented by a single 8-bit number, whose range is from 0 (black) to 255 (white):

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

PICTURE TO MATRIX TRANSFORMATION

This way of storing image data may run counter to your expectations, since the data certainly *appears* to be two-dimensional when it is displayed. Yet, this is the case, since computer memory consists simply of an ever-increasing linear list of address spaces.


How the pixels look:

H	E	L	L	O
O	P	E	N	F
R	A	M	E	W
O	R	K	S	!

How the pixels are numbered:

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

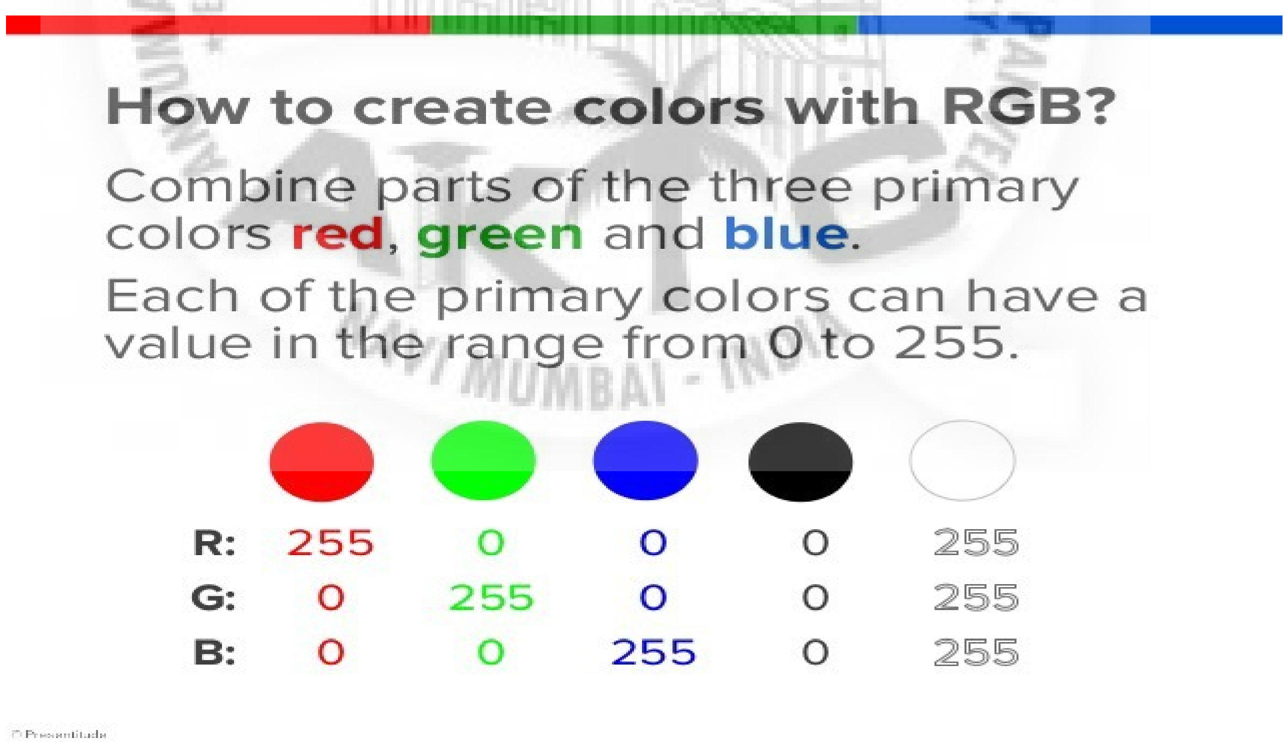
How the pixels are stored in computer memory:

H	E	L	L	O	O	P	E	N	F	R	A	M	E	W	O	R	K	S	!

0 1 2 3...

How pixels are stored in memory

Let's go back to the first picture again and imagine adding a colored one. Now things start to get more complicated. Computers usually read color as a series of 3 values — red, green, and blue (RGB) — on that same 0–255 scale. Now, each pixel actually has 3 values for the computer to store in addition to its position. If we were to colorize President Lincoln, that would lead to 12 x 16 x 3 values, or 576 numbers.



That's a lot of memory to require for one image, and a lot of pixels for an algorithm to iterate over. But to train a model with meaningful accuracy especially when you're talking about Deep Learning you'd usually need tens of thousands of images, and the more the merrier.

1.3.5 f. THE EVOLUTION OF COMPUTER VISION

Before the advent of deep learning, the tasks that computer vision could perform were very limited and required a lot of manual coding and effort by developers and human operators. For instance, if you wanted to perform facial recognition, you would have to perform the following steps:

- Create a database: You had to capture individual images of all the subjects you wanted to track in a specific format.
- Annotate images: Then for every individual image, you would have to enter several key data points, such as distance between the eyes, the width of nose bridge, distance between upper-lip and nose, and dozens of other measurements that define the unique characteristics of each person.
- Capture new images: Next, you would have to capture new images, whether from photographs or video content. And then you had to go through the measurement process again, marking the key points on the image. You also had to factor in the angle the image was taken.

After all this manual work, the application would finally be able to compare the measurements in the new image with the ones stored in its database and tell you whether it corresponded with any of the profiles it was tracking. In fact, there was very little automation involved and most of the work was being done manually. And the error margin was still large.

Machine learning provided a different approach to solving computer vision problems. With machine learning, developers no longer needed to manually code every single rule into their vision applications. Instead they programmed "features," smaller applications that could detect specific patterns in images. They then used a statistical learning algorithm such as linear regression, logistic regression, decision trees or support vector machines (SVM) to detect patterns and classify images and detect objects in them.

Machine learning helped solve many problems that were historically challenging for classical software development tools and approaches. For instance, years ago, machine learning engineers were able to create a software that could predict breast cancer survival windows better than human experts. However building the features of the software required the efforts of dozens of engineers and breast cancer experts and took a lot of time develop.

Classic machine learning approaches involved lots of complicated steps and required the collaboration of dozens of domain experts, mathematicians and programmers

Deep learning provided a fundamentally different approach to doing machine learning. Deep learning relies on neural networks, a general-purpose function that can solve any problem representable through examples. When you provide a neural network with many labeled examples of a specific kind of data, it'll be able to extract common patterns between those examples and transform it into a mathematical equation that will help classify future pieces of information.

For instance, creating a facial recognition application with deep learning only requires you to develop or choose a preconstructed algorithm and train it with examples of the faces of the people it must detect. Given enough examples (lots of examples), the neural network will be able to detect faces without further instructions on features or measurements.

Deep learning is a very effective method to do computer vision. In most cases, creating a good deep learning algorithm comes down to gathering a large amount of labeled training data and tuning the parameters such as the type and number of layers of neural networks and training epochs. Compared to previous types of machine learning, deep learning is both easier and faster to develop and deploy.

Most of current computer vision applications such as cancer detection, self-driving cars and facial recognition make use of deep learning. Deep learning and deep neural networks have moved from the conceptual realm into practical applications thanks to availability and advances in hardware and cloud computing resources.

1.4 RESULT AND DISCUSSION:

In this section, the results obtained from intelligent smart cane will be discussed. To analyze that, this paper details the analysis into different modules.

Developed Intelligent Cane

The internal design while performing a test run. A normal stick is used with LIDAR 360 sensors mounted and one camera attached on the top. A Raspberry Pi is pasted on the stick supplied by a 9 V battery.

A breadboard is also used for the connections. Breadboard and wires can be removed by using a Printing Circuit Board .Hence, the stick is very lightweight and portable which makes it easy to use.

1.4.1 RESULT ASSOSIATED WITH LIDAR 360:

To use the RPLIDAR, you need to start the motor spinning and tell it to start taking readings. It will then stream each reading it takes until you tell it to stop. Each reading includes the distance sensed, and the angle of the reading. To make things easier, the Skoltech library provides a convenient wrapper for this functionality, and that's what we'll use for this guide. This example will consume data from the RPLIDAR and display it on a 2.8" PiTFT display. We start by setting up a few things.

```
import os
from math import cos, sin, pi, floor
import pygame
from adafruit_rplidar import RPLidar

# Set up pygame and the display
os.putenv('SDL_FBDEV', '/dev/fb1')
pygame.init()
lcd = pygame.display.set_mode((320,240))
pygame.mouse.set_visible(False)
lcd.fill((0,0,0))
pygame.display.update()

# Setup the RPLidar
PORT_NAME = '/dev/ttyUSB0'
lidar = RPLidar(None, PORT_NAME)

# used to scale data to fit on the screen
max_distance = 0
```

Next we create a buffer in which to keep distance data. Each item in the list stores the distance measured at each degree for a complete rotation. We use this to maintain the most recent set of measurements even though each scan will be incomplete.

```
scan_data = [0]*360
```

We use the iter_scans function of the RPLIDAR object. This starts the motor and the scanning for us so we can skip that in our code. iter_scans accumulates measurements for a single rotation and returns a list of tuples, from which we are interested in the second and third items. These are the angle and the distance measured at that angle. Both are floating point values.Once we have a scan, we step through each data point. The floor of the angle is taken and used as the index at which to store the measurement value. Because it is possible that multiple consecutive measurements in a scan could beclose to the same angle (and thus having the same floor value) there is a small chance that some measurements arelost. But there is no real disadvantage to this, so it can be disregarded.

```
scan_data = [0]*360

try:
    print(lidar.info)
    for scan in lidar.iter_scans():
        for (_, angle, distance) in scan:
            scan_data[min([359, floor(angle)])] = distance
        process_data(scan_data)

except KeyboardInterrupt:
    print('Stoping.')
lidar.stop()
lidar.disconnect()
```

Once we have updated scan_data , it is passed to the process_data function .The process_data function can do anything from finding the closest object, to choosing the best direction to move in.

The only requirement is that it be as fast as possible. If it takes too long to process a scan, data from the RPLIDAR will eventually be dropped. Since we're constantly reading fresh data, dropping an occasional scan shouldn't be an issue.In this example we display the distance data on a PiTFT.

The first step of a pass is to clear the display. Then it looks at each angle: 0-359. If the distance recorded for that angle is zero, it's because a reading hasn't be acquired for that angle yet, and we can safely ignore it. If there is a distance recorded for the angle, max_distance is adjusted as necessary to keep the display scaled to fit on the screen. The angle is converted to radians (from degrees) and the vector for the reading is converted to a cartesian coordinate which is then used to plot the data point on the display.

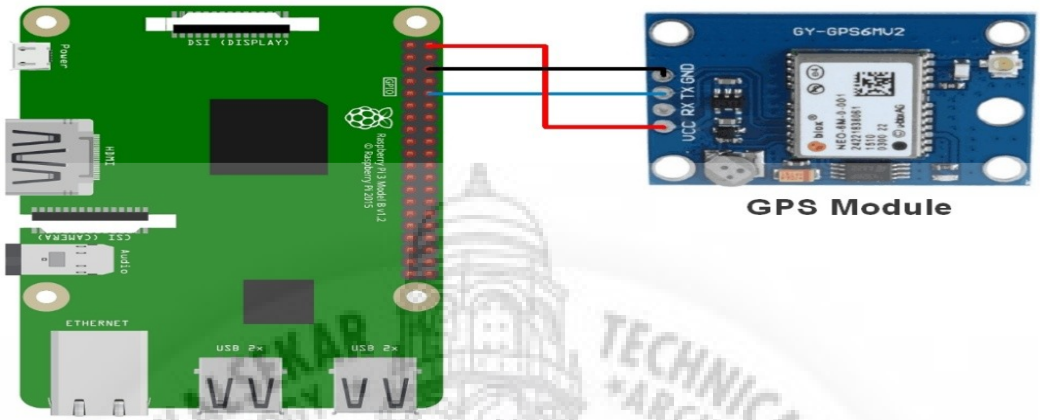
After all measurements have been plotted, the display is updated

```
def process_data(data):
    global max_distance
    lcd.fill((0,0,0))
    for angle in range(360):
        distance = data[angle]
        if distance > 0:                  # ignore initially ungathered data points
            max_distance = max([min([5000, distance]), max_distance])
            radians = angle * pi / 180.0
            x = distance * cos(radians)
            y = distance * sin(radians)
            point = (160 + int(x / max_distance * 119), 120 + int(y / max_distance * 119))
            lcd.set_at(point, pygame.Color(255, 255, 255))
    pygame.display.update()
```

FULL CODE

```
import os
from math import cos, sin, pi, floor
import pygame
from adafruit_rplidar import RPLidar

# Set up pygame and the display
os.putenv('SDL_FBDEV', '/dev/fb1')
pygame.init()
lcd = pygame.display.set_mode((320,240))
pygame.mouse.set_visible(False)
lcd.fill((0,0,0))
pygame.display.update()

# Setup the RPLidar
PORT_NAME = '/dev/ttyUSB0'
lidar = RPLidar(None, PORT_NAME)

# used to scale data to fit on the screen
max_distance = 0

#pylint: disable=redefined-outer-name,global-statement
def process_data(data):
    global max_distance
    lcd.fill((0,0,0))
    for angle in range(360):
        distance = data[angle]
        if distance > 0:                  # ignore initially ungathered data points
            max_distance = max([min([5000, distance]), max_distance])
            radians = angle * pi / 180.0
            x = distance * cos(radians)
            y = distance * sin(radians)
            point = (160 + int(x / max_distance * 119), 120 + int(y / max_distance * 119))
            lcd.set_at(point, pygame.Color(255, 255, 255))
    pygame.display.update()


scan_data = [0]*360

try:
    print(lidar.info)
    for scan in lidar.iter_scans():
        for (_, angle, distance) in scan:
            scan_data[min([359, floor(angle)])] = distance
        process_data(scan_data)

except KeyboardInterrupt:
    print('Stoping.')
lidar.stop()
```

1.4.2 RESULT ASSOSIATED WITH GPS MODULE:



USING PYTHON

Let's extract Latitude, Longitude and time information from NMEA GPGGA string received from GPS module using Python. And print them on console (terminal). By using these latitude and longitude, locate the current position on Google Map.

```
import serial                #import serial pacakge
from time import sleep
import webbrowser            #import package for opening link in browser
import sys                   #import system package

def GPS_Info():
    global NMEA_buff
    global lat_in_degrees
    global long_in_degrees
nmea_time = []
nmea_latitude = []
nmea_longitude = []
nmea_time = NMEA_buff[0]                    #extract time from GPGGA string
nmea_latitude = NMEA_buff[1]                #extract latitude from GPGGA
string
```

```
nmea_longitude = NMEA_buff[3]                    #extract longitude from GPGGA string

    print("NMEA Time: ", nmea_time,'\n')
    print ("NMEA Latitude:", nmea_latitude,"NMEA Longitude:", nmea_longitude,'\n')

    lat = float(nmea_latitude)                   #convert string into float for calculation
    longi = float(nmea_longitude)                #convertr string into float for calculation
     lat_in_degrees = convert_to_degrees(lat)    #get latitude in degree decimal format
    long_in_degrees = convert_to_degrees(longi)  #get longitude in degree decimal format
    #convert raw NMEA string into degree decimal format
def convert_to_degrees(raw_value):
    decimal_value = raw_value/100.00
    degrees = int(decimal_value)
    mm_mmmm = (decimal_value - int(decimal_value))/0.6
    position = degrees + mm_mmmm
    position = "%.4f" %(position)
    return position
    gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyS0")               #Open port with baud rate
GPGGA_buffer = 0
NMEA_buff = 0
lat_in_degrees = 0
long_in_degrees = 0

try:
    while True:
```

```
        received_data = (str)(ser.readline())                 #read NMEA string received
        GPGGA_data_available = received_data.find(gpgga_info)    #check for NMEA GPGGA string
        if (GPGGA_data_available>0):
 GPGGA_buffer = received_data.split("$GPGGA,",1)[1]  #store data coming after "$GPGGA," string
 NMEA_buff = (GPGGA_buffer.split(','))               #store comma separated data in buffer
GPS_Info()                                            #get time, latitude, longitude


print("lat in degrees:", lat_in_degrees," long in degree: ", long_in_degrees, '\n')
map_link = 'http://maps.google.com/?q=' + lat_in_degrees + ',' + long_in_degrees    #create link to plot location on Google map
print("<<<<<<<<press ctrl+c to plot location on google maps>>>>>>\n")    #press ctrl+c to plot on map and exit
print("------------------------------------------------------\n")


except KeyboardInterrupt:
    webbrowser.open(map_link)        #open current position information in google map
    sys.exit(0)
```

1.4.3 RESULT ASSOSIATE WITH OPEN CV:

OpenCV is a library of programming functions mainly aimed at real-time computer vision .Originally developed by Intel, it was later supported by Willow Garage then It seen (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

Applications

OPENFrameworks running the OpenCV add-on example

OpenCV's application areas include:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object detection
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality

REAL TIME OBJECT DETECTION (OPENCV):

LIBRARY:

We begin by importing packages on **Lines 2-8**. For this tutorial, you will need **imutils** and OpenCV 3.3

```
1. # import the necessary packages
2. from imutils.video import VideoStream
3. from imutils.video import FPS
4. import numpy as np
5. import argparse
6. import imutils
7. import time
8. import cv2
```

ARGUMENTS:

--prototxt : The path to the Caffe prototxt file.

--model: The path to the pre-trained model.

--confidence: The minimum probability threshold to filter weak detections. The default is 20%.

```
10. # construct the argument parse and parse the arguments
11. ap = argparse.ArgumentParser()
12. ap.add_argument("-p", "--prototxt", required=True,
13.    help="path to Caffe 'deploy' prototxt file")
14. ap.add_argument("-m", "--model", required=True,
15.    help="path to Caffe pre-trained model")
16. ap.add_argument("-c", "--confidence", type=float, default=0.2,
17.    help="minimum probability to filter weak detections")
18. args = vars(ap.parse_args())
```

CLASSES:

On **Lines 22-26** we initialize CLASS labels and corresponding random COLORS. For more information on these classes (and how the network was trained)

```
20. # initialize the list of class labels MobileNet SSD was trained to
21. # detect, then generate a set of bounding box colors for each class
22. CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
23.    "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
24.    "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
25.    "sofa", "train", "tvmonitor"]
26. COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))
```

VIDEO STREAM:

We load our serialized model, providing the references to our prototxt and model files on **Line 30** notice how easy this is in OpenCV 3.3.

Next let's initialize our video stream (this can be from a video file or a camera). First we start the VideoStream (**Line 35**), then we wait for the camera to warm up (**Line 36**), and finally we start the frames per second counter (**Line 37**). The VideoStream and FPS classes are partof my imutils package.

```
28. # load our serialized model from disk
29. print("[INFO] loading model...")
30. net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
31.
32. # initialize the video stream, allow the cammera sensor to warmup,
33. # and initialize the FPS counter
34. print("[INFO] starting video stream...")
35. vs = VideoStream(src=0).start()
36. time.sleep(2.0)
37. fps = FPS().start()
```

CONVERTING FRAME INTO BLOB

First, we read a frame (**Line 43**) from the stream, followed by resizing it (**Line 44**).Since we will need the width and height later, we grab these now on **Line 47**. This is followed by converting the frameto a blobwith the dnn module (**Lines 48 and 49**).

Now for the heavy lifting: we set the blob as the input to our neural network (**Line 53**) and feed the input through the net(**Line 54**) which gives us our detections.

At this point, we have detected objects in the input frame. It is now time to look at confidence values and determine if we should draw a box + label surrounding the object– you'll recognize this code block from last week

```
39.  # loop over the frames from the video stream
40.  while True:
41.      # grab the frame from the threaded video stream and resize it
42.      # to have a maximum width of 400 pixels
43.      frame = vs.read()
44.      frame = imutils.resize(frame, width=400)
45.
46.      # grab the frame dimensions and convert it to a blob
47.      (h, w) = frame.shape[:2]
48.      blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
49.          0.007843, (300, 300), 127.5)
50.
51.      # pass the blob through the network and obtain the detections and
52.      # predictions
53.      net.setInput(blob)
54.      detections = net.forward()
```

CREATING FRAME AND LABEL:

We start by looping over our detections, keeping in mind that multiple objects can be detected in a single image. We also apply a check to the confidence (i.e., probability) associated with each detection. If the confidence is high enough (i.e. above the threshold), then we'll display the prediction in the terminal as well as draw the prediction on the image with text and a colored bounding box. Let's break it down line-by-line:

Looping through our detections, first we extract the confidence value (**Line 60**).If the confidence is above our minimum threshold (**Line 64**), we extract the class label index (**Line 68**) and compute the bounding box coordinates around the detected object (**Line 69**). Then, we extract the *(x, y)*-coordinates of the box (**Line 70**) which we will will use shortly for drawing a rectangle and displaying text.

We build a text labelcontaining the CLASSname and the confidence (**Lines 73 and 74**).

Let's also draw a colored rectangle around the object using our class color and previously extracted *(x, y)*-coordinates (**Lines 75 and 76**).

In general, we want the label to be displayed above the rectangle, but if there isn't room, we'll display it just below the top of the rectangle (**Line 77**).

Finally, we overlay the colored text onto the frame using the *y*-value that we just calculated (**Lines 78 and 79**).

The remaining steps in the frame capture loop involve (1) displaying the frame, (2) checking for a quit key, and (3) updating our frames per second counter

```
55.     # loop over the detections
56.     for i in np.arange(0, detections.shape[2]):
57.         # extract the confidence (i.e., probability) associated with
58.         # the prediction
59.         confidence = detections[0, 0, i, 2]
60. 
61.         # filter out weak detections by ensuring the `confidence` is
62.         # greater than the minimum confidence
63.         if confidence > args["confidence"]:
64.             # extract the index of the class label from the
65.             # `detections`, then compute the (x, y)-coordinates of
66.             # the bounding box for the object
67.             idx = int(detections[0, 0, i, 1])
68.             box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
69.             (startX, startY, endX, endY) = box.astype("int")
70. 
71.             # draw the prediction on the frame
72.             label = "{}: {:.2f}%".format(CLASSES[idx],
73.                 confidence * 100)
74.             cv2.rectangle(frame, (startX, startY), (endX, endY),
75.                 COLORS[idx], 2)
76.             y = startY - 15 if startY - 15 > 15 else startY + 15
77.             cv2.putText(frame, label, (startX, y),
78.                 cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
```

DISPLAY VIDEOSTREAM:

The above code block is pretty self-explanatory — first we display the frame (**Line 82**). Then we capture a key press (**Line 83**) while checking if the 'q' key (for "quit") is pressed, at which point we break out of the frame capture loop (**Lines 86 and 87**).

Finally we update our fps counter (**Line 90**).

```
80.     # show the output frame
81.     cv2.imshow("Frame", frame)
82.     key = cv2.waitKey(1) & 0xFF
83. 
84.     # if the `q` key was pressed, break from the loop
85.     if key == ord("q"):
86.         break
87. 
88.     # update the FPS counter
89.     fps.update()
```

COST ANALYSIS:

The intelligent cane proposed here is of low cost as it onlymake use of ultrasonic sensors and buzzer. The increase in cost from the normal cane is due the addition of camera on the cane .These canes are quite affordable for the visually challenged people. The estimated cost of all four types of cane are as follows:

_ Normal Cane: Rs. 250

_ Smart Cane: Rs. 550

_ Multi-directional Cane: Rs. 700

_ Intelligent Cane: Rs. 10000

The cost is increased in intelligent cane due to addition of camera and earphones as well as replacment of Arduino with Raspberry Pi

1.5 CONCLUSION:

An Intelligent Cane is developed to help visually impaired people to walk freely without assistance. LIDAR sensors are used to detect the distance and direction of the object. User is notified using either beeps of the buzzer or through earphones. The stick developed is lightweight to carry and very cheap to buy. Also, the camera takes the image of the object and using object classification and image captioning,

an attempt is made to give more information about surrounding to the user so that they can understand the world better. Use of lithium ion battery makes the cane portable and use of GPS module gives location to their family members.

Guides the visually impaired person in his navigation independently in an efficient manner ensuring the person's safety. A variety of future scope are available that can be used of with the stick such as usage of Global positioning System can help the blind person to source to destination route information. GPS can help to find the shortest and best path as accordingly to Google (google map based on real time coordinates). GSM attachment can help in future for any immediate casualty help

- While walking, the smart cane detects nearby obstacles through infrared sensors and alerts its user through vibration and sound.
- It analyzes the user`s walking distance and send the information to the user`s smartphone.
- It also has the GPS communication module to guide the user using his/her current location and the navigation system.
- LED lights on the cane can let other people around the user know that there is a blind person nearby and also prevent unexpected safety accidents.
- These risk notifications are delivered to both the user and his/her guardians through the mobile phone application, and can be reported immediately if the user is in the wrong location or missing.
- The beacon module can also be used situations. These functions are very convenient for the blind. to send danger alerts to unspecified people near the blind person in case of dangerous

1.6 APPENDIX:

APPENDIX I:

This appendix consist of product called CANE INTELLIGENCE. The Working of the cane is depend on the components it gives the blind person idea about whatever is near that person. Image recognition, object detection, Emergency button are installed so that person get to know about the surrounding situation.

APPENDIX II:

LIDAR 360 is used for object detection upto 10m. Any kind of moving object can be detect by this sensors. And it is passed through the code which identifies what kind of object it is.

APPENDIX III:

Rasberry PI 4 module B is used to connect all the component of the product it is the microcontroller with 2GB RAM (QUAD CORE 64bit).

APPENDIX IV:

GPS NEO 3M it is used for tracing the person and the gives the location of person on the mobile phone. It contain a SOS button for emergency purpose.

APPENDIX V:

Camera module with raspberry pi 5 megapixel camera is connected to raspberry pi for object detection adjacent to lidar 360

APPENDIX VI:

Python IDE is required for coding the code related to image recognization with the help of OPEN CV package available in python IDE. This package consist of real time object detection for object detection.

1.7 LITERATURE CITED:

[1] "Global data on visual impairments 2010",

https://www.who.int/blindness/GLOBALDATAFINALforweb.pdf

[2] Sylvain Cardin, Daniel Thalmann and Frederic Vexo, Wearable Obstacle Detection System for visually impaired people Virtual Reality Lab- (VRlab)Ecole Polytechnique Fdrale de Lausanne (EPFL)CH-1015 Lausanne, Switzerlandsylvain.cardin, daniel.thalmann, frederic.vexo@epfl.ch

[3] O. B. Al-barrm and J. Vinouth, "3d ultrasonic stick for blind", IJLTET,vol. 3, no. 3, pp. 108-114, 2014.

[4] Srirama Divya, B. Navya, P. Suma Manasa, S. Chitra, "Ultrasonic And Voice Based Walking Stick For The Blind", (Affiliated to Jawaharlal Nehru Technological University) Hyderabad 500 090,2013.Page No-1.

[5] J. Sakhardande, P. Pattanayak and M. Bhowmick, “Arduino Based Mobility Cane”, International Journal of Scientific and Engineering Research, Vol. 4, Issue 4, pp 1163-1166, 2013.

[6] G. Gayathri, M. Vishnupriya, R. Nandhini, M.M. Banupriya, “Smart walking stick for visually impaired”, International Journal Of Engineering And Computer Science (IJECS), vol. 3, no. 3, pp. 4057-4061,2014.

[7] D. E. Gbenga , A. I. Shani , A. L. Adekunle, “Smart Walking Stick for visually impaired people using ultrasonic sensors and Arduino”,International Journal of Engineering and Technology 9(5):3435-3447October 2017 DOI:10.21817/ijet/2017/v9i5/170905302

[8] D.Sathya, S.Nithyaroopa, P.Betty, G.Santhoshni, S.Sabharinath,M.J.Ahanaa, Smart Walking stick for blind person, International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018,4531-4536

[9] M. H. Daudpota, A. A. Sahito, A. M. Soomro and F. S. Channar, “Giving blind a smart eye: Designing and modeling of intelligent white cane for blind people”, IEEE Global Humanitarian Technology Conference (GHTC), 2017, pp. 1-6.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[11]. A. Nurulnadwan, M.R. Nur-Hazwani, and A.M. Ariffin, “Assistive Courseware for Visually-Impaired,” (2009). in Proc. IVIC 2009, LNCS 5857, 2009, pp. 905-915.

[12]. Ariffin Abdul Mutalib & Faizah Maarof. (2010). Guidelines of Assistive Courseware (AC) for Hearing Impaired Students. In Proceedings of Knowledge Management International Conference 2010. UUM

[13]. Nurulnadwan Aziz, Nur Hazwani Mohamad Roseli, Erratul Shela Eshak, & Ariffin Abdul Mutalib. (2010). Assistive Courseware for the Visually Impaired based on Theory of Multiple Intelligence. In Proceedings of Knowledge Management International Conference 2010. UUM

1.8 ACKNOWLEDGEMNET:

We consider ourselfs lucky to work under guidance of such talented and experienced people who guided me all through the completion of our dissertation.

we express our deep sense of gratitude to my guide **Mr. Ubaid shah**, Lecturer of mechanical Engineering Department, for their generous assistance, vast knowledge, experience, views & suggestions and for giving me their gracious support. We owe a lot to them for this invaluable guidance in spite of their busy schedule.

We are grateful to **Dr. Abduk razzak honnutagi**, Director for his support and co-operation and for allowing me to pursue my Diploma Programme besides permitting me to use the laboratory infrastructure of the Institute.

I am thankful to to my H.O.D **Mr. Zakir** Ansari and Mr.**Prof. Rahul Thavai and Prof. Rizwan Shaikh** for their support at various stages.

Our thanks also goes to other staff members of mechanical Engineering Department, and to my respected college Anjuman-I-Islam's Kalsekar Technical campus school of engineering and technology, Panvel, and also to library staff for their assistance useful views and tips.

We also take this opportunity to thank my project members and my friends for their support and encouragement at every stage of my life.

Last but not the least, We are indebted to our parents and to all those who help me directly or indirectly in completion of this project report.

1. **Khan Irshad kazi**
2. **Sanadi Awees**
3. **Kazi Zammam**
4. **Siddiqui Zahoor Ahmed**

Bachelor of engineering

(Mechanical engineering)

Date: